



**សាកលវិទ្យាល័យ ភូមិន្ទនីតិសាស្ត្រ  
និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច**

**សារណាមញ្ញប័ត្រការសិក្សា**

**កម្មវិធីទូរស័ព្ទសម្រាប់លក់ និងទិញទំនិញ  
Mobile Commerce**

ស្រាវជ្រាវពីថ្ងៃទី១៦ ខែមីនា ឆ្នាំ២០២០ ដល់ថ្ងៃទី១៥ ខែឧសភា ឆ្នាំ២០២០

តាក់តែងឡើងដោយ  
និស្សិតឈ្មោះ **កៅ វ៉ាន់ឌី**  
**ទេន ច័ន្ទវឌ្ឍនា**

សាស្ត្រាចារ្យណែនាំ  
បណ្ឌិត **ទិន ហេង**

ថ្នាក់បរិញ្ញាបត្រ សេដ្ឋកិច្ចព័ត៌មានវិទ្យា  
ជំនាន់ទី ១៥

ឆ្នាំចូលសិក្សា ២០១៦  
ឆ្នាំសរសេរសារណា ២០២០

**សេចក្តីថ្លែងអំណរគុណ**

យើងខ្ញុំឈ្មោះ កៅ វ៉ាន់ឌី និងទែន ច័ន្ទវឌ្ឍនា ជានិស្សិតជំនាន់ទី១៥នៃសកលវិទ្យាល័យភូមិន្ទ នីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ចព័ត៌មានវិទ្យា ឆ្នាំសិក្សា ២០១៩-២០២០។ យើងខ្ញុំទាំងអស់គ្នា សូមបង្ហាញភាពដឹងគុណ គោរព និងថ្លែងអំណរគុណយ៉ាងជ្រាលជ្រៅជូនចំពោះ៖

លោកអ្នកមានគុណទាំងទ្វេ គឺលោកឪពុក និងអ្នកម្តាយជាទីគោរព និងជាទីស្រឡាញ់ដ៏ធំធេង ចំពោះកូនៗគ្រប់គ្នា ដែលលោកបានផ្តល់កំណើត ចិញ្ចឹមបីបាច់ ថែរក្សា ព្រមទាំងសន្តោសប្រណី ទំនុកបំរុងប្រៀនប្រដៅ ចង្អុលបង្ហាញដឹកនាំផ្លូវទៅរកមាតិក្រឹមត្រូវ អត់អោនលើកលែងទោស និង តែងតែជួយប្រឹក្សាយោបលរាល់បញ្ហានានាដែលកូនបានជួបប្រទះ។ លើសពីនេះលោកបានលះបង់ ទាំងកម្លាំងចិត្ត និងកម្លាំងកាយ ជំនះឧបសគ្គគ្រប់បែបយ៉ាងមិនគិតពីការនឿយហត់អ្វីទាំងអស់ ដើម្បីផ្គត់ផ្គង់គ្រួសារ និងកូនៗតាំងពីតូចបាតដើងក្រហម រហូតដល់ដឹងក្តីដល់សព្វថ្ងៃ។

សូមសម្តែងនូវការដឹងគុណយ៉ាងជ្រាលជ្រៅបំផុត ចំពោះលោកឯកឧត្តមសាកលវិទ្យាធិការ សកលវិទ្យាធិការរង លោក និងលោកស្រីសាស្ត្រាចារ្យ ព្រមទាំងបុគ្គលិកទាំងអស់នៃសាកលវិទ្យាល័យ ភូមិន្ទនីតិសាស្ត្រ និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច ដែលតែងតែខិតខំបង្ហាត់បង្រៀន ស្រាវជ្រាវនូវអ្វីដែល ថ្មីៗនិងមានប្រយោជន៍មកចែករំលែកដោយយកចិត្តទុកដាក់ ថែមទាំងអធ្យាស្រ័យ ណែនាំ និងជួយ សម្រួលនូវរាល់បញ្ហាដែលយើងខ្ញុំបានជួបប្រទះក្នុងការសិក្សាតាំងពីចូលរៀនថ្នាក់មូលដ្ឋានរហូត ដល់បញ្ចប់ថ្នាក់បរិញ្ញាបត្រនេះ។

ជាពិសេសលោកសាស្ត្រាចារ្យ បណ្ឌិត ទិន ហេង ដែលបានដឹកនាំ និងចំណាយពេលវេលាក្នុង ការផ្តល់ជាយោបល់ ឯកសារ ត្រួតពិនិត្យ វាយតម្លៃ ពិភាក្សា កែសម្រួល ផ្តល់ជាគំនិត និងខិតខំស្វែង រកឯកសារល្អៗ សម្រាប់ជាជំនួយក្នុងការសរសេរសារណាបញ្ចប់នេះ អោយជោគជ័យ។ ដោយមាន ការជួយជ្រោមជ្រែងយ៉ាងពេញទំហឹងពីសំណាក់លោកសាស្ត្រាចារ្យ ទើមជាហេតុនាំអោយការសរ សេរសារណានេះបញ្ចប់ដោយជោគជ័យជាស្ថាប័នយ៉ាងរលូន។

ជាកិច្ចបញ្ចប់ ក្រុមសិក្សាស្រាវជ្រាវយើងខ្ញុំសូមប្រសិទ្ធិពរជ័យដល់អ្នកមានគុណទាំងពីរ លោក គ្រូ សាស្ត្រាចារ្យ ក៏ដូចជាគណៈគ្រប់គ្រងទាំងអស់របស់សកលវិទ្យាល័យ ដែលបានជួយ អប់រំ ផ្តល់ ចំណេះដឹង ដល់យើងខ្ញុំតាំងពីដើមរហូតដល់ទទួលបានបរិញ្ញាបត្រ។

**អារម្ភកថា**

ក្នុងយុគសម័យនេះ ពិសេសនាចុងសតវត្សទី២១នេះ បច្ចេកវិទ្យាបានគ្របដណ្តប់ស្ទើរតែគ្រប់ ផ្នែកទាំងអស់ក្នុងជីវភាពរស់នៅប្រចាំថ្ងៃរបស់មនុស្ស និងមានជួយសម្រួលដល់ការលំបាកបានមួយ ផ្នែកធំ ដោយរាប់ចាប់ផ្នែកកសិកម្ម ឧស្សាហកម្ម ទេសចរណ៍ ទូរគមនាគមន៍ ប្រព័ន្ធផ្សព្វផ្សាយ ការ កំសាន្ត ការដោះដូរទិញលក់ទំនិញជាច្រើនទៀត។ ដោយសារតែមានការកើនឡើងនូវតម្រូវការឥត ឈប់ឈរ ស្របជាមួយការគាំទ្រផងនោះ ទើបអ្នកស្រាវជ្រាវជាច្រើនបានខិតខំបង្កើតឡើងនូវគំនិត ច្នៃប្រឌិតថ្មីៗជាច្រើនកើតឡើងជាបន្តបន្ទាប់ ដើម្បីផ្គត់ផ្គង់តម្រូវការរបស់មនុស្សជាតិទាំងនោះ ដែល ជាហេតុធ្វើអោយបច្ចេកវិទ្យាមានភាពទំនើបនិងសំបូរវែបវែប។ បច្ចេកវិទ្យាបានវិវត្តពីមួយដំណាក់កាល ទៅមួយដំណាក់កាលយ៉ាងឆាប់រហ័ស ដែលអាចនិយាយបានថាក្នុងសតវត្សទី២១នេះ បច្ចេកវិទ្យា ទំនើបបានកែប្រែរបៀបរបបរស់នៅរបស់មនុស្សស្ទើរទាំងស្រុង។

បើក្រឡេកមើលទៅលើការរស់នៅរបស់សង្គមមនុស្ស យើងប្រាកដជាឃើញនូវភាពខ្វះខាត មួយចំនួនដែលមិនទាន់បំពេញ ពិសេសគឺការដោះដូរទំនិញនៅមានលក្ខណៈបុរាណនៅឡើយ ប្រជាជនភាគច្រើនបានលក់នូវទំនិញរបស់ពួកគេនៅតាមផ្សារ ឬទីសាធារណៈផ្សេងៗដែលធ្វើអោយ ទំនិញរបស់ពួកគេមិនសូវជាមានគេដឹងច្រើន។ អាស្រ័យហេតុនេះ ទើបយើងទាំងពីរនាក់បាន សម្រេចចិត្តធ្វើការសិក្សាស្រាវជ្រាវទៅលើផ្នែក កម្មវិធីលើទូរស័ព្ទ ដោយផ្ដោតសំខាន់ទៅលើការ ទិញ លក់ទំនិញ រវាងអាជីវករ និងអ្នកទិញតាមរយៈប្រព័ន្ធអ៊ីនធឺណែត។ យើងសង្ឃឹមយ៉ាងមុតមាំ ថាការស្រាវជ្រាវរបស់យើងខ្ញុំនឹងជួយសម្រួល និងរួមចំណែកក្នុងការអភិវឌ្ឍន៍វិស័យបច្ចេកវិទ្យាអោយ មានភាពកាន់តែងាយស្រួលនិងរីកចម្រើន។

ជាការពិតណាស់ ដើម្បីសម្រេចបាននូវអ្វីមួយ យើងតែងតែជួបនូវឧបសគ្គ។ ដូចគ្នានេះដែរ ក្នុង ការកំណត់យកប្រធានបទនេះ ដើម្បីចងក្រងជាសារណានេះ យើងបានជួបជាមួយនឹងបញ្ហាជាច្រើន។ ជាសំណាងល្អ ដោយបានទទួលការណែនាំពីសំណាក់ លោកសាស្ត្រាចារ្យរួមផ្សំជាមួយការស្វែងរក ព័ត៌មាននានាពីប្រព័ន្ធអ៊ីនធឺណែត ទើបធ្វើអោយយើងខ្ញុំអាចឈានដល់ការបញ្ចប់យ៉ាង ជោគជ័យ។ ទោះបីយើងខ្ញុំបានធ្វើអស់ពីសមត្ថភាពហើយក្តី តែយើងខ្ញុំនៅតែជឿថាភាពខ្វះ ចន្លោះនៅតែមាន ហើយយើងខ្ញុំសង្ឃឹមថាអ្នកអាននឹងអធ្យាស្រ័យដល់យើងខ្ញុំចំពោះរាល់កំហុស ដែលកើតមានឡើងដោយអចេតនាក្តី និងអចេតនាក្តីដោយសេចក្តីអនុគ្រោះ។

ចុងបញ្ចប់នេះ យើងខ្ញុំសង្ឃឹមថា បណ្តាងកសាវដែលយើងខ្ញុំទាំងពីរបានចងក្រងឡើងនេះ នឹងផ្តល់ ផលប្រយោជន៍ដល់ប្អូនៗជំនាន់ក្រោយដែលចាប់អារម្មណ៍ផ្នែក កម្មវិធីលើទូរស័ព្ទ សំរាប់ធ្វើការ

ស្រាវជ្រាវធ្វើជាឯកសារយោង និងអាចចែករំលែកចំណេះដឹង បទពិសោធន៍នានាដែលយើងខ្ញុំបាន  
ឆ្លងកាត់តាមរយៈសៀវភៅមួយក្បាលនេះផងដែរ។

## **មាតិកា**

បញ្ជីរូបភាព..... v

### **សេចក្តីផ្តើម**

១. ចំណោទបញ្ហានៃការស្រាវជ្រាវ..... ១  
២. គោលបំណង ..... ២  
៣. វិសាលភាព និងដែនកំណត់នៃការសិក្សា..... ២  
៤. ផលប្រយោជន៍..... ៣

### **ជំពូកទី១**

#### **ស្ថានភាពទូទៅ និងប្រវត្តិនៃ E-Commerce**

១.១. ស្ថានភាពទូទៅនៃ E-Commerce..... ៤  
    ១.១.១. អ្វីជា E-Commerce?..... ៤  
    ១.១.២. អត្ថប្រយោជន៍នៃ E-Commerce..... ៧  
        ១.១.២.១. ចំពោះអ្នកលក់ និងអតិថិជន..... ៧  
        ១.១.២.២. ចំពោះសហគ្រាស និងក្រុមហ៊ុន ..... ៧  
        ១.១.២.៣. ចំពោះសង្គមជាតិ..... ៩  
    ១.១.៣. E-Commerce នៅក្នុងប្រទេសកម្ពុជា..... ១០  
១.២. ប្រវត្តិនៃ E-Commerce..... ១២  
    ១.២.១. ការវិវត្តន៍នៃ E-Commerce..... ១២  
        ១.២.១.១. CompuServe is Founded ..... ១២  
        ១.២.១.២. Michael Aldrich បានបង្កើតកម្មវិធីទិញទំនិញដោយអនឡាញ..... ១២  
        ១.២.១.៣. Boston Computer បានធ្វើការផ្លាស់ប្តូរ..... ១៣  
        ១.២.១.៤. ការដាក់លក់នូវបណ្តុំសៀវភៅជាច្រើនលើទីផ្សារ..... ១៣  
        ១.២.១.៥. ការបើកដំណើរការនូវកម្មវិធី Netscape Navigator ..... ១៣  
        ១.២.១.៦. ក្រុមហ៊ុន Amazon និង eBay បានបើក..... ១៣  
        ១.២.១.៧. ក្រុមហ៊ុន Paypal បើកដំណើរការទូទាត់ប្រាក់តាមអនឡាញ..... ១៣  
        ១.២.១.៨. ក្រុមហ៊ុន Alibaba បើកដំណើរការ..... ១៤  
        ១.២.១.៩. Google ណែនាំ Google Adwords..... ១៤

១.២.១.១០. ការណែនាំអោយស្គាល់សមាជិកភាពក្រុមហ៊ុន Amazon Prime .....	១៤
១.២.១.១១. Etsy ត្រូវបានបង្កើតឡើង .....	១៤
១.២.១.១២. ការបើកដំណើរការកម្មវិធី Square.....	១៤
១.២.១.១៣. ការបើកដំណើរការរបស់ BigCommerce .....	១៥
១.២.១.១៤. Google Wallet បានណែនាំជាវិធីសាស្ត្រទូទាត់ឌីជីថល.....	១៥
១.២.១.១៥. ហ្វេសប៊ុកដាក់ចេញនូវការឧបត្ថម្ភ .....	១៥
១.២.១.១៦. Apple Pay ណែនាំជាវិធីសាស្ត្រទូទាត់តាមទូរស័ព្ទ.....	១៦
១.២.១.១៧. គេហទំព័រ Jet.com ចាប់ផ្តើមដំណើរការ .....	១៦
១.២.១.១៨. ការទិញទំនិញតាម Instagram ត្រូវបានណែនាំ.....	១៦
១.៣. ប្រភេទនៃ E-Commerce.....	១៧
១.៣.១. B2B (Business to Business).....	១៧
១.៣.២. B2C (Business to Consumer).....	១៨
១.៣.៣. C2B (Consumer to Business) .....	១៨
១.៣.៤. B2G (Business to Government).....	១៩
១.៣.៥. C2C (Consumer to Consumer).....	២០
១.៣.៦. Mobile Commerce .....	២០
១.៤. ទិដ្ឋភាពក្នុងដំណើរការនៃ E-Commerce.....	២១
១.៤.១. Drop shipping .....	២១
១.៤.២. Wholesaling and Warehousing .....	២២
១.៥. យុត្តិសាស្ត្រ.....	២២
១.៥.១. រៀបចំគម្រោងក្នុងការធ្វើទីផ្សារជាមុន.....	២៣
១.៥.២. មានកាតសម្គាល់ពាណិជ្ជកម្ម.....	២៣
១.៥.៣. ផលិតផលមានគុណភាព (Product).....	២៣
១.៥.៤. តម្លៃ (Price).....	២៤
១.៥.៥. ការផ្សាយយោសនា (Promotion).....	២៤
១.៥.៦. ការចែកចាយ (Placement).....	២៤

## **ជំពូកទី២**

### **វេបទ្រទ្រង់**

២.១. ទ្រទ្រង់ដែលពាក់ព័ន្ធនឹងការបង្កើតកម្មវិធីទូរស័ព្ទ.....	២៥
២.១.១. អ្វីជា HTML?.....	២៥
២.១.២. អ្វីជា CSS?.....	២៦
២.១.៣. អ្វីជា JavaScript? .....	២៧
២.១.៤. អ្វីជា Vue.JS?.....	២៨
២.១.៥. អ្វីជា Vuetify? .....	៣០
២.១.៦. អ្វីជា Dart? .....	៣២
២.១.៧. អ្វីជា Flutter?.....	៣២
២.១.៧. អ្វីជា Firebase?.....	៣៣

## **ជំពូកទី៣**

### **ដំណើរការស្រាវជ្រាវ និងកំណត់មុខងារ**

៣.១. ការវិភាគដើម្បីជ្រើសរើសប្រធានបទ .....	៣៦
៣.២. ការវិភាគបម្រើបម្រាស់កម្មវិធីទូរស័ព្ទ .....	៣៦
៣.២.១. ការកំណត់មុខងារអ្នកប្រើប្រាស់ .....	៣៦
៣.៣. ការវិភាគលើរចនាសម្ព័ន្ធ Mobile app.....	៣៧
៣.៣.១. ការកំណត់នៅ Entities នៅក្នុង Page Admin.....	៣៧
៣.៣.២. ការកំណត់នៅ Entities នៅក្នុង Page User.....	៣៩
៣.៣.៣. សម្មតិកម្មរបស់ Entity Page .....	៤០
៣.៣.៣.១. Get Start Page .....	៤០
៣.៣.៣.២. Log In Page .....	៤១
៣.៣.៣.៣. Register Page.....	៤២
៣.៣.៣.៤. Home Page .....	៤២
៣.៣.៣.៥. Profile Page .....	៤៣
៣.៣.៣.៦. Chat Page.....	៤៤
៣.៣.៣.៧. Category Page .....	៤៥

៣.៣.៣.៨. Detail Page ..... ៤៥

**ជំពូកទី៤**

**ការរចនារូបវន្ត Database**

៤.១. ការកំណត់ឈ្មោះរបស់ Collection..... ៤៧

៤.២. ការពន្យល់អំពី Data Dictionary ..... ៤៧

**សេចក្តីសន្និដ្ឋាន និងការផ្តល់អនុសាសន៍**

១. សេចក្តីសន្និដ្ឋាន ..... ៥០

២. ការផ្តល់អនុសាសន៍ ..... ៥០

**ឯកសារយោង**

**ឧបសម្ព័ន្ធ**



## បញ្ជីរូបភាព

រូបភាពទី១ ៖បង្ហាញពី E-Commerce.....	៥
រូបភាពទី២ ៖E-Commerce គ្របដណ្តប់ពាសពេញសហរដ្ឋអាមេរិក .....	៦
រូបភាពទី៣ ៖ភាពងាយស្រួលក្នុងការបញ្ជាទិញទំនិញ .....	១២
រូបភាពទី៤ ៖ការវិវត្តនៃ E-Commerce.....	១៦
រូបភាពទី៥ ៖បង្ហាញពីការធ្វើអាជីវកម្មលក្ខណៈ Business to Business .....	១៧
រូបភាពទី៦ ៖បង្ហាញពីការធ្វើអាជីវកម្មលក្ខណៈ Business to Consumer.....	១៨
រូបភាពទី៧ ៖C2B(Consumer to Business).....	១៩
រូបភាពទី៨ ៖B2G (Business to Government) .....	១៩
រូបភាពទី៩ ៖C2C (Consumer to Consumer) .....	២០
រូបភាពទី១០ ៖Mobile Commerce .....	២១
រូបភាពទី១១ ៖Drop Shipping .....	២១
រូបភាពទី១២ ៖បង្ហាញពី Wholesaling and Warehousing.....	២២
រូបភាពទី១៣ ៖បង្ហាញពី HTML5.....	២៦
រូបភាពទី១៤ ៖បង្ហាញពី CSS.....	២៨
រូបភាពទី១៥ ៖បង្ហាញពី JavaScript.....	២៨
រូបភាពទី១៦ ៖បង្ហាញពី Vue.js .....	២៩
រូបភាពទី១៧ ៖បង្ហាញពី Vuetify.....	៣១
រូបភាពទី១៨ ៖បង្ហាញពី Dart .....	៣២
រូបភាពទី១៩ ៖បង្ហាញពី Flutter .....	៣៣
រូបភាពទី២០ ៖បង្ហាញពី Firebase .....	៣៣
រូបភាពទី២១ ៖បង្ហាញពី Dashboard លើ Page Admin.....	៣៧
រូបភាពទី២២ ៖បង្ហាញពី Product លើ Page Admin .....	៣៧
រូបភាពទី២៣ ៖បង្ហាញពី User លើ Page Admin .....	៣៨
រូបភាពទី២៤ ៖បង្ហាញពី Category លើ Page Admin .....	៣៨
រូបភាពទី២៥ ៖បង្ហាញពី Page Get Start.....	៤០
រូបភាពទី២៦ ៖បង្ហាញពី Login Page និងReset Password page.....	៤១
រូបភាពទី២៧ ៖បង្ហាញពី Register Page .....	៤២

រូបភាពទី២៨ ៖បង្ហាញពី Home Page.....	៤៣
រូបភាពទី២៩ ៖បង្ហាញពី Profile Page .....	៤៤
រូបភាពទី៣០ ៖បង្ហាញពី Chat Page.....	៤៤
រូបភាពទី៣១ ៖បង្ហាញពី Category Page .....	៤៥
រូបភាពទី៣២ ៖បង្ហាញពី Details ទំនិញរបស់អ្នកដទៃ .....	៤៦
រូបភាពទី៣៣ ៖បង្ហាញពី Detail ទំនិញផ្ទាល់ខ្លួន.....	៤៦

### សេចក្តីផ្តើម

#### ១. ចំណោទបញ្ហានៃការស្រាវជ្រាវ

បច្ចុប្បន្ននេះ បច្ចេកវិទ្យាត្រូវបានប្រើប្រាស់យ៉ាងទូលំទូលាយទូទាំងសកលលោក និងត្រូវបានប្រើប្រាស់ស្ទើរគ្រប់ផ្នែកនៃការរស់នៅរបស់មនុស្ស នាបណ្តាប្រទេសអភិវឌ្ឍ និងប្រទេសកំពុងអភិវឌ្ឍ។ បើក្រឡេកទៅមើលការរស់នៅរបស់ប្រជាជនកម្ពុជានាពេលបច្ចុប្បន្ន យើងសង្កេតឃើញថា ការទាញយកផលប្រយោជន៍ពីការប្រើប្រាស់បច្ចេកវិទ្យាគឺនៅមានកំរិត ប្រជាជនភាគច្រើននៅមិនទាន់ចេះទាញយកផលប្រយោជន៍ពីវានៅឡើយ។ ដោយសារការឃើញភាពងាយស្រួលក្នុងការរស់នៅជាមួយបច្ចេកវិទ្យាមធ្យមមួយទុកលំបាករបស់ប្រជាជនបែបនេះហើយទើបក្រុមយើងខ្ញុំបានស្វែងរកដំណោះស្រាយនិងបានលើកយកប្រធានបទមួយសំរាប់ជួយសំរួលដល់ការដោះដូរទិញលក់ទំនិញដែលអាចផ្តល់ភាពងាយស្រួលដល់អ្នកលក់ និងអ្នកទិញ ក្នុងការលក់ឬទិញផលិតផលផ្សេងៗតាមរយៈប្រព័ន្ធអ៊ីនធឺណែត។ ក្នុងប្រធានបទនេះផងដែរ បន្ទាប់ពីធ្វើការអោយស៊ីជម្រៅ យើងឃើញថាអ្នកលក់និងអ្នកទិញជាច្រើនបានជួបនូវបញ្ហាជាច្រើនដូចខាងក្រោម៖

- ពិបាកក្នុងការផ្សព្វផ្សាយផលិតផល
- មានការភ័យច្រលំនៅពេលទទួលបានការកម្មង់ច្រើនពីភ្ញៀវ
- លក់ និងផ្គត់ផ្គង់បានតែតំបន់ជិតៗ
- មិនមានភាពល្បីល្បាញ
- មិនសូវមានអតិថិជន
- ទំនិញអាចកប់ស្តុក
- អ្នកទិញពិបាកស្វែងរកទំនិញដែលខ្លួនចង់ទិញ
- ចំណាយពេលវេលាច្រើនក្នុងការទិញឬលក់ទំនិញ
- ទំនិញមិនមានប្រភពច្បាស់លាស់
- ប្រឈមនឹងទំនិញក្លែងក្លាយ
- ទំនិញដែលមានតម្លៃខ្ពស់ជាងទីផ្សារ
- ទំនិញគ្មានគុណភាព
- ទំនិញហួសកាលបរិច្ឆេទ
- មិនមានការទទួលខុសត្រូវច្បាស់លាស់ទៅលើទំនិញ

**២. គោលបំណង**

ដើម្បីដោះស្រាយរាល់បញ្ហាដែលកំពុងជួបប្រទះ ក៏ដូចជាជួយលើកកម្ពស់វិស័យពាណិជ្ជកម្ម និងធ្វើអោយប្រជាជនកាន់តែយល់ដឹងពីរបៀបធ្វើការវិទ្យាកាន់តែច្រើន យើងខ្ញុំបានបង្កើតកម្មវិធីទូរស័ព្ទ មួយដែលអនុញ្ញាតអោយម្ចាស់អាជីវកម្មបង្កើតនៅគណនេយ្យផ្ទាល់ខ្លួន ដែលបញ្ជាក់ពីអត្តសញ្ញាណ អាស័យដ្ឋាន លេខទូរស័ព្ទផ្ទាល់ខ្លួនច្បាស់លាស់ សម្រាប់បង្ហោះនូវមុខទំនិញនានា របស់ខ្លួន ផ្សព្វផ្សាយពីការបញ្ចុះតម្លៃ ពង្រីកទីផ្សារអោយកាន់តែទូលំទូលាយ និងផ្តល់ភាពងាយស្រួលដល់អតិថិជនអាចស្វែងរកបាននូវមុខទំនិញនានាតាមតម្រូវការដោយគ្រាន់តែចូលទៅកាន់ កម្មវិធីរួចធ្វើការបញ្ជាទិញជាការស្រេច។

**៣. វិសាលភាព និងដែនកំណត់នៃការសិក្សា**

ដោយសារតែសមត្ថភាពរបស់ពួកយើងនៅមានកំរិតនិងមានដែនកំណត់នៅឡើយ រួមជាមួយ ពេលវេលាមានកំណត់ផងនោះ ហេតុដូចនេះទើបយើងខ្ញុំបង្កើតកម្មវិធីទូរស័ព្ទនេះឡើងដោយផ្តល់ ត្រឹមមុខងារសំខាន់ៗមួយចំនួនដូចជា៖

- អនុញ្ញាតអោយម្ចាស់អាជីវកម្មអាចបង្កើតគណនេយ្យផ្ទាល់ខ្លួន ធ្វើការបង្ហោះនូវមុខទំនិញ កំណត់តម្លៃ ធ្វើការកែសម្រួល និងលុបចោលនូវមុខទំនិញនោះវិញបាន ពិសេសអាចឆាតពិភាក្សា ជាមួយអតិថិជនបានដោយសេរី។
- អតិថិជនទាំងអស់អាចស្វែងរកប្រភេទនៃមុខទំនិញ ដោយគ្រាន់ចុចលើបំណែងចែកមុខ ទំនិញតាមប្រភេទទំនិញ អាចរកទំនិញដែលចង់ទិញតាមរយៈមុខងារស្វែងរក អាចមើលព័ត៌មាន លំអិតនៃទំនិញ អាចធ្វើការរក្សាទុកទំនិញនោះដើម្បីធ្វើការបញ្ជាទិញនៅពេលក្រោយបាន អាចធ្វើ ការឆាតទាក់ទងផ្ទាល់ជាមួយម្ចាស់ទំនិញផ្ទាល់។
- រាល់ទំនិញទាំងអស់មុននឹងអាចដាក់លក់ជាសាធារណៈបានលុះត្រាតែឆ្លងកាត់ការត្រួត ពិនិត្យលើគុណភាព តម្លៃ ប្រភពនៃទំនិញ ប្រភេទនៃទំនិញ ... ពីសំណាក់អ្នកគ្រប់គ្រង ដើម្បីធានា ថាទំនិញដែលដាក់លក់គឺជាទំនិញស្របច្បាប់ មានគុណភាពស្របតាមមាត្រដ្ឋានត្រឹមត្រូវ និង តម្លៃសមរម្យអោយអ្នកទិញ។
- រាល់ព័ត៌មានទាំងអស់ក្នុងកម្មវិធី គឺសំរាប់ប្រើប្រាស់ទៅលើការលក់ និងទិញទំនិញតែ ប៉ុណ្ណោះ។
- អ្នកទិញក៏អាចធ្វើការ Rate ទៅលើមុខទំនិញបានផងដែរ តាមរយៈការអោយផ្កាយ និង បញ្ចេញមតិយោបល់បានផងដែរ។

**៤. ផលប្រយោជន៍**

ក្រោយពេលប្រើប្រាស់កម្មវិធីទូរស័ព្ទមួយនេះ អ្នកប្រើប្រាស់នឹងទទួលបាននូវ៖

- អាចធ្វើការលក់ទំនិញបានដោយមិនចាំបាច់មានហាង
- កាត់បន្ថយកាចំណាយបានមួយផ្នែក សម្រាប់អាជីវករថ្មីដែលមានបំណងក្នុងការចាប់ផ្តើម

អាជីវកម្ម និងងាយស្រួលក្នុងការគ្រប់គ្រង

- មិនតម្រូវអោយមានទំនិញក្នុងស្តុកច្រើន ត្រឹម១ ក៏អាចបង្ហោះលក់បាន
- សេរីភាពក្នុងការលក់ អាជីវករអាចលក់បានគ្រប់ទីកន្លែង
- ចំណាយពេលតិច អាចរកប្រាក់ចំណេញបានច្រើន
- ងាយស្រួលក្នុងការពង្រីកអាជីវកម្ម
- ងាយស្រួលផ្សព្វផ្សាយទៅអោយសាធារណៈជនបានឃើញ
- កាន់តែមានភាពងាយស្រួលដល់អតិថិជនក្នុងការស្វែងរក និងមើលទំនិញជាច្រើនដែល

គេដាក់លក់

- ចំណេញពេលវេលា និងថវិការព្រោះអតិថិជនបញ្ជាទិញទំនិញដោយមិនបាច់ចេញពីផ្ទះ
- ទទួលបានព័ត៌មានគ្រប់គ្រាន់ពីមុខទំនិញនីមួយៗ
- ផ្តល់ជាឱកាសការងារសំរាប់អ្នកដឹកជញ្ជូន។

## ជំពូកទី១

### ស្ថានភាពទូទៅ និងប្រភេទនៃ E-Commerce

#### ១.១. ស្ថានភាពទូទៅនៃ E-Commerce

##### ១.១.១. អ្វីជា E-Commerce ?

E-Commerce គឺជាការលក់ទំនិញដោយប្រើប្រាស់បច្ចេកវិទ្យាធ្វើជាយានដែលអ្នកខ្លះស្គាល់ថាជា Electronic Commerce (ប្រព័ន្ធពាណិជ្ជកម្មអេឡិចត្រូនិច)។ ប្រព័ន្ធពាណិជ្ជកម្មអេឡិចត្រូនិចពឹងទៅលើប្រព័ន្ធបណ្តាញ ហើយនិងប្រព័ន្ធអ៊ីនធឺណែត។ ប្រព័ន្ធនេះបានធ្វើការទូទាត់ប្រាក់តាមរយៈប្រព័ន្ធធនាគារ ដោយអ្នកលក់និងអ្នកទិញអាចប្រើប្រាស់វិសាកាត ម៉ាស់ស្ត្រីកាត រឺក៏អាចប្រើប្រាស់នូវសេវាទូទាត់ប្រាក់ដ៏ពេញនិយមមួយចំនួនដូចជា Paypal, Wing, True Money, ABA និង E-money ជាដើម ។ ជាទូទៅក្រោយពីការទូទាត់ប្រាក់តាមប្រព័ន្ធស្វ័យប្រវត្តិរួចរាល់ហើយអ្នកលក់និងអ្នកទិញនឹងធ្វើការទំនាក់ទំនងគ្នា តាមរយៈសារអេឡិចត្រូនិច ដើម្បីរកវិធីសាស្ត្រដែលងាយស្រួលក្នុងការបញ្ជូនទំនិញអោយដល់ដៃអតិថិជនដោយសុវត្ថិភាព។ ប្រព័ន្ធពាណិជ្ជកម្មអេឡិចត្រូនិចអនុញ្ញាតអោយអតិថិជនធ្វើការស្វែងរកទិញទំនិញ ជាច្រើនហាងក្នុងពេលដំខ្លី ដោយមិនចាំបាច់ចំណាយពេលវេលាក្នុងការដើររកទំនិញ។ អ្វីដែលអ្នកត្រូវការនោះគឺ ប័ណ្ណឥណទាន កុំព្យូទ័រ ទូរស័ព្ទ និងប្រព័ន្ធអ៊ីនធឺណែត តែប៉ុណ្ណោះ។ តាមរយៈប្រព័ន្ធពាណិជ្ជកម្មអេឡិចត្រូនិច អ្នកមិនត្រឹមតែលក់ទំនិញនៅក្នុងតំបន់របស់អ្នកតែប៉ុណ្ណោះទេ អ្នកអាចធ្វើការផ្សព្វផ្សាយទំនិញ និងធ្វើការពាណិជ្ជកម្មបានយ៉ាងទូលំទូលាយ និងចំណាយថវិកាតិច។ សព្វថ្ងៃនេះ ប្រព័ន្ធពាណិជ្ជកម្មអេឡិចត្រូនិចមិនត្រឹមតែអាចអោយអ្នកធ្វើការលក់ទំនិញប្រើប្រាស់ តែប៉ុណ្ណោះទេ អ្នកក៏អាចលក់នូវទំនិញដែលជាប្រភេទបច្ចេកវិទ្យាបានផងដែរ ក្នុងនោះមានដូចជា កម្មវិធីកុំព្យូទ័រ, ល្បែងកំសាន្ត, ចម្រៀង, វីដេអូជាដើម។ សព្វថ្ងៃនេះប្រព័ន្ធប្រព័ន្ធពាណិជ្ជកម្មអេឡិចត្រូនិចបានធ្វើការអភិវឌ្ឍដោយក្រុមហ៊ុនធំៗជាច្រើន ក្នុងគោលបំណង បង្កើននូវចំនួនអតិថិជនរបស់ពួកគេនៅពាសពេញពិភពលោក។ ទោះបីយ៉ាងណាក៏ដោយ តម្លៃដែលផលិតផលដែលត្រូវលក់មានការខុសគ្នាទៅតាមតំបន់។

ក្នុងវិស័យជំនួញ E-commerce គឺជាប្រតិបត្តិការជំនួញ Online ដែលមានលក្ខណៈ ពិសេសភ្ជាប់ទៅនឹងប្រព័ន្ធ Computer នៃអាជីវកម្មសាងជំនួញ និងអ្នកទិញ។ តាមរយៈ ប្រតិបត្តិការប្រព័ន្ធអេឡិចត្រូនិចបានបញ្ចូលនូវការផ្ទេរភាពជាម្ចាស់កម្មសិទ្ធិ ឬសិទ្ធិក្នុងការប្រើប្រាស់ផលិតផល ឬសេវាកម្មណាមួយពីមនុស្សម្នាក់ទៅមនុស្សម្នាក់ទៀត។ មនុស្សភាគច្រើនពេញនិយមជាមួយនឹង

ជំនួញតាមប្រព័ន្ធអេឡិចត្រូនិចប្រភេទ Business-to-consumer (B2C)។ និយាយជារួម E-commerce គឺជាការទិញ លក់ផលិតផល និងសេវាកម្មរវាងម្ចាស់ជំនួញ និងអ្នកប្រើប្រាស់តាមប្រព័ន្ធអ៊ីនធឺណែត។

តាមរយៈ E-commerce អតិថិជនអាចទិញទំនិញនៅពេលណាក៏បាន ទោះជាពេលយប់ឬថ្ងៃ និងមិនថានៅកន្លែងណា អោយតែមានភ្ជាប់ប្រព័ន្ធអ៊ីនធឺណែត។ អតិថិជនមិនចាំបាច់ចំណាយពេលវេលា និងធនធានធ្វើដំណើរទៅកាន់ទីកន្លែងលក់ ឬសហគ្រាសផ្ទាល់ដើម្បីទិញទំនិញ ហើយក៏មិនចាំបាច់រៀបចំពេលវេលាទៅតាមពេលវេលារបស់ហាងទំនិញដែរ។ ភាគច្រើននៃអតិថិជនមិនចូលចិត្តទៅស្វែងរកទំនិញនៅតាមបណ្តាផ្សារនានាទេ ម្យ៉ាងវិញទៀតអតិថិជនអាចនឹងមានការលំបាកក្នុងការរកទិញទំនិញប្រភេទខុសគ្នាក្នុងពេលតែមួយ ព្រោះទំនិញទាំងអស់មិនសុទ្ធ សឹងតែស្ថិតនៅក្នុងហាងជាមួយគ្នានោះទេ។ វត្តមានរបស់ E-Commerce បានផ្លាស់ ប្តូរការលំបាករបស់អតិថិជនអោយមានការងាយស្រួលក្នុងការរស់នៅប្រចាំថ្ងៃ ព្រោះអតិថិជនអាចស្វែងទំនិញ គ្រប់ប្រភេទដោយមិនចាំបាច់ចំណាយពេលវេលាទៅផ្សារ។ ពួកគេអាចប្រើប្រាស់ប្រព័ន្ធអ៊ីនធឺណែត ដើម្បីធ្វើការទិញទំនិញគ្រប់ពេលវេលា និងគ្រប់ទីកន្លែង។ វាបានបង្ករលក្ខណៈងាយស្រួលយ៉ាង ច្រើនទៅដល់អតិថិជន ជាពិសេសអតិថិជនដែលរស់ពុំសូវមានពេល។



រូបភាពទី១. បង្ហាញពី E-Commerce

E-Commerce មានប្រព័ន្ធបញ្ជាទិញដែលមានលក្ខណៈងាយស្រួលក្នុងការប្រើប្រាស់ ទោះជា នៅក្នុងគេហដ្ឋាន ឬការិយាល័យ។ អតិថិជននៅតែអាចស្វែងរកផលិតផលដែលខ្លួនចង់បានពីអ្នក លក់ផ្សេងៗគ្នា ដោយចូលទៅកាន់វេបសាយជាច្រើនដើម្បីទទួលព័ត៌មាន ធ្វើឲ្យអតិថិជន

អាច ជ្រើសរើសនូវផលិតផលទៅតាមចំណង់ចំណូលចិត្តរបស់ពួកគេ។ ទន្ទឹមនេះពាណិជ្ជកម្មអេឡិចត្រូនិចក៏ផ្តល់នូវភាពងាយស្រួលដល់អតិថិជនក្នុងការជ្រើសរើសតម្លៃផងដែរ ព្រោះអតិថិជនអាច ធ្វើការប្រៀបធៀបតម្លៃហើយទើបសំរេចចិត្តទិញជាក្រោយ។ ម្យ៉ាងទៀតតម្លៃផលិតផល ឬ តម្លៃ សេវាកម្មដែលអ្នកលក់ដាក់លក់តាម E-Commerce ភាគច្រើន មានតម្លៃថោកជាងទីផ្សារផ្ទាល់ ហើយអ្វីដែលកាន់តែពិសេសទៀតនោះ តម្លៃនៃ E-Commerce មិនមានការកែប្រែទេ ដែលបង្កការងាយស្រួលដល់អតិថិជនក្នុងការជ្រើសរើសនូវទំនិញណាមួយឲ្យសមស្របនឹងតម្លៃ។

តាមរយៈ E-Commerce អ្នកលក់ និងអ្នកទិញទទួលបានសុវត្ថិភាពដូចគ្នាក្នុងការទូរទាត់ប្រាក់។ អ្នកលក់ទទួលបានថ្លៃទំនិញងាយស្រួល និងឆាប់រហ័ស ព្រោះថាតម្លៃទំនិញត្រូវបានផ្ទេរ ពីគណនីអ្នកទិញទៅអ្នកលក់យ៉ាងលឿន និងមានទំនុកចិត្តថាត្រឹមត្រូវ គ្មានការក្លែងបន្លំ។ ចំនែកឯ អ្នកទិញគឺមានសុវត្ថិភាពក្នុងការចាយវាយដោយមិនចាំបាច់យកលុយសុទ្ធទៅដើរទិញទំនិញស្ទើរគ្រប់ទីកន្លែងនោះទេ។ E-Commerce បានបើកទីផ្សារអោយកាន់តែទូលំទូលាយជាអន្តរជាតិ អ្នកលក់អាចមានលទ្ធភាពតាំងបង្ហាញទំនិញថ្មីៗមានគ្រប់ពេល និងអ្នកទិញអាចទិញទំនិញបាន២៤ម៉ោងក្នុងមួយថ្ងៃ មាននៃថាគ្រប់ពេលវេលា និងគ្រប់ទីកន្លែងដោយគ្រាន់តែចូលទៅក្នុងកម្មវិធីរួចធ្វើការបញ្ជាទិញជាការស្រេច។ មិនតែប៉ុណ្ណោះអ្នកប្រើប្រាស់ក៏អាចធ្វើការប្រៀបធៀបអំពីតម្លៃ និងគុណភាពជាមួយហាងដទៃទៀតបានយ៉ាងងាយស្រួលតាមរយៈការ review ទំនិញរបស់ពួកគេ។



រូបភាពទី២. E-Commerce គ្របដណ្តប់ពាសពេញសហរដ្ឋអាមេរិក



ពាណិជ្ជកម្ម E-Commerce គ្រាន់តែជាដំណើរការនៃការទិញ និងលក់ ទំនិញតាមរយៈ មធ្យោបាយអេឡិចត្រូនិចដូចជាកម្មវិធីទូរស័ព្ទ កុំព្យូទ័រតាមរយៈប្រព័ន្ធអ៊ីនធឺណិត។ E-Commerce សំដៅទៅលើការលក់លើអេឡិចត្រូនិច និងផ្ទេរប្រាក់តាមរយៈអេឡិចត្រូនិច។ E-Commerce បាន កើនឡើងយ៉ាងខ្លាំងនៅក្នុងប៉ុន្មានទសវត្សរ៍ចុងក្រោយនេះ អមជាមួយប្រជាប្រិយភាពយ៉ាងខ្លាំងខ្លា វាក៏បានជួយសម្រួលការងាររបស់ហាងលក់ទំនិញបានយ៉ាងល្អប្រពៃ។

**១.១.២. អត្ថប្រយោជន៍នៃ E-Commerce**

**១.១.២.១. ចំពោះអ្នកលក់ និងអតិថិជន**

**ក. ផ្តល់ភាពងាយស្រួល និងចំណេញពេលវេលា**

អ្នកលក់ងាយស្រួលក្នុងការបង្ហាញនូវប្រភេទទំនិញខុសៗគ្នាជាច្រើនដល់អតិថិជនសំរាប់ ជ្រើសរើស។ ប្រភេទទំនិញ និងតម្លៃត្រូវបានបង្ហាញយ៉ាងច្បាស់ទៅកាន់អ្នកទិញ។ អ្នកលក់មិនចាំ បាច់ពិបាករង់ចាំអ្នកទិញនៅក្នុងហាងដូចពីមុន។ ម៉្យាងទៀតពាណិជ្ជករមិនចាំបាច់ចំណាយពេល វេលាច្រើន ទៅលើការគ្រប់គ្រងអតិថិជន និងការគ្រប់គ្រងផ្នែកគណនេយ្យជាដើម។ E-Commerce អាចអោយពាណិជ្ជករធ្វើការលក់ ហើយអតិថិជនអាចទិញទំនិញ 24 ម៉ោងក្នុងមួយថ្ងៃ និង 7 ថ្ងៃក្នុង 1 សប្តាហ៍ ដែលជាឱកាសល្អក្នុងការ ស្វែងរកប្រាក់ចំណូល។

**ខ. មានជម្រើសច្រើន និងតម្លៃថោក**

អ្នកផ្គត់ផ្គង់ ឬអ្នកលក់អាចផ្តល់ជូនអតិថិជននូវព័ត៌មានលំអិតនៃផលិតផល ឬសេវាកម្ម របៀប ប្រើប្រាស់ និងអត្ថប្រយោជន៍ ដែលជាការទាក់ទាញចំណាប់អារម្មណ៍អតិថិជនអោយចង់ ទិញ ផលិតផល ឬសេវាកម្មទាំងនោះ។

**គ. សុវត្ថិភាពក្នុងការទូទាត់**

តាមរយៈ E-Commerce អ្នកលក់ និងអ្នកទិញទទួលបានសុវត្ថិភាពដូចគ្នាក្នុងការទូទាត់ ប្រា ក់។ អ្នកលក់ទទួលបានថ្លៃទំនិញងាយស្រួល និងឆាប់រហ័ស ព្រោះថាតម្លៃទំនិញត្រូវបានផ្ទេរ ពី គណនីអ្នកទិញទៅអ្នកលក់យ៉ាងលឿន និងមានទំនុកចិត្តថាត្រឹមត្រូវ គ្មានការក្លែងបន្លំ។ ចំនែកឯ អ្នកទិញគឺមានសុវត្ថិភាពក្នុងការចាយវាយ ដោយមិនចាំបាច់យកលុយសុទ្ធតទៅដើរទិញទំនិញស្ទើរ គ្រប់ទីកន្លែងនោះទេ។

**១.១.២.២. ចំពោះសហគ្រាស និងក្រុមហ៊ុន**

**ក. ភាគបន្ថយការចំណាយ**

ប្រតិបត្តិការនៃអង្គភាព ឬក្រុមហ៊ុនមួយមិនមែនជាងាយស្រួលនោះទេគឺគេត្រូវការ ចំណាយ ជាច្រើន ដែលចំណាយមុនដំបូងនោះការចំណាយទៅលើទីតាំង។ ប៉ុន្តែការចំណាយនោះនឹង ត្រូវ

កាត់បន្ថយចំពោះអ្នកផ្គត់ផ្គង់ដែលប្រើប្រាស់E-Commerce នៅក្នុងជំនួញរបស់គេ។ ទីតាំងគឺជាកត្តាលើករណីចំណាប់ក្នុងការស្វែងរកនូវប្រាក់ចំណូលនៅក្នុងពាណិជ្ជកម្ម ព្រោះថាទីតាំងគឺជាកន្លែងជួបគ្នារវាងអ្នកលក់ និងអ្នកទិញ។ ដូចនេះពាណិជ្ជករត្រូវការចំណាយច្រើនទៅលើទីតាំងក្នុងគោលបំណងទាក់ទាញអតិថិជនផង និងធ្វើការប្រកួតប្រជែងផង។ នេះបង្ហាញអោយឃើញថាការចូលខ្លួនរបស់ពាណិជ្ជកម្មអេឡិចត្រូនិច បានធ្វើឲ្យពាណិជ្ជករកាត់បន្ថយស្ទើរតែទាំងស្រុងទៅលើការចំណាយលើទីតាំង ព្រោះពាណិជ្ជករគ្រាន់តែមានគេហទំព័រ កម្មវិធីទូរស័ព្ទ ឬក៏ប្រើប្រាស់សេវាកម្មនៃ គេហទំព័រដែលមានប្រជាប្រិយភាពក្នុងការស្វែងរកអតិថិជន។ ទាំងអស់នេះនឹងអាចកាត់បន្ថយ យ៉ាងច្រើនក្នុងការចំណាយទៅលើទីតាំង។

**ផែនការចំណាយទៅលើកម្មវិធីផ្សព្វផ្សាយ**

**ព្រះរាជាណាចក្រកម្ពុជា**

អ

ឡិចត្រូនិចមិនចាំបាច់ទាមទារឲ្យអ្នកផ្គត់ផ្គង់ចំណាយទៅលើបុគ្គលិកលក់ផ្ទាល់ អ្នកធ្វើទីផ្សារ អ្នកថែរក្សាហាង។ដូចនេះអាចកាត់បន្ថយបុគ្គលិកមួយផ្នែកធំ ដែលជាហេតុនាំឲ្យការចំណាយ ទៅលើកម្លាំង កម្មមានការកាត់បន្ថយជាលំដាប់។ ចំណាយថ្លៃដើមប្រតិបត្តិការទាប និងផ្តល់ ជូននូវសេវាកម្មដែលរហ័សទាន់ចិត្ត។

ការផ្សព្វផ្សាយក៏ជាផ្នែកមួយសំខាន់នៅក្នុងដំណើរនៃការធ្វើពាណិជ្ជកម្មដែរ ព្រោះថាការផ្សព្វផ្សាយបានជួយទាក់ទាញអតិថិជនយ៉ាងច្រើន។ ការផ្សព្វផ្សាយអាចកើតឡើងតាមរយៈទូរ ទស្សន៍ វីដេអូ សារព័ត៌មាន អ៊ីនធើណែត ការចែកខិតប័ណ្ណតាមផ្លូវជាដើម ដែលការផ្សព្វផ្សាយទាំងនោះសុទ្ធតែទាមទារឲ្យមានការចំណាយច្រើន ដែលរួមមាន ការចំណាយលើការបែបបទទិញលក់ ការបំពេញបែបបទនាំចេញ នាំចូល។

អង្គការអ្នកលក់ និងអ្នកទិញ ទំនាក់ទំនងគ្នាដោយផ្ទាល់តាមរយៈអ៊ីនធើណែតដោយមិនចាំបាច់ស្វែងរកភាគីទី៣មកទំនាក់ទំនងឡើយ។ នេះជាហេតុដែលបណ្តាលឲ្យថ្លៃទំនាក់ទំនងធ្លាក់ចុះ។ ការធ្លាក់ចុះនៃថ្លៃចំណាយទៅលើការស្វែងរកព័ត៌មានដែលពាក់ព័ន្ធ ផ្តល់ផលប្រយោជន៍ជាច្រើន ដល់អតិថិជន។ នៅពេលដែលក្រុមហ៊ុនបានកាត់បន្ថយចំណាយខ្លះខ្លាយក្នុងប្រតិបត្តិការរបស់ខ្លួនហើយ ធ្វើអោយផលិតផលទំនិញដែលខ្លួនផលិតបានអាចលក់ក្នុងតម្លៃសមរម្យ សំរាប់ការប្រកួតប្រជែង លើទីផ្សារក៏ដូចជាការផ្តល់នូវភាពពេញចិត្តខាងតម្លៃដល់អតិថិជន។

**១. ការខិតខំកាត់តែឱ្យទាបនៃការចំណាយ**

ការទាក់ទាញអតិថិជននេះទៀតសោតមិនមែនងាយស្រួលនោះឡើយប៉ុន្តែE-Commerce បានធ្វើអោយអង្គការ ឬក្រុមហ៊ុនទទួលបាននូវអតិថិជន ព្រោះដោយសារតែបច្ចេកវិទ្យាកាន់តែ

ជឿនលឿន បានធ្វើមនុស្សជុំវិញពិភពលោកមានការប្រើប្រាស់ប្រព័ន្ធអ៊ីនធឺណែត ស្ទើរតែទាំងអស់ ដែលជាហេតុធ្វើឲ្យពាណិជ្ជកម្មអេឡិចត្រូនិកកាន់តែមានការពេញនិយមពីមួយ ថ្ងៃទៅមួយថ្ងៃ។ E-Commerce លុបបំបាត់គំលាតរវាងអ្នកផ្គត់ផ្គង់ និងអតិថិជន ធ្វើអោយអ្នកផ្គត់ផ្គង់ខិតកាន់តែជិតទៅដល់អ្នកប្រើប្រាស់ផ្ទាល់ តាមរយៈនេះអ្នកផ្គត់ផ្គង់អាចដឹងពីឥរិយាបថរបស់ ទីផ្សារ និងចំណូលចំណូលចិត្តរបស់អ្នកប្រើប្រាស់បានយ៉ាងងាយស្រួល ជាប្រយោជន៍ទៅដល់ផលិតកម្ម និងសេវាកម្ម ផលិតអោយសមស្របទៅនឹងតម្រូវការ។

**គ. E-Commerce បង្កើតអោយមាន Mass customization**

ពាណិជ្ជកម្មអេឡិចត្រូនិកបង្កើតអោយមានទំនាក់ទំនងយ៉ាងជិតស្និទ្ធ ដែលធ្វើអោយតម្រូវការ និងការបញ្ជាទិញរបស់អតិថិជនបញ្ជូនយ៉ាងឆាប់រហ័សដល់អ្នកផ្គត់ផ្គង់ ហើយអ្នកផ្គត់ផ្គង់អាចឆ្លើយ តបតាមការបញ្ជាទិញវិញយ៉ាងរហ័សទាន់ចិត្ត។ ប្រតិបត្តិការ Online រួមទាំងប្រព័ន្ធបញ្ជាទិញ ងាយស្រួលប្រើធ្វើអោយអតិថិជនអាចបញ្ជាទិញផលិតផលដែលគេចង់បានទៅតាមចំនង់ ចំណូលចិត្ត ដូចជា បរិមាណ ឯកតាទំនិញ ពិណ លក្ខណៈពិសេសរបស់ផលិតផល ដែលពួកគេស្រលាញ់ ហើយផលិតទាំងនោះ នឹងផលិតឆ្លើយតបក្នុងរយៈពេលខ្លី។ ឧទាហរណ៍៖ តាមរយៈការ កុម្មតាម Online អតិថិជននឹងទទួលបានឡានថ្មីក្នុងរយៈពេល២ ឬ ៣ថ្ងៃ ដែលពីមុនត្រូវចំណាយ ពេល២ ឬ ៣ សប្តាហ៍ក្នុងការធ្វើការងារដំឡើងដើម្បីផលិតឡានមួយគ្រឿង នេះបង្ហាញអោយ ឃើញពីប្រសិទ្ធិ ភាព ដើម្បីឆ្លើយតបទៅនឹងសមាហរណកម្មនៃការបញ្ជាទិញ។

**ឃ. លទ្ធិកម្រិត**

E-commerce បានជួយជំរុញអ្នកផ្គត់ផ្គង់ចូលប្រឡូកខ្លួនទៅក្នុងទីផ្សារសាកលមួយផ្នែក ដែរ។ E-commerce បានផ្តល់ឱកាសអោយការធ្វើពាណិជ្ជកម្មចូលខ្លួនទៅបំរើអតិថិជនយ៉ាងច្រើន ពេញ ពិភពលោកដែលបណ្តាញ Internet គ្របដណ្តប់។ ផលិតផល និងសេវាត្រូវបានលក់ជូន អតិថិជន គ្រប់តំបន់លើសកលលោក ដែលជំរុញអោយធុរកិច្ចស្រូបទាញនូវអតិថិជនគោលដៅដ៏ច្រើន។ ទាំងនេះជាសក្តានុពលយ៉ាងសំខាន់ក្នុងការប្រកួតប្រជែងការលក់នៅលើទីផ្សារពិភពលោក។

**១.១.២.៣. ចំពោះសង្គមជាតិ**

ការរីកចំរើននៃបច្ចេកវិទ្យាអេឡិចត្រូនិកពាណិជ្ជកម្ម ធ្វើអោយរដ្ឋាភិបាលងាយស្រួលក្នុងការ គ្រប់គ្រងព័ត៌មានវិនិយោគ ស្ថានភាពទំនិញ និងអាត្រាពន្ធបានល្អប្រសើរ។ ចរន្តនៃសេដ្ឋកិច្ចមាន ដំណើរការល្អនាំអោយសង្គមមានការរីកចំរើន។ សក្តានុពលនៃ E-Commerce ធ្វើអោយប្រជាជន ទទួលបាននូវការរស់នៅប្រកបដោយភាពពេញចិត្ត សុវត្ថិភាព សប្បាយរីករាយលើការចាយវាយ

ក្នុងរស់នៅដែលនាំអោយមានសុខដុមនីយកម្ម កំនើនសេដ្ឋកិច្ច និងធ្វើអោយប្រទេសជាតិរីកចម្រើន។

**១.១.៣. E-Commerce នៅក្នុងប្រទេសកម្ពុជា**

ប្រទេសកម្ពុជាគឺជាប្រទេសដែលកំពុងអភិវឌ្ឍន៍ ដែលមានខ្សែនសេដ្ឋកិច្ចរីកចម្រើនក្នុងអាត្រា កំនើន១០ភាគរយពីឆ្នាំ២០០៤ដល់ឆ្នាំ២០០៧ ដែលកំនើននេះទទួលបានមកពីការរីកលូតលាស់ពី វិស័យកាត់ដេរ ការសាងសង់ កសិកម្ម និងទេសចរណ៍ផងដែរ។ ក៏ប៉ុន្តែសេដ្ឋកិច្ចមានការថយចុះ មកវិញនៅឆ្នាំ២០០៨ ៧ភាគរយ បើធៀបទៅនឹងឆ្នាំចាស់ និងបន្តថយចុះ ២ភាគរយទៀតនៅ ឆ្នាំ ២០០៩ ដោយសារការជះឥទ្ធិពលពីវិបត្តិសេដ្ឋកិច្ច ពិភពលោក។

ជាការពិតប្រទេសកម្ពុជា គឺជាប្រទេសដែលមានការស្វាគមន៍ចំពោះវិនិយោគិនបរទេស ដែលមានបំណងចូលមកវិនិយោគលើគ្រប់វិស័យ។ ក្រៅពីមានគោលនយោបាយទាក់ទាញការ វិនិយោគពីបរទេស កម្ពុជាក៏អាចធ្វើពាណិជ្ជកម្មបានតាមរយៈ Ecommerce ដែលជាវិធីនៃការធ្វើ ពាណិជ្ជកម្មថ្មីមួយរបស់កម្ពុជាតាមរយៈប្រព័ន្ធុInternet។ រូបភាព E-Commerce មួយបែបទៀតគឺ ការចំណាយប្រើប្រាស់ Visa Card ឬ Credit Card, ក៏ប៉ុន្តែ នៅក្នុងស្ថានភាពវិបត្តិសេដ្ឋកិច្ច Ecommerce ហាក់ដូចមិនមានការរីកចម្រើនខ្លាំងនោះទេ បើទោះជាធនាគារក្នុងស្រុកមានការចេញ អោយប្រើប្រាស់នូវ Visa Card ឬ Credit Card ក្នុងការទូទាត់ចំណាយតាមរយៈ Internet ក៏ដោយ។ កត្តាធំៗបីដែលជាឧបសគ្គ រាំងស្ទះចំពោះ Ecommerce មានដូចជា៖

- ១. ហេដ្ឋារចនាសម្ព័ន្ធ Internet មិនគ្រប់គ្រាន់
- ២. ការយល់ដឹង និងចំនួនអ្នកប្រើប្រាស់ Internet នៅមានកំរិត
- ៣. ចំនួនអ្នកប្រើប្រាស់ Credit Card ក្នុងការទិញទំនិញមានចំនួនតិចតួច

ផ្ទុយទៅវិញចាប់ពីដើមឆ្នាំ២០១០រហូតមកដល់បច្ចុប្បន្ន E-Commerce ហាក់ដូចជាដើរតួរ យ៉ាងសំខាន់មួយផ្នែកនៅលើទីផ្សារ Online ដែលប្រជាជនកម្ពុជាប្រើប្រាស់ Internet ក៏ដូចជា Website ក្នុងការបញ្ជាទិញ (Order) ឬដាក់លក់ផលិតផលទាំងឡាយនៅលើទីផ្សារ Online នៅលើ Internet។ ការមានវត្តមានឡើងវិញនៃ E-Commerce នៅកម្ពុជា គឺមកពីការ យល់ដឹងបានច្រើន និងកាន់តែច្បាស់ពី E-Commerce ម្យ៉ាងប្រជាជនភាគច្រើនបានយល់ពីដំណើរការនៃ E-Commerce ត្រង់ថាវាបានផ្តល់ភាពងាយស្រួលដល់ស្តង់ដាររស់នៅរបស់ពួកគេបានច្រើនជាងមុន។

ទិដ្ឋភាពទូទៅរបស់ E-Commerce នៅកម្ពុជា គឺបង្ហាញយ៉ាងជាក់ច្បាស់ត្រង់ថាការជួបគ្នា រវាងអ្នកលក់ និងអ្នកទិញអាចប្រព្រឹត្តទៅបានតាមរយៈ Website មួយចំនួនដែលធ្វើការបង្កើតឡើង ក្នុងគោលដៅសំរាប់ដាក់លក់ផលិតផល (សំរាប់អ្នកដែលមានបំណងចង់លក់) និង សំរាប់ស្វែងរកទិញផលិតផល(សំរាប់អ្នកមានបំណងចង់ទិញ)។ ជាក់ស្តែងជាឧទាហរណ៍ គេហទំព័រ <http://www.khmer24.com> ជាគេហទំព័រ Online Shopping មួយដ៏ល្អ ដែលបង្កើតឡើងដើម្បីអោយអ្នកលក់ និងអ្នកទិញបង្ហោះដោយសេរី និងមិនគិតថ្លៃសេវាកម្ម រាល់ផលិតផល ទាំងឡាយដូចជា ទូរស័ព្ទដៃ កុំព្យូទ័រ រថយន្ត ម៉ូតូ និងផលិតផលដទៃទៀត។ ប្រសិនបើផលិតផលមួយប្រភេទត្រូវបានចរចារគ្រឹមត្រូវ និងមានការពេញចិត្តពីសំណាក់ភាគីអ្នកលក់ និងអ្នកទិញ ភាគី ទាំងពីរនឹងធ្វើការណាត់ជួបគ្នាដើម្បីមើលពីគុណភាពផលិតផល និងដើម្បីបង់ប្រាក់សំរាប់ផលិត ផលនោះ ព្រោះថាប្រជាជនខ្មែរភាគច្រើនមិននិយមប្រើ Credit Card ក្នុងការទូទាត់ការបង់ប្រាក់ នោះទេ។ ជាទូទៅចំពោះការទិញផលិតផលក្នុង ស្រុកតាមរយៈ online គឺប្រជាជនមិននិយមប្រើ Credit Card នោះទេ ព្រោះពួកគាត់មានគំនិតមួយមិនច្បាស់ ត្រង់ថាតើគេអាចនឹងយកទំនិញទៅ ជូនគាត់ ឬអត់នៅពេលគាត់បង់ប្រាក់រួច។ ឧទាហរណ៍ ជាក់ស្តែងមួយទៀត អ្នកលក់នៅស្រុកខ្មែរនាពេលបច្ចុប្បន្នចាប់ផ្តើមទាញយកផលចំណេញ របស់ខ្លួនតាមរយៈបណ្តាញទំនាក់ទំនងសង្គម Facebook គឺពួកគាត់បង្កើតជា online shop ដែលពួកគាត់អាច Post នូវប្រភេទផលិតផលរបស់ខ្លួនមានលក់នៅលើ Profile Photos របស់ពួកគេ បន្ទាប់មកបើមានអតិថិជនណាមានចំណាប់អារម្មណ៍ចង់ទិញ អ្នកលក់នឹងបញ្ជូន (Deliver) ផលិតផលទៅជូនអតិថិជនដល់ទីកន្លែង ហើយអ្នកទិញនឹងធ្វើការបង់ប្រាក់ភ្លាមៗនោះតែម្តង។ ចំពោះរូបភាពនៃ E-Commerce តាមរយៈការប្រើប្រាស់ Credit Card វិញ គឺត្រូវបានធ្វើឡើងតាមរយៈស្ថាប័នផ្សេង ជាពិសេស ក្រុមហ៊ុនឯកជនដែលធ្វើពាណិជ្ជកម្មក្នុងស្រុកខ្មែរ ហើយចង់បញ្ជាទិញ(Order) ផលិតផលផ្សេងៗ ពីសំណាក់ក្រុមហ៊ុនក្រៅស្រុក ដែលជាក្រុមហ៊ុនលក់ផលិតផលonline។ ពេលនោះយើងគ្រាន់តែ ចុះឈ្មោះជាមួយនឹងក្រុមហ៊ុន នោះព្រមទាំងផ្តល់លេខID របស់Credit Cardយើង បន្ទាប់មក យើងនឹងអាច Order ផលិតផលនោះបានជាការស្រេច ហើយការដឹកជញ្ជូនគឺធ្វើឡើងតាមរយៈ ក្រុមហ៊ុនក្រៅស្រុក កំណត់ ជាទូទៅពី៥ថ្ងៃ ដល់ ៧ថ្ងៃ។

រឿងដែលគួរឲ្យព្រួយបារម្ភនោះគឺ តើគេនឹងអាចដក់ប្រាក់ប៉ុន្មានពីCredit Card របស់ យើងនិយាយរួមការចំណាយតាម Credit Card ក៏ត្រូវការប្រឈមហានិភ័យខ្លះដែរ។ ដូច្នោះ យើងត្រូវអោយប្រាកដថារវាងអ្នកលក់ និងអ្នកទិញនៅលើទីផ្សារonline គឺជាមនុស្សដែលអាច ទុកចិត្តគ្នា

បានទៅវិញទៅមក ព្រោះថាមិនមានច្បាប់ណាមួយកំណត់ជាក់លាក់ចំពោះដំណើរការនៃការធ្វើ ពាណិជ្ជកម្ម online នោះទេ។



រូបភាពទី៣. ភាពងាយស្រួលក្នុងការបញ្ជាទិញទំនិញ

**១.២. ប្រភេទនៃ E-Commerce**

**១.២.១. ការវិនិយោគនៃ E-Commerce**

មុននឹងឈានមកដល់សព្វថ្ងៃនេះ E-Commerce បានវិវត្តខ្លួនទៅតាមដំណាក់កាលដូចខាង ក្រោម៖

**១.២.១.១. CompuServe is Founded**

ក្នុងឆ្នាំ១៩៦៩ប្រព័ន្ធនៃCompuServe ត្រូវបានស្វែងរកឃើញដោយនិសិត្យវិស្វករអេឡិចត្រូនិច ពីររូបគឺលោក Dr. John R.Goltz and Jeffrey Wilkins នៅក្នុងឆ្នាំ១៩៦៩ដែលជាបច្ចេកវិទ្យាដំបូង ដែលប្រើការតភ្ជាប់តាមរយៈបណ្តាញទំនាក់ទំនង។ នៅក្នុងឆ្នាំ១៩៨០ CompuServe ត្រូវបាន បើកបង្ហាញជាសាធារណៈតំបូងបង្អស់តាមរយៈEmail ហើយនិងការតភ្ជាប់ប្រព័ន្ធអ៊ីនធឺណែតទៅ កាន់សាធារណៈជន និងបានបន្តគ្របដណ្តប់លើទិដ្ឋភាពពាណិជ្ជកម្មតាមប្រព័ន្ធអ៊ីនធឺណែតនៅ ពាក់កណ្តាលឆ្នាំ១៩៩០។

**១.២.១.២. Michael Aldrich បានបង្កើតកម្មវិធីទិញទំនិញតាមរយៈទូរទស្សន៍**

នៅក្នុងឆ្នាំ១៩៧៩ កម្មវិធីទិញទំនិញអ៊ីវ៉ាន់តាមប្រព័ន្ធអេឡិចត្រូនិចត្រូវបានបង្កើតឡើងដោយលោក Michael Aldrichជនជាតិអង់គ្លេសម្នាក់ ដែលបានប្រតិបត្តិការណ៍ការតភ្ជាប់ក្នុងការកែប្រែពីទូរទស្សន៍ ទៅកាន់ប្រព័ន្ធកុំព្យូទ័រតាមរយៈទូរទស្សន៍។ វិធីសាស្ត្រនេះគឺអាចធ្វើទៅបានក្នុងការបើកទូរទស្សន៍

មានឡើងវិញ ចំពោះព័ត៌មានណាដែលបានបិទហើយក៏បានចែករំលែកទៅកាន់មជ្ឈដ្ឋានដទៃ ទៀតសម្រាប់ការបញ្ចូលទិន្នន័យប្រកបដោយសុវត្ថិភាពហើយបច្ចេកវិទ្យានេះផងដែលក៏បាន ក្លាយជាគ្រឹះដ៏រឹងមាំក្នុងការបង្កើតឡើងនេះ។

**១.២.១.៣. Boston Computer បានធ្វើការផ្លាស់ប្តូរ**

Boston Computer ត្រូវបានចាប់ផ្តើមការផ្លាស់ប្តូរនៅក្នុងឆ្នាំ១៩៨២។ វាគឺជាក្រុមហ៊ុនពាណិជ្ជ កម្មអេឡិចត្រូនិចដំបូងបង្អស់នៅក្នុងពិភពលោក។ មុខងារចំបងរបស់របស់ពាណិជ្ជកម្មអេឡិចត្រូ និចនេះគឺដើម្បីបម្រើដល់ទីផ្សារតាមប្រព័ន្ធអ៊ីនធឺណែតសម្រាប់អ្នកដែលចាប់អារម្មណ៍ក្នុងការ លក់នៅកុំព្យូទ័រចាស់ឬក៏ប្រើរូបរបស់ពួកគេ។

**១.២.១.៤. ការដាក់លក់នូវបណ្តុំសៀវភៅជាច្រើនលើទីផ្សារ**

ក្នុងឆ្នាំ១៩៩២ Charles M.stack បានណែនាំបណ្តុំសៀវភៅគ្មានដែនកំណត់នៅប្រព័ន្ធអ៊ីនធឺណែត។ រយៈពេលបីឆ្នាំពេញមុនពេលលោកJeff Bezos បានប្រកាសណែនាំអំពី Amazon.com ដើមឡើយក្រុមហ៊ុនបានប្រើប្រាស់ទម្រង់ ព័ត៌មានតាមទូរស័ព្ទប៉ុន្តែនៅក្នុងឆ្នាំ១៩៩៤ គេហទំព័រ នេះបានផ្លាស់ប្តូរទៅជាប្រព័ន្ធអ៊ីនធឺណែតវិញហើយបានដំនើរការជា Book.com។

**១.២.១.៥. ការបើកដំណើរការនូវកម្មវិធី Netscape Navigator**

ក្នុងឆ្នាំ១៩៩៤ លោក Marc Andreessen និងលោក Jim Clark បានរួមគ្នាបង្កើត Netscape Navigator ដែលជាឧបករណ៍រុករកគេហទំព័រ ហើយបានប្រកាសជាផ្លូវការនូវសេចក្តីណែនាំរបស់ ខ្លួននៅខែតុលាឆ្នាំ ១៩៩៤។ ក្នុងកំឡុងទសវត្សឆ្នាំ ១៩៩០ Netscape Navigator បានក្លាយជាកម្ម វិធីរុករកបណ្តាញដែលត្រូវបានប្រើជាចម្បងនៅលើ Window មុនពេលការរីកចម្រើនរបស់ក្រុម ហ៊ុនយក្សទំនើបដូចជាGoogle ជាដើម។

**១.២.១.៦. ក្រុមហ៊ុន Amazon និង eBayបានបើក**

លោក Jeff Bezos បានណែនាំក្រុមហ៊ុន Amazon នៅឆ្នាំ ១៩៩៥ ជាវេទិកាពាណិជ្ជកម្មអេឡិចត្រូ និចសម្រាប់សៀវភៅ។ នៅឆ្នាំដដែលនោះ Pierre Omidyar បានណែនាំឱ្យដេញថ្លៃ AuctionWeb ដែលក្រោយមក ត្រូវបានគេហៅថា eBay មកដល់សព្វថ្ងៃនេះ។ ចាប់តាំងពីពេលនោះមកអ្នកទាំងពីរបានក្លាយជាវេទិកាលក់អេឡិចត្រូនិចដ៏ធំមួយដែលអាចឱ្យ មានការលក់តាមអ៊ីនធឺណែតដល់អតិថិជននៅជុំវិញពិភពលោក។

**១.២.១.៧. ក្រុមហ៊ុន Paypal បើកដំណើរការទូទាត់ប្រាក់តាមអនឡាញ**

ដើមឡើយត្រូវបានណែនាំជា Confinity ដោយស្ថាបនិក Max Levhin, Peter Thiel, Nosek និង Ken Howery។ PayPal បានបង្ហាញខ្លួននៅលើអ៊ីនធឺណែតនៅចុងឆ្នាំ ១៩៩៨ ជាឧបករណ៍ផ្ទេរប្រា

ក៏។ នៅឆ្នាំ ២០០០ វាបានបញ្ចូលគ្នាជាមួយក្រុមហ៊ុនធនាគារតាមអ៊ិនធឺណែតរបស់ Elon Musk ហើយចាប់ផ្តើមមានកេរ្តិ៍ឈ្មោះនិងភាពល្បីល្បាញតាំងពីពេលនោះមក។

**១.២.១.៨. ក្រុមហ៊ុន Alibaba បើកដំណើរការ**

ក្នុងឆ្នាំ១៩៩៩ Alibaba Online បានបើកដំណើរការនៅឆ្នាំ ១៩៩៩ ជាទីផ្សារទីផ្សារតាមអ៊ិនធឺណែតដែលមានថវិកាជាង ២៥ លានដុល្លារ។ នៅឆ្នាំ ២០០១ ក្រុមហ៊ុនទទួលបានប្រាក់ចំណេញ។ វាបានប្រែក្លាយទៅជាវេទិកា B2B, C2C និង B2C ដែលនៅតែត្រូវបានប្រើប្រាស់យ៉ាងទូលំទូលាយសព្វថ្ងៃនេះ។

**១.២.១.៩. Google ណែនាំ Google Adwords**

Google Adwords ត្រូវបានណែនាំក្នុងឆ្នាំ ២០០០ ជាមធ្យោបាយមួយសម្រាប់ពាណិជ្ជកម្មតាមប្រព័ន្ធអេឡិចត្រូនិចដើម្បីផ្សព្វផ្សាយដល់មនុស្សដោយប្រើឧបករណ៍ស្វែងរកហ្វូហ្គល។ ដោយមានជំនួយពីការចំលងអត្ថបទខ្លីនិងបង្ហាញ URLs អ្នកលក់រាយតាមអ៊ិនធឺណែតបានចាប់ផ្តើមប្រើប្រាស់ឧបករណ៍នេះតាមរយៈ pay-per-click (PPC) context.

**១.២.១.១០. ការណែនាំអោយស្គាល់សមាជិកភាពក្រុមហ៊ុន Amazon Prime**

ក្រុមហ៊ុន Amazon បានណែនាំក្រុមហ៊ុន Amazon Prime ក្នុងឆ្នាំ ២០០៥ ជាមធ្យោបាយមួយសម្រាប់អតិថិជនដើម្បីទទួលបានការដឹកជញ្ជូនដោយឥតគិតថ្លៃរយៈពេលពីរថ្ងៃសម្រាប់ថ្លៃសេវាប្រចាំឆ្នាំផ្ទះល្វែង។ សមាជិកភាពនេះក៏បានរួមបញ្ចូលនូវគុណប្រយោជន៍ផ្សេងទៀតដូចជាការបញ្ចុះតម្លៃលើការដឹកជញ្ជូនមួយថ្ងៃនិងក្រោយមកទៀតអាចទទួលបានសេវាកម្មស្ត្រីមដូចជា Amazon Video និងព្រឹត្តិការណ៍ដែលមានតែសមាជិកដូចជា "Prime Day" ។ ការផ្លាស់ប្តូរយុទ្ធសាស្ត្រនេះបានជួយជំរុញភាពស្មោះត្រង់របស់អតិថិជននិងលើកទឹកចិត្តដល់ការទិញម្តងទៀត។ សព្វថ្ងៃនេះការដឹកជញ្ជូនដោយឥតគិតថ្លៃនិងល្បឿននៃការដឹកជញ្ជូនគឺជាសំណើទូទៅបំផុតពីអ្នកប្រើប្រាស់តាមអ៊ិនធឺណែត។

**១.២.១.១១. Etsy ត្រូវបានបង្កើតឡើង**

Etsy បើកដំណើរការនៅឆ្នាំ ២០០៥ ដែលអនុញ្ញាតឱ្យសិប្បករនិងអ្នកលក់តូចៗលក់ទំនិញតាមរយៈទីផ្សារអនឡាញ។ នេះបាននាំសហគមន៍អ្នកផលិតតាមអ៊ិនធឺណែតពង្រីកការឈានដល់អតិថិជនដែលទិញ ២៤ ម៉ោង។

**១.២.១.១២. ការបើកដំណើរការកម្មវិធី Square**

កម្មវិធី Square ត្រូវបានបង្កើតឡើងក្នុងឆ្នាំ ២០០៩ ដោយលោក Jack Dorsey and Jim McKelvey។ Square app ត្រូវបានបើកបម្រើសេវាកម្មដំបូងបង្អស់នៅក្នុងឆ្នាំ២០១០។ Squareបាន



អនុញ្ញាតឱ្យអ្នកលក់រាយក្រៅបណ្តាញទទួលយកកាតឥណទាននិងកាតឥណទាននៅក្នុងBrick and Mortarរបស់ពួកគេហើយជាការពិតណាស់វាមាននៅគ្រប់ទីកន្លែងជាលើកដំបូងដែលមិនធ្លាប់មានពីមុនមក។ គំនិតនេះបានកើតឡើងចំពោះ Dorsey នៅ ក្នុងឆ្នាំ ២០០៩ នៅពេលដែល McKelvey (ដែលSt. Louis គឺជាមិត្តរបស់ Dorseyអំឡុងពេលនោះ) មិនអាចបញ្ចប់ការលក់ ក្បាលម៉ាស៊ីនទឹកនិងបំពង់បង្ហូរទឹករបស់គាត់ដល់ចំនួនដែលបានកំណត់ ២,០០០ ដុល្លារដោយសារតែគាត់មិនអាចទទួលយកកាតឥណទានបាន។

**១.២.១.១៣. ការបើកដំណើរការរបស់ BigCommerce**

Eddie Machaalani និង Mitchell Harper បានបង្កើតក្រុមហ៊ុន BigCommerce ក្នុងឆ្នាំ ២០០៩ ហើយបានណែនាំវាទៅកាន់មនុស្សគ្រប់គ្នានៅឆ្នាំនោះផងដែរថាជាវេទិកាហាងលក់ផលិតផលអេឡិចត្រូនិកដែលបានដាក់ចេញ ១០០ ភាគរយ។ ចាប់តាំងពីពេលនោះមកការលក់ច្រើនជាង ៨ ពាន់លានដុល្លារត្រូវបានដំណើរការតាមរយៈវេទិកាហើយឥឡូវនេះក្រុមហ៊ុនមានការិយាល័យកណ្តាលនៅអូស្ត្រាលីស្ថិតនៅប្រាស់ស៊ីស្តូនិងស៊ីដនី។ អ្នកផ្តល់វេទិកាបច្ចេកវិទ្យាអេឡិចត្រូនិកផ្សេងទៀតត្រូវបានដាក់ឱ្យដំណើរការនៅក្នុងយុគសម័យតែមួយ។ Shopify (2006) និង Magento (2008) ត្រូវបានគេទទួលស្គាល់ផងដែរថាជាអ្នកដឹកនាំទីផ្សារនៅក្បែរ BigCommerce ។ ឆ្នាំ ២០១៨មគ្គុទេសក៍របស់អ្នកលក់រាយតាមអ៊ិនធឺណែតបាន ណែនាំទៅកាន់វេទិកាពាណិជ្ជកម្មអេឡិចត្រូនិកកំពូលបានមើលឃើញវេទិកាទាំង៣ នេះនៅក្នុងបញ្ជីជាមួយនឹងចំនួនហាងប្រចាំឆ្នាំរបស់ BigCommerce និងចំនួនប្រាក់ចំណូលកើនឡើងនៅលំដាប់ទី ១ ។

**១.២.១.១៤. Google Wallet បានណែនាំវាវិធីសាស្ត្រទូទាត់ឌីជីថល**

ក្នុងឆ្នាំ២០១១ Google Wallet ត្រូវបានណែនាំក្នុងឆ្នាំ ២០១១ ជាសេវាកម្មទូទាត់ប្រាក់ពីម្នាក់ទៅមនុស្សម្នាក់ដែលអាចឱ្យបុគ្គលផ្ញើនិងទទួលប្រាក់ពីឧបករណ៍ចល័តឬកុំព្យូទ័រលើតុ។ តាមរយៈការភ្ជាប់ Wallet ឌីជីថលទៅកាតឥណទានឬគណនីធនាគារអ្នកប្រើអាចបង់ប្រាក់សម្រាប់ផលិតផលឬសេវាកម្មតាមរយៈឧបករណ៍ទាំងនេះ។ សព្វថ្ងៃនេះ Google Wallet បានចូលរួមជាមួយ Android Pay សម្រាប់អ្វីដែលគេហៅថា Google Pay ។

**១.២.១.១៥. ហ្វេសប៊ុកដាក់ចេញនូវការឧបត្ថម្ភ**

ក្នុងឆ្នាំ ២០១១ ហ្វេសប៊ុកបានចាប់ផ្តើមបើកឱកាសផ្សព្វផ្សាយពាណិជ្ជកម្មដល់ម្ចាស់គេហទំព័រតាមរយៈការឧបត្ថម្ភ។ ជាមួយនឹងការចំណាយដើមទុននៅក្នុងយុទ្ធនាការទាំងនេះ អាជីវកម្មអេឡិចត្រូនិកអាចដឹងអំពីចំនួន ទស្សនិកជនជាក់លាក់មួយ ដោយប្រើបណ្តាញសង្គមនិងការទទួលបានព័ត៌មានថ្មីៗពីទស្សនិកជនគោលដៅផ្សេងៗគ្នា។

**១.២.១.១៦. Apple Pay ណែនាំជាវិធីសាស្ត្រទូទាត់តាមទូរស័ព្ទ**

នៅពេលដែលអ្នកទិញវ៉ាន់តាមអ៊ិនធើណែតបានចាប់ផ្តើមប្រើប្រាស់ទូរស័ព្ទរបស់ពួកគេ កាន់តែញឹកញាប់ Apple បានណែនាំ Apple Pay ជាឧបករណ៍ទូទាត់តាមទូរស័ព្ទនិងwallet ឌីជីថលដែលអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់បង់ប្រាក់សម្រាប់ផលិតផលឬសេវាកម្មជាមួយ Apple Pay.

**១.២.១.១៧. គេហទំព័រ Jet.com ចាប់ផ្តើមដំណើរការ**

Jet.com ត្រូវបានបង្កើតឡើងក្នុងឆ្នាំ ២០១៤ ដោយសហគ្រិន Marc Lore (ដែលបានលក់ក្រុមហ៊ុនមុនឈ្មោះ Diapers.com ទៅ Amazon.com) រួមជាមួយ Mike Hanrahan និង Nate Faust ។ ក្រុមហ៊ុននេះប្រកួតប្រជែងជាមួយក្លឹប Costco និង Sam's ដែលផ្តល់ជូនអតិថិជនដែលស្វែងរកតម្លៃទាបបំផុតសម្រាប់ពេលវេលាដឹកជញ្ជូនយូរនិងការបញ្ជាទិញច្រើន។

**១.២.១.១៨. ការទិញទំនិញតាម Instagram ត្រូវបានណែនាំ**

ហាងលក់ទំនិញ Instagram បានចាប់ផ្តើមនៅឆ្នាំ ២០១៧ ដំបូងជាមួយដៃគូពាណិជ្ជកម្មតាមអ៊ីនធើណែត Big Commerce។ ចាប់តាំងពីពេលនោះសេវាកម្មនេះបានពង្រីកដល់វេទិកាពាណិជ្ជកម្មអេឡិចត្រូនិចបន្ថែមនិងអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ Instagram ចុចលើទំនិញភ្លាមៗហើយចូលទៅកាន់គេហទំព័រផលិតផលរបស់ផលិតផលនោះដើម្បីទិញ។



រូបភាពទី៤. ការវិវត្តនៃ E-Commerce

### ១.៣. ប្រភេទនៃ E-Commerce

E-commerce ជាសក្តានុពល និងជាសមទ្វីផលសេដ្ឋកិច្ចដ៏មាន ប្រសិទ្ធភាពដែលកើតចេញ ពី លទ្ធផលនៃការបន្ថយថ្លៃទំនាក់ទំនង ថ្លៃហេដ្ឋារចនាសម្ព័ន្ធ បច្ចេកទេសទាប ក៏ដូចជាភាពងាយ ស្រួលក្នុងការធ្វើប្រតិបត្តិការពាណិជ្ជកម្មតាមបែបអេឡិចត្រូនិច ជាមួយអ្នកផ្គត់ផ្គង់ដែលមានភាព រហ័ស និងសន្សំសំចៃខ្ពស់ រួមជាមួយថ្លៃនៃការចែករំលែកព័ត៌មាន ជាសកល ថ្លៃនៃការផ្សព្វផ្សាយ ពាណិជ្ជកម្មដែលមានកំរិតទាប និងសេវាកម្មបំរើអតិថិជនបែបថ្មី ផងដែរ។ ជាធម្មតាគេបែងចែក E-Commerce ជា៦ប្រភេទ។

#### ១.៣.១. B2B (Business to Business)

នេះជាប្រភេទសំដៅលើក្រុមហ៊ុនដែលធ្វើជំនួញជាមួយគ្នា មានដូចជាក្រុមហ៊ុនផលិតកម្ម លក់ផលិតផលដល់អ្នកបែងចែកលក់បន្ត ឬអ្នកលក់ដុំ លក់បន្តទៅដល់អ្នកលក់រាយ។ ចំពោះ តម្លៃអាស្រ័យលើបរិមាណនៃការបញ្ជាទិញហើយជាទូទៅតម្លៃអាចត្រូវបានចរចាគ្នាបាន។ ឧទាហរណ៍ ប្រភេទអាជីវកម្ម B2B មានដូចជា ExxonMobil និងChevron ដែលសុទ្ធតែជាក្រុមហ៊ុន ខាងផលិតប្រេង និងAlibaba ដែលជាក្រុមហ៊ុនលក់ផលិតផលបោះដុំបំផុតរបស់ពិភពលោក។



រូបភាពទី៥. បង្ហាញពីការធ្វើអាជីវកម្មលក្ខណៈ Business to Business

**១.៣.២. B2C (Business to Consumer)**

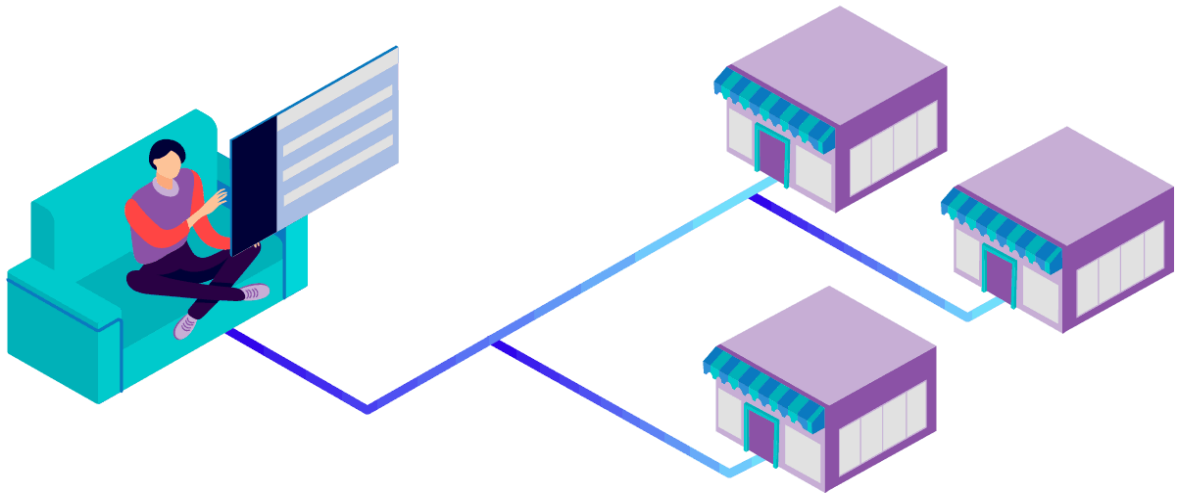
សំដៅលើជំនួញដែលលក់ផលិតផល ឬសេវាកម្មជា សាធារណៈទៅដល់អ្នកប្រើប្រាស់តាមប្រព័ន្ធអ៊ីនធឺណែត ដោយក្នុងនោះគេបានបញ្ចូលនូវរូបភាពគំរូនានាពីផលិតផល និង សេវាកម្មរបស់ខ្លួន។ នេះជាប្រភេទពាណិជ្ជកម្មអេឡិចត្រូនិចដែលគេប្រើប្រាស់ច្រើនជាងគេ ឧទាហរណ៍មានដូចជា Amazon, Taobao និងEbay ជាដើម សូម្បីតែការលក់អនឡាញនៅប្រទេសយើងមួយចំនួនក៏គេអាចចាត់ទុកជា B2C ផងដែរ។



រូបភាពទី៦. បង្ហាញពីការធ្វើអាជីវកម្មលក់ដាច់ដល់អ្នកប្រើប្រាស់: Business to Consumer

**១.៣.៣. C2B (Consumer to Business)**

អ្នកប្រើប្រាស់ផលិតផល ឬសេវាកម្ម អាចបង្ហាញគំរោង របស់ខ្លួនតាមអ៊ីនធឺណែត ដើម្បីស្វែងរកក្រុមហ៊ុនណាមួយមកធ្វើជាម្ចាស់ផ្គត់ផ្គង់គំរោង របស់ខ្លួន។ តាមការធ្វើបែបនេះអ្នកអាចនឹងឃើញក្រុមហ៊ុនមួយចំនួននឹងបង្ហាញខ្លួនក្នុងការអោយអ្នកជ្រើសរើសសំរាប់គំរោងរបស់អ្នក។ បញ្ជាក់: រាល់ទាំងគំរោងសុទ្ធតែបង្ហាញពី គំរោងថវិការចំណាយរបស់គំរោងរបស់អ្នក។ អាជីវកម្មប្រភេទនេះគឺរាងប្លែកពីគេ និងមិនសូវមានអ្នកចាប់អារម្មណ៍ផងដែរ។ បើសិនជាយើងប្រៀបធៀបនិងអាជីវកម្មធម្មតាគឺវាប្រៀបដូចជា ការលក់ពីហាងលក់ដាច់ដល់អ្នកប្រើប្រាស់ក្រុមហ៊ុនធំៗ ហើយអាជីវកម្មប្រភេទនេះទើបតែមានសន្ទុះនារយពេលប៉ុន្មានឆ្នាំចុងក្រោយផងដែរ ដោយសារតែមានការរីកចម្រើនខ្លាំងនូវបច្ចេកវិទ្យា។



រូបភាពទី៧. C2B(Consumer to Business)

**១.៣.៤. B2G (Business to Government)**

ជាទូទៅសំដៅលើពាណិជ្ជកម្មរវាងក្រុមហ៊ុន និងផ្នែក សាធារណៈជន។ វាសំដៅទៅលើការប្រើប្រាស់នៃប្រព័ន្ធអ៊ីនធឺណិតជូនដល់ជាសាធារណៈ ដំណើរការដែលមានច្បាប់ត្រឹមត្រូវ និងដំណើរប្រតិបត្តិការផ្សេងៗទៀតដែលមានរដ្ឋាភិបាលដឹងឮ។ ប្រភេទនៃ E-commerce មួយនេះមានលក្ខណៈពិសេសពីរយ៉ាងគឺ៖ ទី១ ផ្នែកសាធារណៈចេះសន្មត់ទុកជាមុនលើក្បួនដឹកនាំផ្លូវនៅក្នុងការបង្កើតពាណិជ្ជកម្ម និងទី២ វាត្រូវបានសន្មត់ជាមុនថាផ្នែកសាធារណៈមានសេចក្តីត្រូវការដ៏ធំបំផុតសំរាប់ធ្វើអោយប្រព័ន្ធនៃការផ្គត់ផ្គង់កាន់តែមានប្រសិទ្ធភាព។



រូបភាពទី៨. B2G (Business to Government)

**១.៣.៥. C2C ( Consumer to Consumer )**

តាមរយៈ E-commerce ប្រភេទនេះ បុគ្គលម្នាក់ៗអាច ទិញ ឬលក់ឡើងវិញរបស់ដែលខ្លួន មាន តាមOnline យ៉ាងងាយស្រួល។ មាន Website ជាច្រើនដែលអ្នកអាចផ្សព្វផ្សាយរបស់អ្វីដែល អ្នកចង់លក់ ឬក៏ចង់មើលរបស់អ្វីដែលអ្នកចង់ទិញផងដែរ។ ជាឧទាហរណ៍ សុំសាកល្បងសេវាកម្ម ការលក់ និងទិញទំនិញឡើងវិញរបស់ វិបសាយ <http://www.eBay.com>។



រូបភាពទី៩. C2C (Consumer to Consumer)

**១.៣.៦. Mobile Commerce**

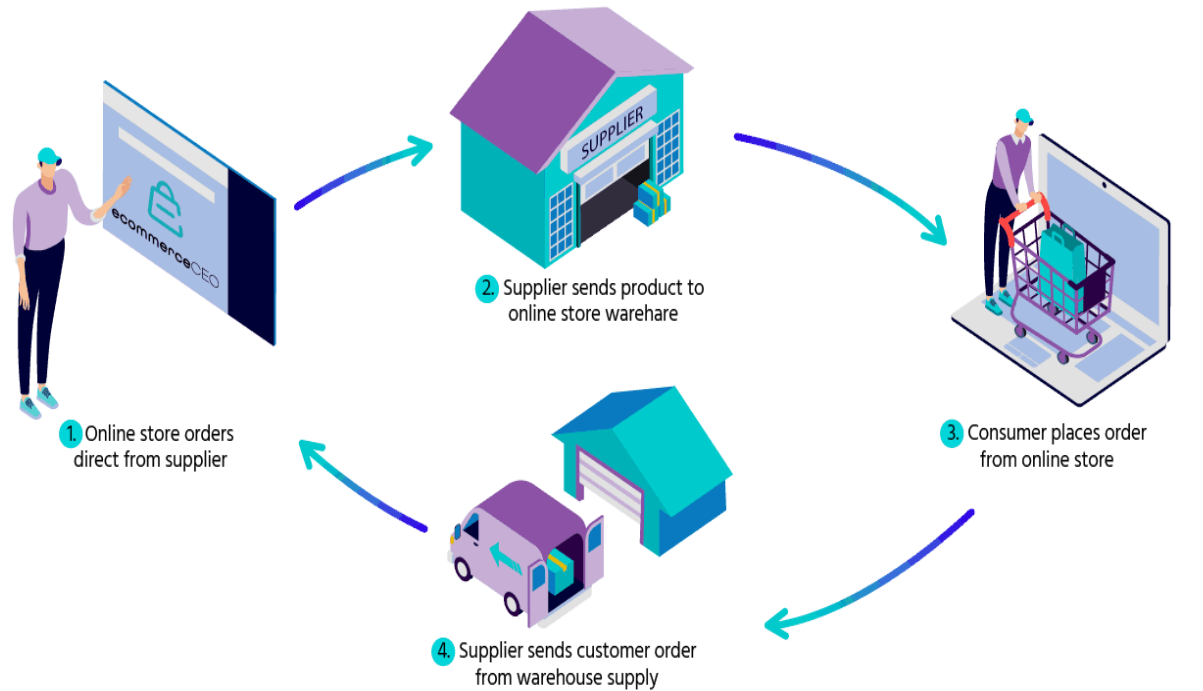
គឺជាសមត្ថភាពដើម្បីធ្វើអោយ ពាណិជ្ជកម្មមានដំណើរការដោយតាមរយៈការប្រើប្រាស់ ឧបករណ៍ចល័តនានាដូចជា ទូរស័ព្ទចល័ត PDA (Personal Digital Assistant), smartphone និង dashtop mobile ជាដើម។ និយាយរួមមក M-commerce គឺជាប្រតិបត្តិការទាំងឡាយណាដែល ពាក់ព័ន្ធ ការផ្ទេរនៃភាពជាម្ចាស់កម្មសិទ្ធិ វិសិទ្ធិក្នុងការប្រើប្រាស់ទំនិញ ឬសេវាកម្មដែលត្រូវបាន បំពេញដោយការប្រើប្រាស់ mobile access ទៅនឹង networks ដែលភ្ជាប់ជាមួយ កុំព្យូទ័រ ដោយ មានការចូលរួមជួយពីឧបករណ៍អេឡិចត្រូនិចណាមួយ។





១.៤.២. Wholesaling and Warehousing

ការលក់បែបនេះត្រូវចំណាយដើមទុនធំបន្តិច ហើយទាមទារចំណេះដឹងខាងផ្នែកគ្រប់គ្រងស្តុក គ្រប់គ្រងទីតាំងសម្រាប់ទុកទំនិញ និងទំនួលខុសត្រូវចំពោះទំនិញនៅពេលបញ្ជូនទៅអោយភ្ញៀវ។ ជំហានដំបូងម្ចាស់គេហទំព័រត្រូវទិញទំនិញពីនាក់ផ្គត់ផ្គង់ ហើយផ្ទុកនៅក្នុងឃ្នាំងរបស់ខ្លួន។ បន្ទាប់មកដាក់តាំងទំនិញទាំងនោះលើគេហទំព័រ នៅពេលដែលអតិថិជនធ្វើការទិញទំនិញនោះ ម្ចាស់គេហទំព័រត្រូវដឹកជញ្ជូនទៅកាន់អតិថិជនផ្ទាល់។



រូបភាពទី១២. បង្ហាញពី Wholesaling and Warehousing

១.៥. យុទ្ធសាស្ត្រ

នៅក្នុងវិស័យធុរកិច្ចអាជីវកម្មទាំងឡាយមិនអាចចៀសផុតពីការផ្សព្វផ្សាយបានឡើយបើទោះបីជាផលិតផលឬសេវាកម្មរបស់យើងមានគុណភាពល្អយ៉ាងណាក៏ដោយ។ ការផ្សព្វផ្សាយមានន័យថាជាការផ្តល់ដំណឹងឬប្រាប់ដល់អតិថិជនពីអ្វីដែលអ្នកមានជាពិសេសគឺធ្វើឲ្យគេស្គាល់ថាអ្នកលក់ផលិតផលឬបម្រើសេវាកម្មអ្វីខ្លះ។

ផ្ទុយមកវិញមនុស្សភាគច្រើនទុកចិត្តលើផលិតផលឬសេវាកម្មដែលគេធ្លាប់ប្រើតែប៉ុណ្ណោះ។ ចុះបើយើងជាម្ចាស់អាជីវកម្មថ្មី តើយើងគួរធ្វើបែបណាដើម្បីឲ្យគេងាកមកប្រើផលិតផលឬសេវាកម្មរបស់យើង? យ៉ាងណាមិញចំពោះអាជីវកម្មថ្មី កញ្ចប់ថវិកាដែលពួកគេមានសម្រាប់ផ្គត់ផ្គង់ទៅលើការផ្សព្វផ្សាយគឺសុទ្ធតែមានកំណត់ឬមានតិចតួចណាស់។



**១.៥.១. រៀបចំគម្រោងក្នុងការធ្វើទីផ្សារជាមុន**

គម្រោងដើម្បីអនុវត្តក្នុងការធ្វើយុទ្ធសាស្ត្រទីផ្សារគឺជាផែនទីសម្រាប់នាំផ្លូវយើង។ យើងត្រូវមានគម្រោងមួយឬច្រើនអាស្រ័យលើវិធីដែលអ្នកចង់ប្រើ។ វានឹងបង្ហាញប្រាប់យើងថាវិធីសាស្ត្រដែលយើងប្រើមានប្រសិទ្ធភាពទេមុននឹងឈានដល់ការអនុវត្តជាពិសេសយើងមិនត្រូវភ្លេចត្រៀមផែនការណ៍បម្រុងឬផែនការណ៍ផ្សេងដើម្បីប្តូរប្រសិនបើវាមិនមានប្រសិទ្ធភាពនោះ។ គម្រោងរបស់យើងគួរដាក់បញ្ចូលលក្ខខណ្ឌមួយចំនួនដូចជា

- ថវិកាដែលអ្នកនឹងចំណាយ
- កំណត់ថាអតិថិជនគោលដៅរបស់អ្នកជាអ្នកណា
- ទីតាំងដែលអតិថិជនរបស់អ្នកស្ថិតនៅ
- ច្រកផ្សព្វផ្សាយដែលអ្នកចង់ប្រើ (បណ្តាញសង្គម គេហទំព័រ ដោយផ្ទាល់ ទូរទស្សន៍...)
- ផែនការណ៍សង្គ្រោះឬផែនការណ៍ផ្សេង
- ជ្រើសរើសបុគ្គលិកដែលមានជំនាញច្បាស់លាស់
- ស្វែងរកដៃគូសហការ។

**១.៥.២. មានភាគសម្គាល់ពាណិជ្ជកម្ម**

ការធ្វើភាគសម្គាល់ពាណិជ្ជកម្មដែលមាននិមិត្តសញ្ញា ឬម៉ាកសញ្ញាត្រឹមត្រូវជាការបង្ហាញថាយើងមានអាជីវកម្មមួយដែលមានលក្ខណៈពិតប្រាកដហើយមានវិជ្ជាជីវៈត្រឹមត្រូវ។ ក្រៅពីនេះយើងត្រូវមាន Letterhead សន្លឹកប័ណ្ណ(Leaflet) សៀវភៅព័ត៌មានជាដើមដែលវាមានភ្ជាប់ជាមួយនិមិត្តសញ្ញាក្រុមហ៊ុន ឈ្មោះនិងតួនាទីរបស់ អាស័យដ្ឋាន លេខទូរស័ព្ទ អ៊ីមែល និងសេវាកម្មឬផលិតផលរបស់យើង។ យើងអាចប្រើភាគប្តូរត្តុទាំងនោះដើម្បីចែកទៅកាន់សាច់ញាតិ បងប្អូនមិត្តភក្តិ ឬអ្នកដទៃទៀតឲ្យគេស្គាល់ពីអាជីវកម្មរបស់យើង និងអាចពឹងគេឲ្យជួយប្រាប់ដល់អ្នកដទៃតបើគេត្រូវការផលិតផលឬសេវាកម្មនោះដែរ។

**១.៥.៣. ផលិតផលមានគុណភាព (Product)**

ផលិតផលត្រូវបានគេមើលឃើញថាជាតម្រូវការរបស់អ្នកប្រើប្រាស់។ វាអាចជាទំនិញរូបិយឬសេវាកម្ម។ ផលិតផលនីមួយៗតែងតែមានវដ្តជីវិតរបស់វា រួមមានកំណើនថេរ លែងស្ងួតពេញនិយម និងនៅទីបញ្ចប់ការលក់ថយចុះឈានរហូតដល់លែងមានតម្រូវការ។ ដូច្នេះ បណ្តាអ្នកនាំទំនិញទៅកាន់ផ្សារ គួរសិក្សាទីផ្សារឲ្យបានល្អិតល្អន់ទៅលើអាយុកាល នៃវដ្តផលិតផល។ តើទំនិញឬសេវាកម្មរបស់ខ្លួនជាទំនិញឬសេវាកម្មដែលមនុស្សគ្រប់រូបតែងតែត្រូវការជាចាំបាច់?

**១.៥.៤. តម្លៃ (Price)**

ទំហំថវិកា ដែលអតិថិជនត្រូវតបស្ម័ងសម្រាប់ផលិតផលឬសេវាកម្មមានសារៈសំខាន់ណាស់ ក្នុងការកំណត់ប្រាក់ចំណេញរបស់ក្រុមហ៊ុន និងរយៈពេលឈរជើង។ ការប្រែប្រួលតម្លៃមានឥទ្ធិពលខ្លាំងណាស់ទៅលើយុទ្ធសាស្ត្រទីផ្សារ និងអាស្រ័យលើភាពបត់បែននៃតម្លៃផលិតផល ហើយការជះឥទ្ធិពលទៅលើតម្រូវការនិងការលក់ផងដែរ។ យុទ្ធសាស្ត្រកំណត់តម្លៃផ្អែកលើកត្តាសំខាន់ៗជាមូលដ្ឋាន៖ ការពិនិត្យមើលតម្លៃប្រកួតប្រជែងទីផ្សារ តម្លៃវាយលុកទីផ្សារ និងតម្លៃសមរម្យអាចទទួលយកបាន។

**១.៥.៥. ការផ្សាយយោសនា (Promotion)**

មធ្យោបាយផ្សាយយោសនាត្រូវបានប្រើប្រាស់ដើម្បីផ្តល់ព័ត៌មានឲ្យ អតិថិជនគោលដៅបានយល់ និងបង្ហាញផលិតផល និងសេវាកម្ម។ មធ្យោបាយទាំងនោះរួមមាន៖ ការផ្សាយពាណិជ្ជកម្ម ទំនាក់ទំនងសាធារណៈ ការរៀបចំលក់ និងកម្មវិធីលក់ពិសេស។ ការយោសនាអាចធ្វើទៅបានតាមប្រព័ន្ធផ្សព្វផ្សាយ អ៊ីនធឺណេត ផ្លាកបាណូ កម្មវិធីតាំងពិពណ៌ និងព្រឹត្តិការណ៍ផ្សេងៗ។ ជាពិសេសក្នុងបរិបទនេះគឺគេផ្តោចយកចិត្តទុកដាក់លើបណ្តាញសង្គមខ្លាំងបំផុត។ ក្រៅពីនេះ ការដែលអតិថិជនបានប្រើប្រាស់ផលិតផលលោកអ្នកហើយពេញចិត្ត រួចណែនាំផ្ទាល់មាត់ទៅដល់អ្នកប្រើប្រាស់ដទៃទៀត ក៏ចាត់ទុកជាការផ្សាយយោសនាដែរ ហើយថែមទាំងមានប្រសិទ្ធភាពខ្ពស់ថែមទៀតផង។ ក្រៅពីនេះ ការបញ្ចុះបញ្ចូលរបស់បុគ្គលិកផ្នែកលក់ដើរតួនាទីយ៉ាងសំខាន់សម្រាប់ ទំនាក់ទំនងសាធារណៈ។

**១.៥.៦. ការចែកចាយ (Placement)**

ការចែកចាយគឺឆ្ពោះទៅយកលទ្ធភាពទទួលបានផលិតផលឬភាពងាយស្រួលសម្រាប់អតិថិជនក្នុងការរកទិញផលិតផល។ យុទ្ធសាស្ត្រចែកចាយ ដែលពេញនិយមជាងគេគឺការចែកចាយទៅកាន់ដេប៉ូឬឃ្នកលក់រាយលក់បន្ត។ យុទ្ធសាស្ត្រធ្វើយ៉ាងណាដើម្បីអោយសាធារណជនបានមើលឃើញពីផលិតផលរបស់យើងបានគ្រប់ទីកន្លែងទាំងអស់។ ការធ្វើបែបនេះគឺជាការសម្រួលដល់អតិថិជនផងដែរ។

## ជំពូកទី២

### វេយប៊ែកអិល

#### ២.១. វេយប៊ែកអិលដែលពាក់ព័ន្ធនឹងការបង្កើតកម្មវិធីទូរស័ព្ទ

##### ២.១.១. អ្វីជា HTML?

HTML គឺកើតចេញពីពាក្យ Hyper-Text Markup Language ដែលវាជាភាសាគ្រឹះ និងសាមញ្ញបំផុតក្នុងការបង្កើតគេហទំព័រ។ ម៉្យាងវិញទៀត គឺជាភាសាមួយដែលត្រូវបានគេបង្កើតឡើងដោយសរសេរនៅក្នុង Notepad ឬ Adobe Dreamweaver ជាដើម ហើយយើងបើកវាតាម Browser ដើម្បីបង្ហាញព័ត៌មាននៃវេយប៊ែកអិលនោះលើគេហទំព័រអោយអ្នកប្រើប្រាស់បានឃើញ។ HTML ជាភាសាគំរូ និង ជាភាសាមានប្រសិទ្ធភាពខ្ពស់ ដែលត្រូវបានគេ ប្រើប្រាស់យ៉ាង ទូលំទូលាយ ដើម្បីពណ័នា អំពី គេហទំព័រ និង នៅតែ ត្រូវបាន គេប្រើប្រាស់ សំរាប់ អន្តរកាល នៅក្នុង គេហទំព័រ។ នៅក្នុងភាសា HTML មានចែកចេញជា Head, Body និង Footer។ នៅក្នុង Head យើងអាចបញ្ចូល File Script មួយចំនួនដូចជា CSS, JavaScript, ហើយ HTML មាន Extension.html ។

ឧទាហរណ៍៖

```

<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>
</body>
</html>

```

# HTML



រូបភាពទី១៣. បង្ហាញពី HTML5

## ២.១.២. អ្វីជា CSS?

CSS គឺជាអក្សររបស់Cascading Style Sheets។ គេប្រើវាក្នុងការតុបតែងទៅលើកូដ HTML និងជាជំនួយក្នុងការរចនា Website ដើម្បីឲ្យ Website កាន់តែមានភាពទាក់ទាញ និងស្រស់ស្អាត ហើយវាអាចឲ្យយើងរចនាទៅលើកូដដូចជា៖ ពណ៌, រូបភាព, អក្សរ ជាដើម។ ហើយវាផ្តល់លក្ខណៈងាយស្រួលដល់អ្នកបង្កើត Website ក្នុងការបង្កើត ឬកែប្រែរូបរាងរបស់វេបសាយទាំងមូលដោយមិនចាំបាច់ធ្វើ ការកែប្រែទៅលើគេហទំព័រនីមួយៗទេ គេគ្រាន់តែធ្វើការកែសម្រួកទៅលើ CSS Fileនោះវានឹងធ្វើការផ្លាស់ប្តូរទៅលើគេហទំព័រទាំងមូលតែម្តង។

ឧទាហរណ៍៖

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```



រូបភាពទី១៥. បង្ហាញពី CSS

**២.១.៣. អ្វីជា JavaScript ?**

Javascript ជាភាសាកុំព្យូទ័រមួយប្រភេទដែលគេសរសេរឡើងសម្រាប់ជួយក្នុងការបង្កើតគេហទំព័រឱ្យមានភាពរស់រវើក (Dynamic) វាមានលក្ខណៈ Client and Server-Side Scripting ដែលត្រូវបានបង្កើតឡើងដោយ Netscape ដែលមាន syntax ភាគច្រើនមកពីភាសា C វាក៏អាចយកមកសរសេរ game ឬកម្មវិធី desktop applications ហើយវាក៏ជាភាសាដែលពេញនិយមបំផុតសម្រាប់បង្កើតគេហទំព័រផងដែរ ... ។ ហើយជាទូទៅវាធ្វើការ (Support) ជាមួយ Browser ដូចជា Internet Explorer, Firefox, Chrome, Opera, និង Safari ។

ឧទាហរណ៍៖

```

<!DOCTYPE html>
<html>
<body>
  <h2>JavaScript Variables</h2>
  <p id="demo"></p>
  <script>
    var price1 = 5;
  
```

```

var price2 = 6;

var total = price1 + price2;

document.getElementById("demo").innerHTML =

"The total is: " + total;

</script>

</body>

</html>

```

JS



រូបភាពទី១៥. បង្ហាញពី JavaScript

### ២.១.៤. អ្វីជា Vue.JS?

Vue.JS គឺជា Framework មួយនៃភាសា Javascript ដែលអាចឲ្យអ្នកប្រើប្រាស់បង្កើត និង រចនាគេហទំព័រ (Client Side) ឲ្យមានភាពទាក់ទាញ និងរស់រវើក។ វាមានភាពងាយស្រួលក្នុងការ សិក្សា មានមុខងារពេញលេញ និងដំណើរការលឿន។ អ្នកអាចចាប់ផ្តើមបង្កើតកម្មវិធីជាមួយ Vue.js បានប្រើសិនបើអ្នកមានចំនេះដឹងគ្រឹះនៃ HTML និង Javascript ដោយគ្រាន់តែចំណាយ ពេលបន្តិចបន្តួចអាននូវគន្លឹះខ្លះៗរបស់ Vue។

ឧទាហរណ៍៖

=>HTML

```
<div id="demo">  
  <p>{{ message }}</p>  
  <button v-on:click="reverseMessage">Reverse Message</button>  
</div>
```

=>JS

```
var demo=new Vue({  
  el: '#app-5',  
  data: {  
    message: 'Hello Vue.js!'  
  },  
  methods:{  
    reverseMessage: function(){  
      this.message= this.message.split(' ').reverse().join(' ')  
    }  
  }  
})
```



រូបភាពទី១៦. បង្ហាញពី Vue.js

### ២.១.៥. អ្វីជា Vuetify ?

Vuetify គឺជា Front-End Responsive Framework ដែលជាកូដមួយកញ្ចប់ដែលត្រូវបានគេចងក្រងឡើងដោយប្រើប្រាស់ភាសានិង សមាសធាតុមួយចំនួនដូចជា Css, JavaScript, Font, ...។ ហើយត្រូវបានគេយកមកប្រើប្រាស់ជាមួយ Vue.js ដើម្បីបង្កើតជា Template ដោយមិនចាំបាច់ប្រើប្រាស់ភាសា Css ក៏បានដែរ ។ Vuetify អាចបង្កើត Website ដែលមានលក្ខណៈ: Responsove (អាចបើកបាននៅក្នុង ទូរស័ព្ទ អាយដេតនិង កុំព្យូទ័រដោយមិនខូចទ្រង់ទ្រាយ) ហើយវាបានបង្កើតជា Components (ធាតុ) ដែលចាំបាច់ប្រើជាច្រើន ហើយប្រសិនបើលោកអ្នកចង់ប្រើប្រាស់នូវ Components ណាមួយគឺគ្រាន់តែហៅឈ្មោះ:Class របស់ Component នោះមកដាក់ក្នុង Elements របស់ HTML រួចជាការស្រេច។

ឧទាហរណ៍៖

```

<!DOCTYPE html>
<html>
<head>
  <link href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900"
rel="stylesheet">
  <link
href="https://cdn.jsdelivr.net/npm/@mdi/font@5.x/css/materialdesignicons.min.css"
rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/vuetify@2.x/dist/vuetify.min.css"
rel="stylesheet">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no, minimal-ui">
</head>
<body>
  <div id="app">
    <v-app>
      <v-main>
        <v-container>
          Hello world <br/>

```



My Name is my name.

```
</v-container>

</v-main>

</v-app>

</div>

<script src="https://cdn.jsdelivr.net/npm/vue@2.x/dist/vue.js"></script>
<script src="https://cdn.jsdelivr.net/npm/vuetify@2.x/dist/vuetify.js"></script>
<script>
new Vue({
  el: '#app',
  vuetify: new Vuetify(),
})
</script>
</body>
</html>
```

# Vuetify

The Path to 2.0



រូបភាពទី១៧. បង្ហាញពី Vuetify

### ២.១.៦. អ្វីជា Dart ?

Dart ជាភាសាសម្រាប់សរសេរកម្រោងលើ Multi Platform។ Dart ត្រូវបានបង្កើតឡើងដោយក្រុមហ៊ុន Google និងត្រូវបានប្រើប្រាស់ដើម្បីបង្កើតកម្មវិធីសម្រាប់ប្រើប្រាស់លើ ទូរស័ព្ទ កុំព្យូទ័រ Server និង វេបសាយ។

ឧទាហរណ៍៖

```
int fib(int n) => (n > 2) ? (fib(n - 1) + fib(n - 2)) : 1;
void main() {
  print('fib(20) = ${fib(20)}');
}
```



# Dart

រូបភាពទី១៨. បង្ហាញពី Dart

### ២.១.៧. អ្វីជា Flutter ?

Flutter ជា UI SDK ដែលបង្កើតដោយ Google។ Flutter ត្រូវបានប្រើប្រាស់ដើម្បីបង្កើតកម្មវិធីសម្រាប់ប្រើប្រាស់លើ Android iOS Linux Mac Windows និង វេបសាយ ដោយសរសេរកូដតែមួយសម្រាប់ ប្រព័ន្ធប្រតិបត្តិការទាំងអស់នេះ។

ឧទាហរណ៍៖

```
import 'package:flutter/material.dart';
void main() => runApp(HelloWorldApp());
class HelloWorldApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

```

title: 'Hello World App',
home: Scaffold(
  appBar: AppBar(
    title: Text('Hello World App'),
  ),
  body: Center(
    child: Text('Hello World'),
  ),);}}

```



រូបភាពទី១៩. បង្ហាញពី Flutter

### ២.១.៧. អ្វីជា Firebase ?

Firebase គឺជា Platform មួយដែលត្រូវបានអភិវឌ្ឍដោយ Google។ Firebase មាន Service ជាច្រើនសម្រាប់ជំនួយក្នុងការបង្កើតកម្មវិធីទូរស័ព្ទ និងវេបសាយ។ Service ទាំងនោះមានដូចជា៖

Cloud Firestore, Firebase ML, Cloud Functions, Authentication, Hosting, Cloud Storage, Realtime Database, Crashlytics, App Distribution, Performance Monitoring, Test Lab, Analytics, Extensions, Predictions, Firebase A/B Testing, Clous Messaging, In-App Messaging, Remote Config, Dynamic Links, App Indexing.

- Cloud Firestore (Store data) ប្រើសម្រាប់ផ្ទុកទិន្នន័យ
- Cloud Storage (Store Image) ប្រើសម្រាប់ផ្ទុក File
- Authentication (Sign in with Email-Password)។



# Firebase

រូបភាពទី២០. បង្ហាញពី Firebase

### ជំពូកទី៣

### ដំណើរការស្រាវជ្រាវ និងកំណត់មុខងារ

ប្រតិបត្តិការអាជីវកម្មតាមប្រព័ន្ធអ៊ីនធឺណែតនេះភាគច្រើនស្ថិតនៅក្នុងវិស័យធំៗទាំង៤រួមមានទេសចរណ៍ ប្រព័ន្ធផ្សព្វផ្សាយ ទូរគមនាគមន៍និងដឹកជញ្ជូន និងពាណិជ្ជកម្មអ៊ីនធឺណែត (E-Commerce)។ តាមរបាយការណ៍រួមគ្នារបស់ក្រុមហ៊ុនទាំង២បានបង្ហាញថា ទំហំទឹកប្រាក់នេះនឹងកើនឡើងដល់២០០លានដុល្លាររងឈ្មោះក្នុងឆ្នាំ២០២៥។

ក្នុងចំណោមអាជីវកម្មជាច្រើន ក្រុមហ៊ុនដូចជា Alibaba និងTencent សុទ្ធតែមានចំណែកយ៉ាងធំនៅក្នុងទឹកប្រាក់សរុបនេះ។ ក្នុងនោះផងដែរ ប្រទេសដូចជា សិង្ហបុរីជាដើម បានមើលឃើញពាណិជ្ជកម្មអ៊ីនធឺណែត ជាវិស័យមួយដែលកំពុងតែមានការរីកដុះដាលយ៉ាងគំហុក យោងលើហេតុផលថា ក្នុងចំណោមអាជីវកម្ម៥ ក្នុងនោះមាន៣ ហើយដែលផ្សារភ្ជាប់ខ្លួនជាមួយប្រព័ន្ធអ៊ីនធឺណែត។ ភាពជោគជ័យនេះ គឺដោយសារតែលើសពី៨០%នៃជនជាតិសិង្ហបុរីសុទ្ធតែមានអ៊ីនធឺណែតប្រើប្រាស់។

ដោយឃើញពីសក្តានុពលយ៉ាងខ្លាំងខ្លាចបែបនេះ ទើបអាស៊ានបានសម្រេចចិត្តចេញនូវគោលការណ៍ ក្នុងគោលបំណងដើម្បីពង្រឹង និងលើកទឹកចិត្តទៅដល់អាជីវកម្មតូចធំក្នុងតំបន់អោយពង្រីកខ្លួន ក្រោមការជួយជ្រោមជ្រែងដោយបង្កើតច្បាប់ពង្រឹងទំនាក់ទំនងតាមប្រព័ន្ធឌីជីថល និងកាត់បន្ថយរបាំងសេដ្ឋកិច្ចតាមដែលអាចធ្វើបាន។

➢ ការបង់ប្រាក់ពេលទំនិញដល់ដៃ (Cash on Delivery)

ថ្វីត្បិតបើប្រជាជនមួយចំនួនធំបានយល់ដឹងពីការទិញទំនិញតាមរយៈអនឡាញហើយក៏ពិមែន ប៉ុន្តែភាគច្រើននៃប្រជាជនក្នុងតំបន់អាស៊ីអាគ្នេយ៍មិនសូវទំលាប់ក្នុងការបង់ប្រាក់តាមកាតឥណទាន ឬតាមប្រព័ន្ធធនាគារឌីជីថលប៉ុន្មាននោះទេ ពោលគឺថាពួកគេនិយមបង់ប្រាក់នៅពេលដែលទំនិញទៅដល់ដៃរបស់ពួកគេច្រើនជាង។

➢ ទីផ្សារថ្មីភាគច្រើនមានការលូតលាស់ខ្លាំងចំពោះការធ្វើទីផ្សារតាមប្រព័ន្ធទូរស័ព្ទ

ទីផ្សារដែលទើបនឹងចាប់ផ្តើមលូតលាស់ដូចជាទីផ្សារក្នុងប្រទេសឡាវដែលបានទទួលការជោគជ័យគួរអោយកត់សម្គាល់តាមរយៈការប្រើប្រាស់យុទ្ធសាស្ត្រទីផ្សារដើម្បីទាក់ទាញអតិថិជនតាមរយៈការផ្សព្វផ្សាយព័ត៌មានផ្សេងៗចូលទៅក្នុងទូរស័ព្ទរបស់អតិថិជនតែម្តង ព្រោះអតិថិជនមីយ៉ាន់ម៉ាសុទ្ធតែប្រើស្មាសហ្វូន។

➢ ការបម្លែងអោយចេញជាការលក់នៅតិចនៅឡើយ

ការបង្កើតអ្នកដែលចាប់អារម្មណ៍ពីផលិតផលមួយនៅលើប្រព័ន្ធអ៊ីនធឺណែត អោយក្លាយទៅជាអ្នកទិញនៅមានអត្រាទាបនៅឡើយ ដោយក្នុងនោះប្រទេសដែលខអន់ជាងគេបំផុតគឺប្រទេសវៀតណាម ដែលមានត្រឹមតែ១.៣%ប៉ុណ្ណោះ។ វាមានន័យថា ក្នុងចំណោមមនុស្ស១០០០នាក់ នាក់ដែលចូលមកមើលផលិតផលនោះ មានតែ១៣នាក់ប៉ុណ្ណោះដែលសម្រេចចិត្តទិញ។

➢ **អាជីវកម្មខ្នាតតូចងាយស្រួលក្នុងការចូលរួមក្នុងវិស័យពាណិជ្ជកម្មអ៊ីនធឺណែត**

កម្មវិធីដូចជា Khmer24 ជាដើមបានផ្តល់លក្ខណៈងាយស្រួលជាខ្លាំងទៅដល់បណ្តាអាជីវកម្មខ្នាតតូច ងាយស្រួលក្នុងការចូលរួមវិស័យពាណិជ្ជកម្មអ៊ីនធឺណែត ដោយសារតែវាអាចអោយពួកគេចូលទៅបង្កើត និងលក់ទំនិញបានដោយខ្លួនរបស់ពួកគេដោយប្រាក់តិចបំផុត។

➢ **ទម្លាប់នៃការទិញទំនិញរបស់អតិថិជន**

ជាទូទៅ សម្រាប់អតិថិជនដែលទិញសម្ភារៈដែលមានតម្លៃថ្លៃ ពួកគេច្រើនចូលតាមរយៈកុំព្យូទ័រច្រើនជាងទូរស័ព្ទដែរ។ ម៉ោងដែលគេនិយមទិញគឺចាប់ពីម៉ោង៩ព្រឹក ដល់ម៉ោង៥ល្ងាចប៉ុណ្ណោះ ដោយក្នុងនោះមានដូចជាប្រទេសឥណ្ឌូនេស៊ី ម៉ាឡេស៊ី ថៃ ហ្វីលីពីន និងវៀតណាម។ ឯប្រទេសសិង្ហបុរី ម៉ោងដែលប្រជាជនទិញទំនិញថ្លៃៗគឺម៉ោង១០យប់។

➢ **កាចូលរួមពីវិនិយោគិនចិន**

ក្រុមហ៊ុនធំៗដូចជាអាឡីបាបា(Alibaba) និងចេនសេន(Tencent) នៅតែបន្តបណ្តាក់ទុនយ៉ាងសម្បូរទៅដល់អាជីវកម្មតាមប្រព័ន្ធអ៊ីនធឺណែតនៅក្នុងតំបន់អាស៊ីអាគ្នេយ៍។

**៣.១. ការវិនិយោគដើម្បីជ្រើសរើសប្រធានបទ**

ការជ្រើសរើសប្រធានបទគឺធ្វើតាមតម្រូវការរបស់អ្នកប្រើប្រាស់ អោយស្របតាមយុគសម័យ Technology និងកាត់បន្ថយភាពលំបាក ព្រមទាំងបង្កើនភាពងាយស្រួលក្នុងការទិញទំនិញអោយមានភាពងាយស្រួលព្រមទាំងស្របតាមចំណុចខាងលើ។

**៣.២. ការវិនិយោគបម្រើបម្រាស់កម្មវិធីទូរស័ព្ទ**

ដើម្បីបង្ហាញពីបម្រើបម្រាស់ប្រព័ន្ធ ឬទំនាក់ទំនងរវាងអ្នកប្រើប្រាស់និងមុខងារ យើងសូមលើកយក User Diagram មកបង្ហាញ។

**៣.២.១. ការកំណត់មុខងារអ្នកប្រើប្រាស់**

បន្ទាប់ពីបានសិក្សាពីតម្រូវការអ្នកប្រើប្រាស់រួចមក យើងខ្ញុំបានកំណត់នូវមុខងារសម្រាប់អ្នកប្រើប្រាស់ដូចខាងក្រោម៖

- ❖ អ្នកគ្រប់គ្រងទិន្នន័យទាំងមូល (Administration)
  - Product Management (aproved, add, edit, update, delete)

- User Management
- Catagory Management

❖ អ្នកប្រើប្រាស់ (User)

- Sign in as Anonymous (View only)
- Sign in with Account
  - Product Management (add, edit, update, delete)
  - Sell Management
  - Buy Management
  - Profile Management (add, edit, update, delete)
  - Chat with any account
  - Shopping cart
- Account Management
  - Sign up
  - Change password

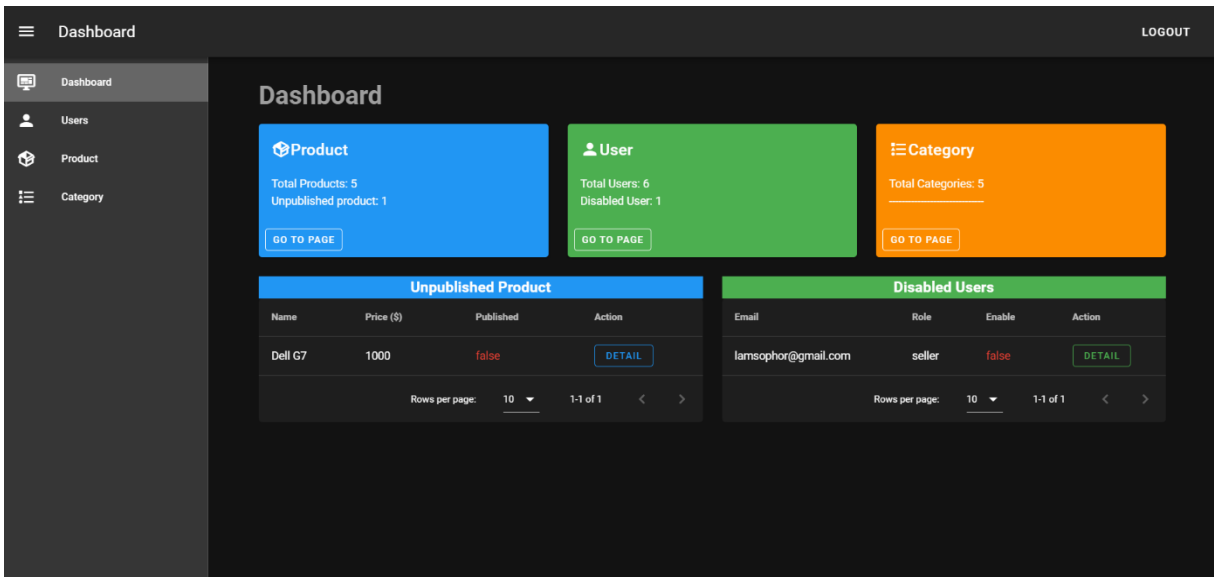
**៣.៣. ការវិភាគលើរចនាសម្ព័ន្ធ Mobile app**

ការវិភាគលើរចនាសម្ព័ន្ធនៃ Mobile App គឺជាចំណុចសំខាន់ ព្រោះការគ្រប់គ្រង Mobile App ដែលប្រកបដោយប្រសិទ្ធភាព គឺផ្តើមចេញពីការសិក្សាលើរចនាសម្ព័ន្ធនៃ Mobile App<sup>1</sup> ដូច្នោះហើយទើបយើងខ្ញុំបានសិក្សា និងការវិភាគលើរចនាសម្ព័ន្ធនៃការបញ្ជារលក់ និងទិញទំនិញអោយសមស្របនឹងតម្រូវការនៃ Mobile App របស់យើងខ្ញុំដែលមានរៀបរាប់ដូចខាងក្រោម៖

**៣.៣.១. ការកំណត់នៅ Entities នៅក្នុង Page Admin**

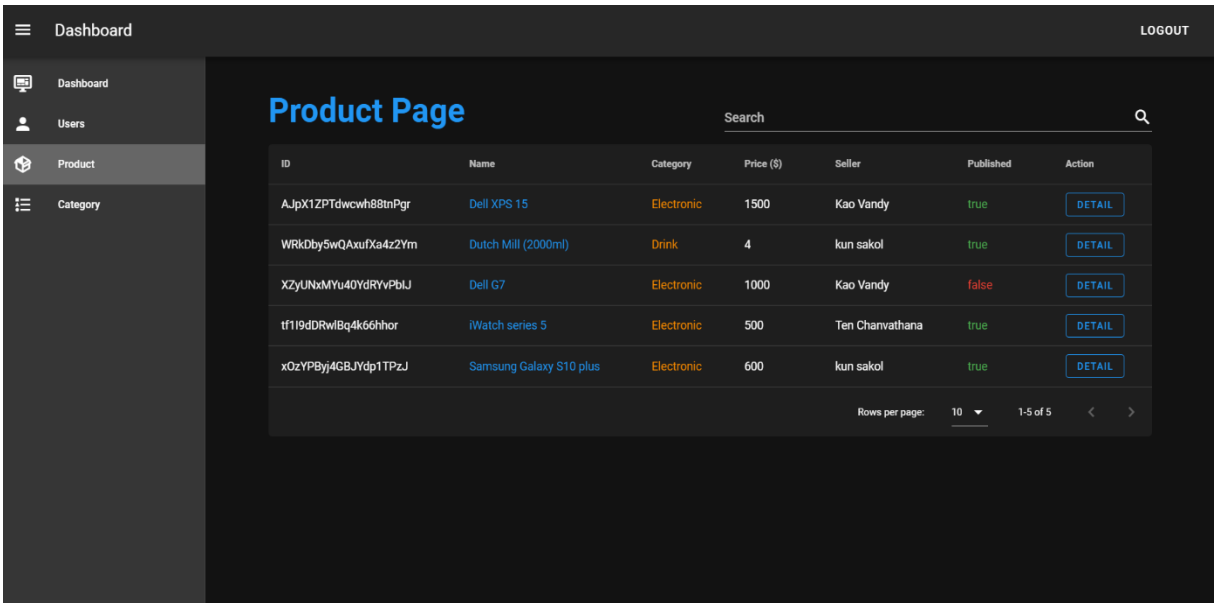
ស្របតាមការវិភាគលើ Mobile App បានយ៉ាងល្អិតល្អន់ យើងខ្ញុំបានកំណត់នូវទិន្នន័យសំខាន់ៗដូចខាងក្រោម៖

- Dashboard: បង្ហាញពីចំនួនផលិតផលទាំងអស់ ចំនួនផលិតផលដែលមិនអនុញ្ញាតឱ្យបង្ហាញ ចំនួនអ្នកប្រើប្រាស់ ចំនួនអ្នកប្រើប្រាស់ដែលត្រូវបានបិទគណនេយ្យ ចំនួន Category សរុប និងមានតារាងចំនួនពីរគឺ តារាងបង្ហាញពីផលិតផលដែលមិនទាន់អនុញ្ញាតឱ្យបង្ហាញលើ Mobile App និងតារាងបង្ហាញគណនេយ្យដែលត្រូវបានបិទ



រូបភាពទី២១. បង្ហាញពី Dashboard លើ Page Admin

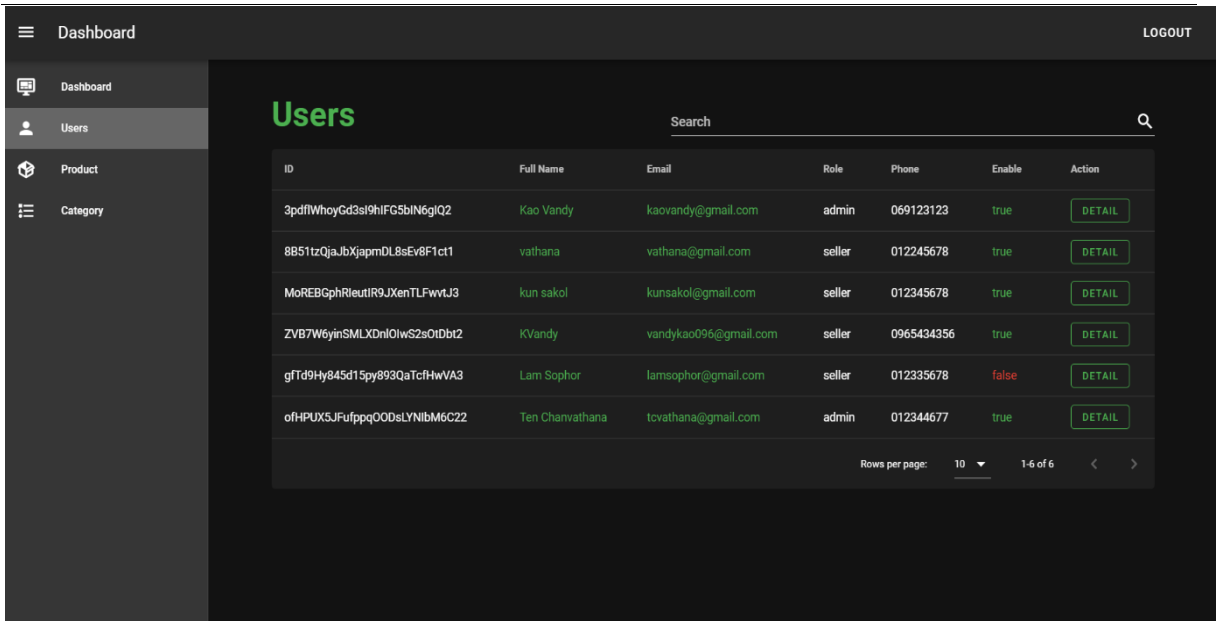
- ការគ្រប់គ្រងទំនិញ(Product Management)៖ រាល់ទំនិញទាំងអស់មុននឹងអាចផុសដាក់លក់បាន លុះត្រាតែមានការត្រួតពិនិត្យពី admin សិនទើបអាចដាក់លក់បាន។



រូបភាពទី២២. បង្ហាញពី Product លើ Page Admin

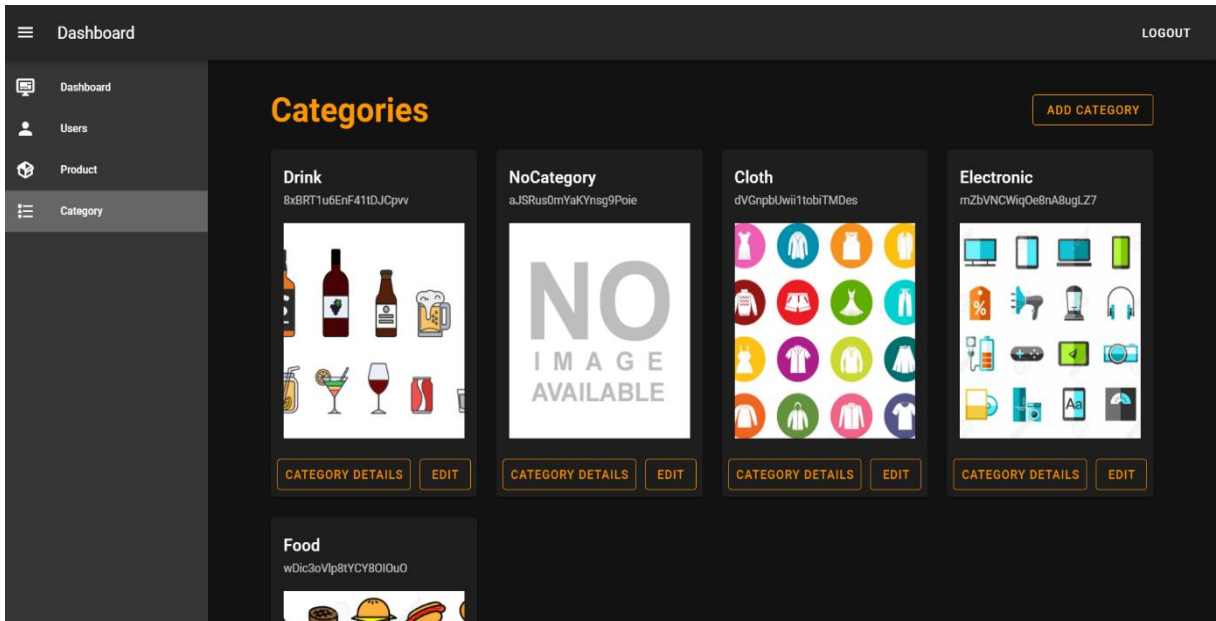
- ការគ្រប់គ្រងទៅលើអ្នកប្រើប្រាស់(User Management)៖ អាចធ្វើការ Enable ឬ Disable User កុំអោយផុសលក់ផលិតផល។





រូបភាពទី២៣. បង្ហាញពី User លើ Page Admin

- ការគ្រប់គ្រងទៅលើប្រភេទនៃទំនិញ(Category Management)៖ អាចធ្វើការAddបន្ថែម កែប្រែ និងលុបបន្ធូរប្រភេទនៃទំនិញបាន។



រូបភាពទី២៤. បង្ហាញពី Category លើ Page Admin

### ៣.៣.២. ការកំណត់នៅ Entities នៅក្នុង Page User

Page User ក៏ជាផ្នែកមួយយ៉ាងសំខាន់ដែលនៅក្នុង Mobile Appរបស់យើង ក្នុងនោះយើងបានបែងចែកនូវមុខងារដូចជា៖

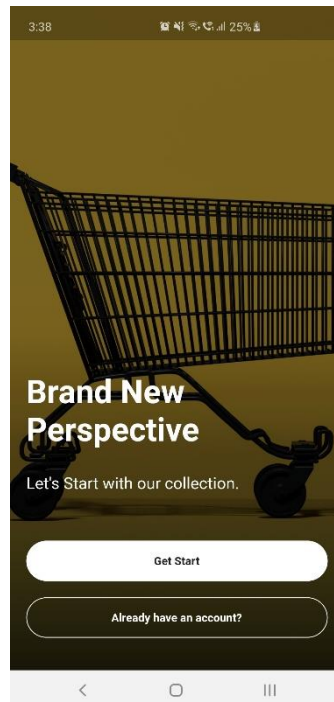
- ការចុះឈ្មោះមិនទាន់ពេញសិទ្ធិ(Sign in as Anonymous)៖ ក្នុងនោះផងដែរ អ្នកប្រើប្រាស់ដែលបានបង្កើត ឬចុះឈ្មោះមិនទាន់ពេញសិទ្ធិ(Sign in as Anonymous) មានសិទ្ធិបានត្រឹមតែមើលលើមុខទំនិញដែលបានដាក់លក់ប៉ុណ្ណោះ។

- ការចុះឈ្មោះពេញសិទ្ធិ(Sign in with Account)៖ អ្នកប្រើប្រាស់អាចធ្វើការAdd, Edit, Update, និង Delete ទៅលើមុខទំនិញរបស់ខ្លួន អាចគ្រប់គ្រងទៅលើការលក់ ការទិញ Profile របស់ខ្លួន(Add, Edit, Update, Delete) អាចធ្វើការផ្ញើជាMessageទៅគណនេយ្យផ្សេងៗបានផលដែល ជាពិសេសអាចធ្វើការSaveមុខទំនិញដែលខ្លួនចង់ទិញទុកទិញនៅពេលក្រោយបានផងដែរ។

- ការគ្រប់គ្រងទៅលើគណនេយ្យ(Account Management)៖ User អាចធ្វើការបំពេញព័ត៌មានផ្ទាល់ខ្លួន ជាមួយនឹងលេខសំដាត់សម្រាប់ធ្វើការ Login។ User ក៏អាចធ្វើការប្តូរលេខសំដាត់របស់ខ្លួនបានផងដែរ។

**៣.៣.៣. សម្មតិកម្មរបស់ Entity Page**

**៣.៣.៣.១. Get Start Page**

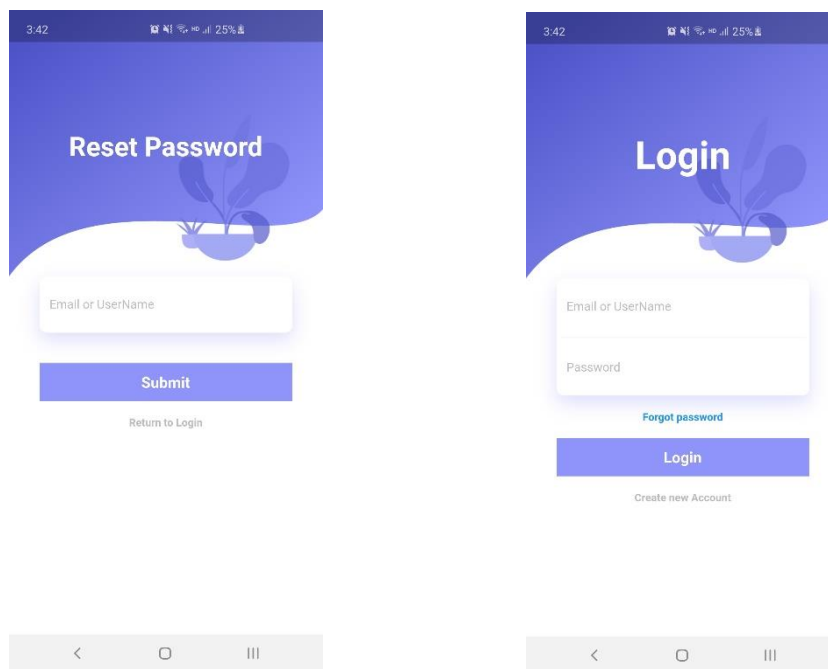


រូបភាពទី២៥. បង្ហាញពី Page Get Start

Get Start Page គឺជាដើមទំព័រនៃកម្មវិធីដែលមានពីរជម្រើសសម្រាប់អ្នកប្រើប្រាស់ជ្រើសរើស៖

- Button Get Start សម្រាប់អោយ User Sign in as Anonymous គឺមានន័យថា អ្នកប្រើប្រាស់អាចមានសិទ្ធិត្រឹមមើលតែប៉ុណ្ណោះ រាល់ Funtion ផ្សេងៗមិនត្រូវបានអនុញ្ញាតអោយប្រើប្រាស់នោះទេ។
- Button Already have an account សម្រាប់អោយអ្នកប្រើប្រាស់ធ្វើការ Sign in ឬ Sign up ដើម្បីអាចប្រើប្រាស់បាននូវ Full Funtion របស់ App.

៣.៣.៣.២. Log In Page



រូបភាពទី២៦. បង្ហាញពី Login Page និងReset Password page

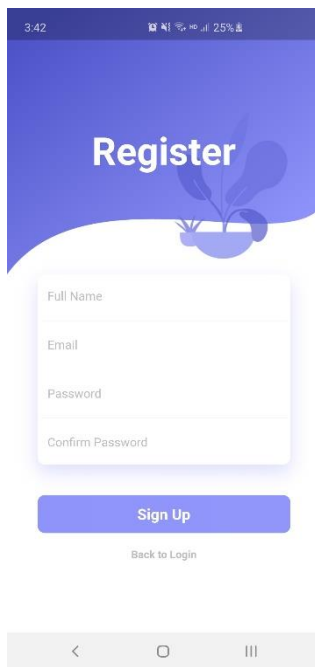
Log In Page គឺជាជម្រកមួយសម្រាប់អោយអ្នកប្រើប្រាស់ធ្វើការ Log in នូវគណនេយ្យផ្ទាល់ខ្លួនរបស់ពួកគេ។ វាមានមុខងារជាច្រើនដូចជា៖

- ប្រអប់ទាំងពីរខាងលើគឺសម្រាប់អោយអ្នកប្រើប្រាស់បញ្ចូលនូវ Email និងPassword ដែលខ្លួនបានធ្វើការបង្កើត។
- Button Login សម្រាប់ធ្វើការ Login និងធ្វើការផ្ទៀងផ្ទាត់ ប្រសិនបើបញ្ចូលមិនត្រឹមត្រូវវានឹងអោយយើងធ្វើការបញ្ចូលម្តងទៀត។
- Create new Account សម្រាប់អោយអ្នកប្រើប្រាស់ដែលមិនទាន់មានគណនេយ្យផ្ទាល់ខ្លួនធ្វើការបង្កើតផងដែរ។

- Forgot Password សម្រាប់អោយអ្នកប្រើប្រាស់ធ្វើការ Reset Password តាមរយៈ Email ក្នុងករណីបាត់បង់ណាមួយ ដោយគ្រាន់តែបញ្ជូលនូវ Email នោះវានឹង Verify ផ្ទាល់ទៅកាន់ Email របស់យើង រួចធ្វើការដាក់នូវ Password ថ្មីជាការស្រេច។

**៣.៣.៣. Register Page**

Register Page គឺសម្រាប់អោយ User ធ្វើការ Register ដោយបំពេញនូវឈ្មោះ Email និង Password ដើម្បីធ្វើការបង្កើតគណនេយ្យថ្មី។ បន្ទាប់ពីបានធ្វើការ Register រួចមក UserអាចLogin ចូលប្រើប្រាស់បានហើយ។ យើងក៏អាចធ្វើការReset Password បានតាមរយៈ Email បានផង ដែលនៅក្នុងករណីភ្លេចឬក៏បាត់លេខសំងាត់។



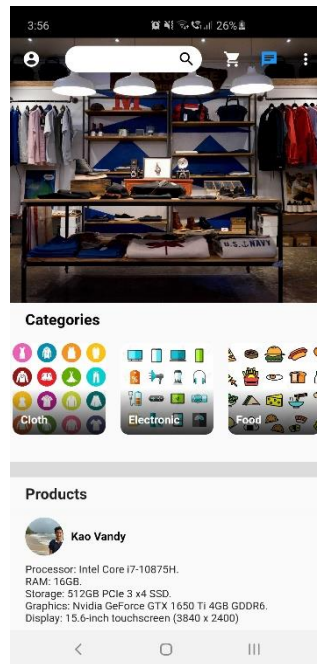
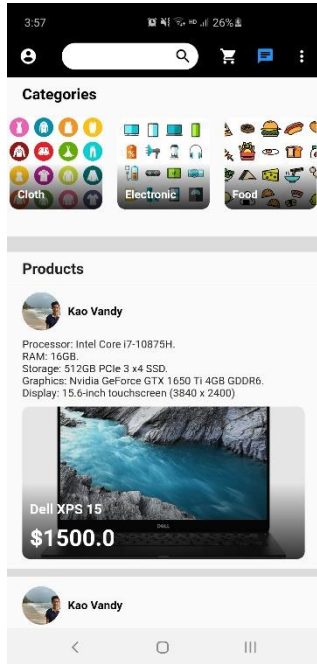
រូបភាពទី២៧. បង្ហាញពី Register Page

**៣.៣.៤. Home Page**

Home page គឺជាទំព័រដើមនៃកម្មវិធីដែលវាមាននូវមុខងារជាច្រើនដូចជា៖

- Profile គឺសម្រាប់អោយ User ចូលមើលព័ត៌មានផ្ទាល់ ក្នុងនោះផងដែលគេក៏អាចធ្វើការ Upload នូវ Profile Picture, Cover Picture, និងព័ត៌មានផ្ទាល់ខ្លួន ដ៏ដូចជាអាស័យដ្ឋានបាន
- Search គឺសម្រាប់អោយ User ធ្វើការស្វែងរកមុខទំនិញដែលខ្លួនចង់បាន ដោយគ្រាន់តែវាយនូវឈ្មោះមុខទំនិញជាការស្រេច
- Cart គឺសម្រាប់អោយ User ធ្វើការរក្សាទុកនូវទំនិញសម្រាប់មើល ឬក៏ទិញពេលក្រោយ យើងក៏អាចធ្វើការ Remove បានផងដែល

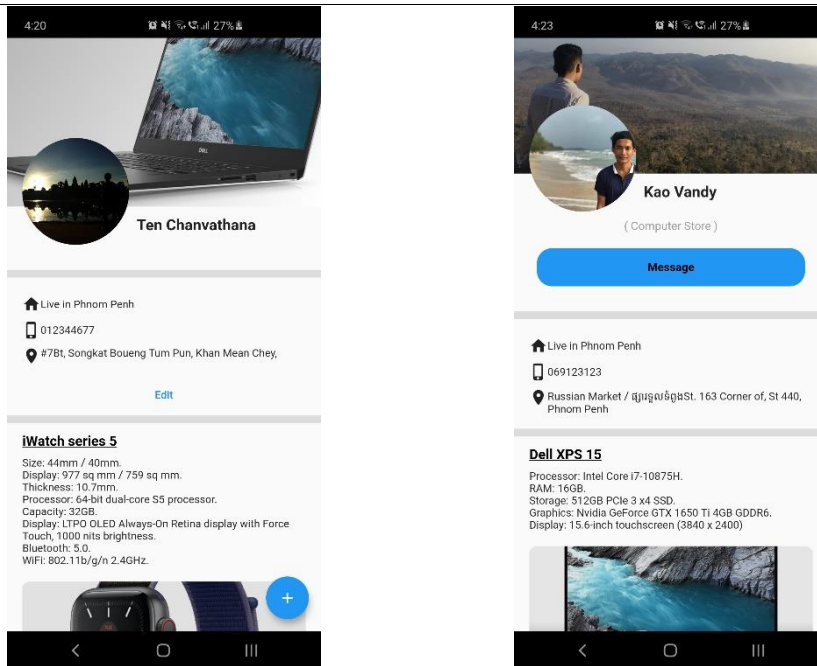
- Chat គឺសម្រាប់អោយ User ធ្វើការសន្ទនា
- Log out គឺសម្រាប់អោយ User Log out និងធ្វើការ Sign in new account
- Categories ការបែងចែកទំនិញទៅតាមប្រភេទ
- Product for Sale
- Item detail



រូបភាពទី២៨. បង្ហាញពី Home Page

៣.៣.៣.៥. Profile Page

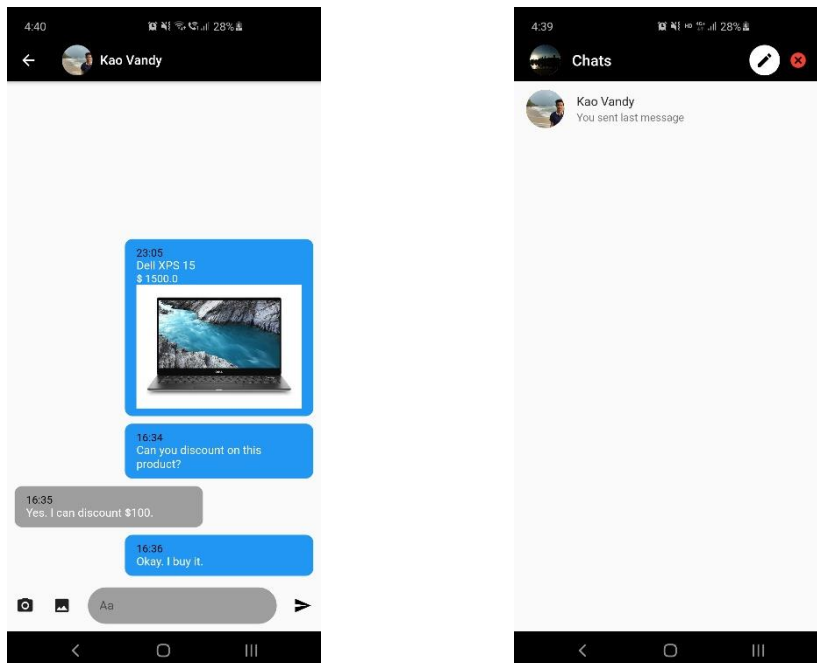
Profile Page គឺជាជេកសម្រាប់បង្ហាញព័ត៌មានលំអិតរបស់អ្នកប្រើប្រាស់មានដូចជា ឈ្មោះ អាស័យដ្ឋាន ទីកន្លែងរស់នៅ លេខទូរស័ព្ទ និងផលិតផលសម្រាប់លក់នៅខាងក្រោមនោះផង ដែល យើងក៏អាចធ្វើការ Upload រូបភាពជាក់លើ Profile ឬCover និងអាចធ្វើការកែប្រែនូវព័ត៌មាន ផ្ទាល់ខ្លួនបានផងដែរ។ ហើយប្រសិនបើអ្នកចង់បន្ថែមផលិតផលថ្មី គឺគ្រាន់តែចុចសញ្ញាបូករួច បំពេញព័ត៌មានលំអិតនៃផលិតផលជាការស្រេច បញ្ជាក់រាល់ផលិតទាំងអស់អាចលក់ជា Public បានលុះត្រាតែមានការអនុញ្ញាតពី Admin។



រូបភាពទី២៩. បង្ហាញពី Profile Page

៣.៣.៣.៦. Chat Page

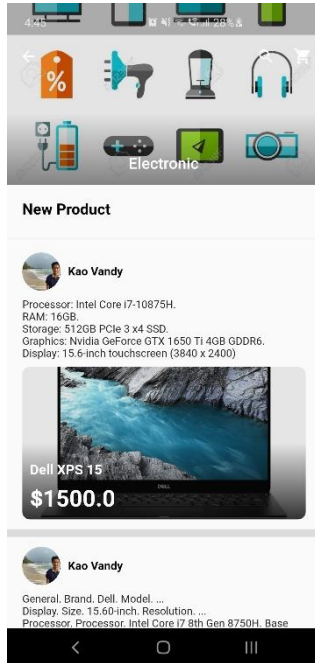
Chat Page គឺបង្ហាញពីការសន្ទនារបស់ User។ User អាចធ្វើការសន្ទនាគ្នាជាមួយ User ដទៃ ទៀតក្នុងការដោះដូរទំនិញ និងជជែកកំសាន្ត គេក៏អាចផ្ញើររូបភាព ថតរូបភាពបានផងដែរ ។



រូបភាពទី៣០. បង្ហាញពី Chat Page

៣.៣.៣.៧. Category Page

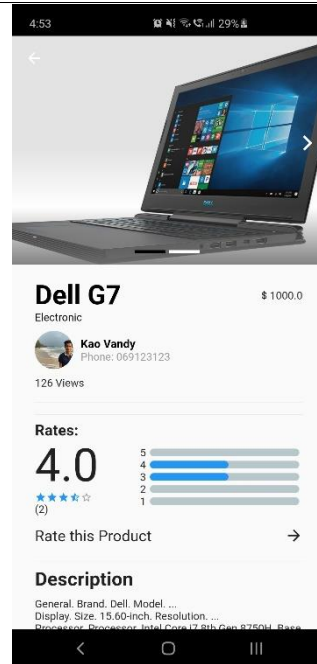
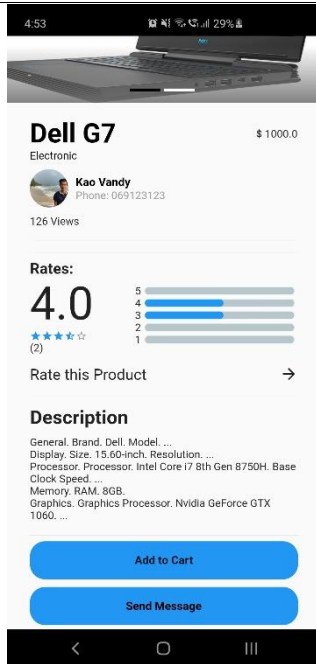
យើងបានបែងចែកទំនិញទៅតាមប្រភេទរបស់វារៀងៗខ្លួន ដើម្បីអោយអតិថិជនងាយស្រួលក្នុងការស្វែងរក ដោយគ្រាន់តែចុចនូវប្រភេទទំនិញដែលខ្លួនចង់រកជាការស្រេច។ ក្នុងនោះផងដែរយើងបានរៀបចំនូវប្រភេទទំនិញជាច្រើនសម្រាប់អតិថិជនមើលផងដែរ។



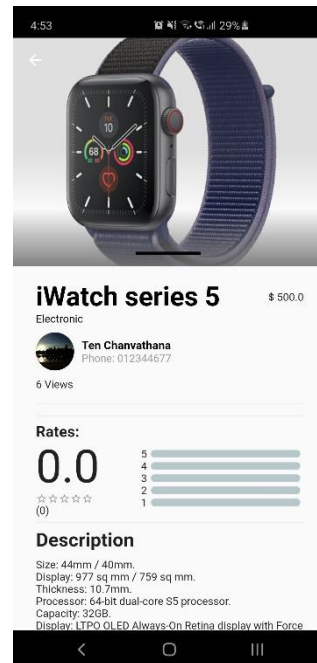
រូបភាពទី៣១. បង្ហាញពី Category Page

៣.៣.៣.៨. Detail Page

Detail Page គឺជាជេកសម្រាប់បង្ហាញពីព័ត៌មានលម្អិតរបស់ផលិតផល ឈ្មោះអ្នកលក់ ទីតាំងហាង និងតម្លៃជាដើម។ យើងក៏អាចធ្វើការរក្សាទុកសម្រាប់មើលនិងទិញថ្ងៃក្រោយក៏បានប្រសិនបើយើងពេញចិត្ត យើងក៏អាចឆ្លាតនិងធ្វើការកម្មង់ភ្លាមៗបានផងដែរ ពិសេសយើងអាចធ្វើការវាយតម្លៃផលិតផលរបស់យើងបានផងដែរ តាមរយៈការអោយផ្កាយ (Rate and Review) ។ យើងក៏អាចធ្វើការកែប្រែ ដាក់រូបចូលបន្ថែមបានផងដែរប្រសិនបើផលិតផលនោះជាផលិតផលដែលយើងជាអ្នកដាក់លក់។



រូបភាពទី៣២. បង្ហាញពី Details ទំនិញនៅហាងដទៃ



រូបភាពទី៣៣. បង្ហាញពី Detail ផ្ទាល់ខ្លួន



## ជំពូកទី៤

### ការរចនាលើ Database

#### ៤.១. ការកំណត់ឈ្មោះរបស់ Collection

ដោយផ្អែកលើ Entity នៅជំពូកទី៣ យើងកំណត់ឈ្មោះតារាងដូចខាងក្រោម៖

Collection	Description
Users	ព័ត៌មានរបស់ User
Category	បំណែងចែកទំនិញ
Products	ព័ត៌មានរបស់ផលិតផល
Cart	ការរក្សាទុកផលិតផល
Chats	កត់ត្រាការសន្ទនា

#### ៤.២. ការពន្យល់អំពី Data Dictionary

Data Dictionary ជាផ្នែកដែលគេប្រើប្រាស់ដើម្បីធ្វើការបកស្រាយលើ Entity និង Attribute ។ ដើម្បីផ្តល់ភាពងាយស្រួលដល់ការសិក្សានិងអ្នកអានកាន់តែយល់អំពី តារាងនីមួយៗព្រមទាំងទទួលបានការលំអិតអំពី គុណភាពនៃទិន្នន័យដូចខាងក្រោម៖

Users		
Field	Type	Description
uid	String	User ID
fullName	String	ឈ្មោះ User
email	String	អ៊ីមែល
phone	String	លេខទូរស័ព្ទ
role	String	មុខងាររបស់ User
enable	Boolean	Enable User
province	String	ខេត្តរស់នៅរបស់ User
address	String	អាសយដ្ឋានរបស់ User
profiles	Array	រូបភាព Profile
covers	Array	រូបភាព Cover

Category		
Field	Type	Description
documentID	Reference	Category ID
categoryName	String	ឈ្មោះ : Category
imageUrl	String	Link រូបភាពតំណាង Category

Products		
Field	Type	Description
name	String	ឈ្មោះផលិតផល
price	Number	តម្លៃផលិតផល
category	Reference	Category របស់ផលិតផល
seller	Reference	ម្ចាស់ផលិតផល
description	String	ព័ត៌មានលំអិតអំពីផលិតផល
imageUrls	Array	រូបភាពផលិតផល
rates	Array	បញ្ជីអ្នក Rate និង Review
views	Number	ចំនួនចុចមើលផលិតផល
createdAt	Timestamp	ពេលវេលាបញ្ចូលផលិតផល
published	Boolean	ដែលអាចឲ្យគេឃើញបាន

Cart		
Field	Type	Description
documentID	Reference	ID របស់ User
products	Array	បញ្ជីផលិតផលក្នុង Cart

Chats		
Field	Type	Description
documentID	Reference	Chat ID
createdAt	timestamp	ពេលវេលាដែលបង្កើត Chat

userIds	Array	បញ្ជី User ដែលនៅក្នុង Chat
messages	Collection	សារឆ្លងឆ្លើយរវាង User

## **សេចក្តីសន្និដ្ឋាន និងការផ្តល់អនុសាសន៍**

### **១. សេចក្តីសន្និដ្ឋាន**

បន្ទាប់ពីបានឃើញនូវតម្រូវការរបស់មនុស្សក្នុងសង្គមទៅលើវិស័យពាណិជ្ជកម្មអេឡិចត្រូនិច ហើយបានធ្វើការស្រាវជ្រាវលើប្រព័ន្ធ និងបង្កើតកម្មវិធីទូរស័ព្ទ E-Commerce រួច ដែលចំណាយរយៈពេល១ខែកន្លះលើការសរសេរកូដ និងកន្លះខែលើការកែសម្រួល ដែលជាចុងក្រោយយើងទទួលបានកម្មវិធីទូរស័ព្ទមួយ ដែលអាចអោយអ្នកអតិថិជនធ្វើការ Download យកទៅប្រើប្រាស់ក្នុងជំនួញពាណិជ្ជកម្មរបស់គេបាន។ ស្របជាមួយភាពជោគជ័យមួយនេះ កម្មវិធីទូរស័ព្ទមួយនេះ និងជួយឲ្យម្ចាស់ជំនួញអាចពង្រីកទីផ្សារជាអន្តរជាតិ ផ្តល់ជាដំណឹងឬព័ត៌មាននានាទៅកាន់អតិថិជនកាន់តែឆាប់រហ័សមិនចាំបាច់ចំណាយប្រាក់ទៅលើការផ្សាយពាណិជ្ជកម្ម និងផ្តល់ជាឱកាសការងារសម្រាប់បុគ្គលិកផ្នែកដឹកជញ្ជូន និងសេវាកម្មដែលពាក់ព័ន្ធ។ កម្មវិធីទូរស័ព្ទនេះមិនត្រឹមតែផ្តល់ផលប្រយោជន៍ចំពោះម្ចាស់អាជីវកម្មប៉ុណ្ណោះទេ អ្នកប្រើប្រាស់ក៏អាចទទួលផលប្រយោជន៍ពីកម្មវិធីទូរស័ព្ទនេះដូចគ្នាដែលអ្នកទិញអាចធ្វើការទំនាក់ទំនងជាមួយហាងលក់ទំនិញពីផ្ទះរបស់គេ ឬពីទីកន្លែងណាក៏បាន។ ទោះបីជានេះជាភាពជោគជ័យដែលអាចជួយសម្រួលដល់ការរស់នៅក្នុងសង្គមបច្ចុប្បន្នយ៉ាងសម្បូរវៃបែបរបស់ពួកយើងក៏ពិតមែនតែដោយសារពេលវេលាមានកំណត់រួមជាមួយចំណេះដឹងនៅមានកម្រិត ដែលជាហេតុធ្វើឱ្យពួកយើងយល់ច្បាស់ថាចំណុចខ្វះខាតនោះគឺនៅតែមាននឹងមិនអាចបំពេញនូវសេចក្តីត្រូវការគ្រប់ជ្រុងជ្រោយនៅឡើយនោះទេ ហេតុដូច្នេះយើងខ្ញុំទាំងអស់គ្នានឹងរងចាំរាល់នូវការវិះគន់ដើម្បីស្ថាបនាពីគ្រប់មជ្ឈដ្ឋាន នឹងរីករាយក្នុងការសហការក្នុងការធ្វើឲ្យកាន់តែប្រសើរឡើងថែមទៀត។

### **២. ការផ្តល់អនុសាសន៍**

ដើម្បីឲ្យដំណើរការនៃកម្មវិធីទូរស័ព្ទនេះមានភាពល្អប្រសើរ និងផ្តល់នូវគុណភាពក្នុងការប្រើប្រាស់ឲ្យកាន់តែមានប្រសិទ្ធភាព ក្រុមយើងខ្ញុំសូមធ្វើការផ្តល់អនុសាសន៍ដូចខាងក្រោម៖

- ធ្វើការស្វែងយល់អំពីការបង់ប្រាក់តាម online
- បន្ថែមមុខងារអាស័យដ្ឋានដោយភ្ជាប់ជាមួយ Map
- បន្ថែមនូវមុខងារថ្មីៗដែលជួយសំរួលក្នុងការស្វែងរកទំនិញឲ្យកាន់តែមានប្រសិទ្ធភាពដូចជា មុខងារស្វែងរកទំនិញនៅជិតទីតាំងរបស់អ្នកប្រើប្រាស់ មុខងារស្វែងរកទំនិញដែលមានតម្លៃតាមការកំណត់ណាមួយរបស់អ្នកប្រើប្រាស់។

## ឯកសារយោង

<https://www.bigcommerce.com/blog/ecommerce/#what-is-ecommerce>

<https://sosothea.wordpress.com>

<https://www.w3schools.com/>

<https://dart.dev/>

<https://flutter.dev/>

<https://firebase.google.com/>

<https://vuejs.org/>

<https://vuetifyjs.com/en/>

## ឧបសម្ព័ន្ធ ក្នុងក្នុងការបង្កើតកម្មវិធី

- main.dart

```
import
'package:firebase_core/firebase_core
.dart';
import
'package:flutter/cupertino.dart';
import
'package:flutter/material.dart';
import
'package:my_ecommerce/landing_page.d
art';

void main() async {

WidgetsFlutterBinding.ensureInitiali
zed();
  await Firebase.initializeApp();
  return runApp(
    MaterialApp(
      debugShowCheckedModeBanner:
false,
      home: LandingPage(),
    ),
  );
}
```

- Landing Page

```
import
'package:flutter/material.dart';
import
'package:my_ecommerce/models/user_ac
count.dart';
import
'package:my_ecommerce/pages/getstart
_page.dart';
import
'package:firebase_auth/firebase_auth
.dart';
import
'package:my_ecommerce/pages/home_pag
e.dart';
import
'package:my_ecommerce/services/auth_
service.dart';
import
```

```
'package:my_ecommerce/widget/alert_m
essage.dart';

// TODO: Clean code
class LandingPage extends
StatelessWidget {
  // FirebaseUser _user;
  final AuthService authService =
new AuthService();
  UserAccount user;

  // TODO: Implement not have
animate if already open
  bool _newOpen = true;

  @override
  Widget build(BuildContext context)
  {
    return StreamBuilder<User>(
      stream:
FirebaseAuth.instance.authStateChang
es(),
      builder: (context, snapshot) {
        print("Changed at Firebase
User");
        if (snapshot.connectionState
== ConnectionState.active) {
          if (snapshot.hasData) {
            return
_buildStream(context,
snapshot.data);
          } else {
            return GetStartPage(
              authService:
this.authService,
            );
          }
        } else {
          return Container(
            child: Center(
              child:
CircularProgressIndicator(),
            ),
          );
        }
      },
    );
  }
}
```

```

Widget _buildStream(BuildContext
context, User user) {
  if (user.isAnonymous) {
    return HomePage(
      authService:
this.authService,
      authUser: null,
    );
  }
  return
StreamBuilder<UserAccount>(
  stream:
  authService.onAuthStateChanged(
    uid: user.uid,
  ),
  builder: (context, snapshot) {
    print("changed at
UserAccount");
    if (snapshot.connectionState
== ConnectionState.active) {
      if (snapshot.data.enable
== false) {
        print(
          "Account
Disabled.....");
        _onAccountDisable(context);
        return GetStartPage(
          authService:
this.authService,
        );
      } else {
        print(
          "Enabled
Account.....");
        return HomePage(
          authService:
this.authService,
          authUser:
snapshot.data,
        );
      }
    } else {
      return Container(
        child: Center(
          child:
CircularProgressIndicator(),
        ),
      );
    }
  }
);

```

```

    }
  },
);
}

Future<void>
_onAccountDisable(BuildContext
context) async {
  showDialog(
    context: context,
    child: AlertMessage(
      title: "Can't login",
      content: "Your Account is
disabled",
    ),
  );
}

• Get Start Page
import
'package:flutter/material.dart';
import
'package:my_ecommerce/Animation/fade
_animation.dart';
import
'package:my_ecommerce/pages/login_pa
ge.dart';
import
'package:my_ecommerce/services/auth_
service.dart';

class GetStartPage extends
StatefulWidget {

  final AuthService authService;
  GetStartPage({@required
this.authService});

  @override
  _GetStartPageState createState()
=> _GetStartPageState();
}

class _GetStartPageState extends
State<GetStartPage>
  with TickerProviderStateMixin {
  AnimationController
_scaleController;
  Animation<double> _scaleAnimation;

  bool _hide = false;

```

```

@override
void initState() {
  // TODO: implement initState
  super.initState();

  _scaleController =
    AnimationController(vsync:
  this, duration:
  Duration(milliseconds: 500));

  _scaleAnimation =
    Tween<double>(begin: 1.0,
  end: 20.0).animate(_scaleController)

  ..addListener((status) {
    if (status ==
  AnimationStatus.completed) {
  // widget.signIn();
  widget.authService.signInAnonymously
  ();
    }
  });
}

@override
void dispose() {
  // TODO: implement dispose
  super.dispose();
  _scaleController.dispose();
}

@override
Widget build(BuildContext context)
{
  return Scaffold(
    body: _buildBody(),
  );
}

_buildBody() {
  return Container(
    width: double.infinity,
    decoration: BoxDecoration(
      image: DecorationImage(
  AssetImage('assets/images/splash.jpg
  '),
      fit: BoxFit.cover,
    ),
  ),
}

```

```

child: Container(
  decoration: BoxDecoration(
    gradient: LinearGradient(
      begin:
  Alignment.bottomRight,
      colors: [
  Colors.black.withOpacity(.9),
  Colors.black.withOpacity(.5),
    ],
  ),
  child: _buildContentPart(),
),
);

_buildContentPart() {
  return Padding(
    padding: const
  EdgeInsets.all(20.0),
    child: Column(
      crossAxisAlignment:
  CrossAxisAlignment.start,
      mainAxisAlignment:
  MainAxisAlignment.end,
      children: <Widget>[
  FadeAnimation(
    1.5,
    Text(
      "Brand New
  Perspective",
      textAlign:
  TextAlign.start,
      style: TextStyle(
        color:
  Colors.white,
        fontSize: 40,
        fontWeight:
  FontWeight.bold),
    ),
  ),
  SizedBox(
    height: 30,
  ),
  FadeAnimation(
    2.0,
    Text(
      "Let's Start with our
  collection.",
      style:

```





```

        ),
      ),
    ),
    SizedBox(
      height: 20,
    ),
  ],
),
);
}
}

```

- Home Page

```

import
'package:flutter/cupertino.dart';
import
'package:flutter/material.dart';
import
'package:flutter/rendering.dart';
import
'package:flutter/services.dart';
import
'package:my_ecommerce/animation/fade_animation.dart';
import
'package:my_ecommerce/models/category.dart';
import
'package:my_ecommerce/models/product.dart';
import
'package:my_ecommerce/models/user_account.dart';
import
'package:my_ecommerce/pages/cart_page.dart';
import
'package:my_ecommerce/pages/category_page.dart';
import
'package:my_ecommerce/pages/chat/chat_page.dart';
import
'package:my_ecommerce/pages/profile/user_profile_page.dart';
import
'package:my_ecommerce/pages/product/search_product_page.dart';
import
'file:///E:/FlutterProject/my_ecommerce/lib/uihelper/uihelper_product.da

```

```

rt';
import
'package:my_ecommerce/services/auth_service.dart';
import
'package:cloud_firestore/cloud_firestore.dart';
import
'package:my_ecommerce/services/product_service.dart';
import
'package:my_ecommerce/widget/alert_message.dart';
import
'package:page_transition/page_transition.dart';

// TODO: Fix animation: Don't has animation if it not from Login or register page
class HomePage extends StatefulWidget {
  final AuthService authService;
  final UserAccount authUser;

  HomePage({@required this.authService, @required this.authUser});

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  void initState() {
    // TODO: implement initState
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: _buildBody(),
      ),
    );
  }
}

```

```

Widget _buildBody() {
  return NestedScrollView(
    scrollDirection:
Axis.vertical,
    body: SingleChildScrollView(
      scrollDirection:
Axis.vertical,
      child: Container(
        child: Column(
          children: <Widget>[
            _buildCategoryPart(),
            Divider(
              thickness: 20,
            ),
          ],
        ),
      ),
    ),
    headerSliverBuilder:
(BuildContext context, bool
innerBoxIsScrolled) {
  return <Widget>[
    _buildSliverAppBar(),
  ];
},
);
}

Widget _buildSliverAppBar() {
  return SliverAppBar(
    expandedHeight: 350,
    pinned: true,
    elevation: 0,
    title: _buildSearchBar(),
    leading: IconButton(
      icon:
Icon(Icons.account_circle),
      onPressed: () {
        _onPressedAccount();
      },
    ),
    actions: <Widget>[
      IconButton(
        icon:
Icon(Icons.shopping_cart),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context)

```

```

=> CartPage(
      authUser:
widget.authUser,
    ),
  ),
);
},
),
IconButton(
  icon: Icon(Icons.chat),
  color: Colors.blue,
  onPressed: () {
    if (widget.authUser ==
null) {
      showDialog(
        context: context,
        child: AlertDialog(
          title: "Anonymous
Account",
          content:
            "You logged in
as anonymous. Please login with
email & password if you want to
chat",
        ),
      );
    } else {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context)
=> ChatPage(
            user:
widget.authUser,
          ),
        ),
      );
    }
  },
),
PopupMenuButton(
  icon:
Icon(Icons.more_vert),
  offset: Offset(0, 50),
  itemBuilder: (context) {
    return [
      PopupMenuItem(
        value: 'logout',
        child:
Text("logout"),
      ),
    ];
  },
);

```

```

    },
    onSelect: (val) {
      widget.authService.signOut();
    },
    onCancel: () {},
  ),
],
backgroundColor: Colors.black,
flexibleSpace: FadeAnimation(
  0.5,
  FlexibleSpaceBar(
    background: Container(
      decoration:
BoxDecoration(
  image:
DecorationImage(
  AssetImage('assets/images/background
.jpg'),
    fit: BoxFit.cover,
  ),
  ),
 )),
);

Widget _buildCategoryPart() {
  return Container(
    padding:
EdgeInsets.symmetric(vertical: 10),
    child: Column(
      children: <Widget>[
        Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: <Widget>[
            FadeAnimation(
              1.2,
              Padding(
                padding: const
EdgeInsets.symmetric(horizontal:
20),
                child: Text(
                  "Categories",
                  style:
TextStyle(
                    color:
Colors.black,
                    fontWeight:
FontWeight.bold,

```

```

              fontSize: 20,
            ),
          ),
        ],
      ),
      SizedBox(
        height: 20,
      ),
      FadeAnimation(
        1.4,
        Container(
          height: 120,
          child:
StreamBuilder<QuerySnapshot>(
  stream:
FirebaseFirestore.instance
.collection("category")
.snapshots(),
  builder: (context,
snapshot) {
    if
(snapshot.connectionState ==
ConnectionState.active) {
      return
_buildCategoryList(snapshot.data.doc
s);
    } else {
      return
Center(child:
CircularProgressIndicator());
    }
  },
),
),
),
),
SizedBox(
  height: 30,
),
],
),
);

Widget
_buildCategoryList(List<DocumentSnap
shot> documents) {
  List<Widget> _children = [];
  _children.add(SizedBox(
    width: 20,

```

```

    ));
    for (DocumentSnapshot doc in
documents) {

_children.add(_buildCategoryItem(cat
egory: Category.fromSnapshot(doc));
_children.add(SizedBox(width:
20));
}

return ListView(
scrollDirection:
Axis.horizontal,
children: _children,
);
}

Widget
_buildCategoryItem({Category
category}) {
return AspectRatio(
aspectRatio: 2 / 1.7,
child: Hero(
tag: category.imageUrl,
child: GestureDetector(
onTap: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context)
=> CategoryPage(
category:
category,
authUser:
widget.authUser,
),
),
);
},
child: Material(
child: Container(
//margin:
EdgeInsets.only(right: 20),
decoration:
BoxDecoration(
borderRadius:
BorderRadius.circular(10),
image:
DecorationImage(
image:
NetworkImage(category.imageUrl),
fit: BoxFit.cover,

```

```

),
),
child: Container(
padding:
EdgeInsets.all(10),
decoration:
BoxDecoration(
borderRadius:
BorderRadius.circular(10),
gradient:
LinearGradient(
begin:
Alignment.bottomRight,
colors: [
Colors.black.withOpacity(.8),
Colors.black.withOpacity(.0),
],
),
),
child: Align(
alignment:
Alignment.bottomLeft,
child: Text(
category.categoryName,
style:
TextStyle(
color:
Colors.white,
fontWeight:
FontWeight.bold,
fontSize: 15,
),
),
),
),
),
),
),
),
),
);

Widget _buildProductListStream() {
return
StreamBuilder<QuerySnapshot>(
stream:
ProductService.getAllPublished(),
builder: (context, snapshot) {
if (snapshot.connectionState

```

```

== ConnectionState.active) {
    return
    _buildProductList(snapshot.data.docs
);
    } else {
    return Center(
        child:
CircularProgressIndicator(),
    );
    },
);
}

Widget
_buildProductList(List<DocumentSnaps
hot> documents) {
    List<Widget> _children = [];
    _children.add(
        Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            children: <Widget>[
                Padding(
                    padding:
EdgeInsets.symmetric(horizontal: 20,
vertical: 10),
                    child: Text(
                        "Products",
                        style:
TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),
                    ),
                ),
                Divider(
                    height: 0,
                ),
            ],
        ),
    );
    for (DocumentSnapshot doc in
documents) {
        _children.add(_buildProduct(Product.
fromSnapshot(doc)));
        _children.add(Divider(
            thickness: 10,
        ));
    }
    return Container(
        child: Column(
            crossAxisAlignment:

```

```

CrossAxisAlignment.start,
            children: _children,
        ),
    );
}

Widget _buildProduct(Product
product) {
    return Container(
        padding:
EdgeInsets.symmetric(vertical: 10,
horizontal: 20),
        child: Column(
            children: [
                Container(
                    width: double.infinity,
                    child: Column(
                        mainAxisAlignment:
MainAxisAlignment.start,
                        children: <Widget>[
                            buildProfilePictureNameStream(
                                context,
                                widget.authUser, product.seller.id),
                                SizedBox(
                                    height: 10,
                                ),
                                Align(
                                    alignment:
Alignment.centerLeft,
                                    child:
Text(product.description),
                                ),
                                SizedBox(
                                    height: 10,
                                ),
                                buildProductItem(
                                    context: context,
                                    authUser:
                                widget.authUser,
                                    product: product,
                                    suffixHeroTag:
                                "home",
                                )
                            ],
                        ),
                    ],
                ),
            ],
        );
}

```

```

Widget _buildSearchBar() {
  return InkWell(
    onTap: () {
      Navigator.push(
        context,
        PageTransition(
          type:
PageTransitionType.fade,
          duration:
Duration(milliseconds: 300),
          child:
SearchProductPage(
            authUser:
widget.authUser,
          ),
        ),
      );
    },
    child: Container(
      height: 35,
      padding:
EdgeInsets.symmetric(horizontal: 8),
      alignment:
Alignment.centerRight,
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius:
BorderRadius.circular(20),
      ),
      child: Icon(
        Icons.search,
        color: Colors.black,
      ),
    ),
  );
}

void _onPressedAccount() {
  if (widget.authUser == null) {
    showDialog(
      context: context,
      child: AlertMessage(
        title: "Anonymous
Account",
        content: "You haven't
logged in",
      ),
    );
  } else {
    Navigator.push(
      context,
      PageTransition(

```

```

        type:
PageTransitionType.fade,
        child: UserProfilePage(
          userProfile:
widget.authUser,
          authUser:
widget.authUser,
        )),
      );
    }
  }
}

• Login Page
import
'package:flutter/material.dart';
import
'package:my_ecommerce/pages/forget_p
assword_page.dart';
import
'package:my_ecommerce/pages/register
/register_page.dart';
import
'package:my_ecommerce/services/auth_
service.dart';
import
'package:firebase_auth/firebase_auth
.dart';
import
'package:my_ecommerce/widget/alert_e
rror.dart';
import
'package:my_ecommerce/widget/rounded
_corner_button.dart';
import
'package:page_transition/page_transi
tion.dart';

// TODO: Fix keyboard go up to block
input field (Done)
class LoginPage extends
StatefulWidget {
  @override
  _LoginPageState createState() =>
  _LoginPageState();
}

class _LoginPageState extends
State<LoginPage> {
  TextEditingController
_emailController =
TextEditingController();

```

```

    TextEditingController
    _passwordController =
    TextEditingController();
    final AuthService _auth =
    AuthService();

    @override
    void dispose() {
      // TODO: implement dispose
      super.dispose();
      _emailController.dispose();
      _passwordController.dispose();
    }

    @override
    Widget build(BuildContext context)
    {
      return Scaffold(
        body: SingleChildScrollView(
          scrollDirection:
            Axis.vertical,
          child: Container(
            height:
              MediaQuery.of(context).size.height,
            color: Colors.white,
            child: Column(
              children: <Widget>[
                _buildTopPart(),
                _buildBodyPart(),
                _buildActionPart(),
              ],
            ),
          ),
        );
    }

    _buildTopPart() {
      return Stack(
        children: <Widget>[
          Container(
            height: 350,
            width: double.infinity,
            decoration: BoxDecoration(
              image: DecorationImage(
                AssetImage('assets/images/background
                .png'),
                fit: BoxFit.fill),
            ),
          ),
          Positioned(

```

```

            top: 160,
            right: 50,
            left: 50,
            child: Text(
              "Login",
              style: TextStyle(
                fontWeight:
                  FontWeight.bold, fontSize: 50,
                color: Colors.white),
              textAlign:
                TextAlign.center,
            ),
          ),
        ],
      );
    }

    _buildBodyPart() {
      return Padding(
        padding: EdgeInsets.only(left:
          40, right: 40, bottom: 20),
        child: _buildInputPart(),
      );
    }

    _buildInputPart() {
      return Container(
        padding: EdgeInsets.all(5),
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius:
            BorderRadius.circular(10),
          boxShadow: [
            BoxShadow(
              color:
                Color.fromRGBO(143, 148, 251, .3),
              blurRadius: 20.0,
              offset: Offset(0, 10))
          ],
        ),
        child: Column(
          children: <Widget>[
            Container(
              padding:
                EdgeInsets.all(8.0),
              child: TextField(
                controller:
                  _emailController,
                decoration:
                  InputDecoration(
                    border:
                      InputBorder.none,

```





```

        ),
      ),
    ),
  )),
  ],
),
);
}

void _onLogin() async {
  showDialog(
    context: context,
    builder: (_) {
      return AlertDialog(
        content: Container(
          height: 70,
          child: Center(
            child:
CircularProgressIndicator(),
          ),
        ),
      );
    },
  );
  String email =
_emailController.text;
  String password =
_passwordController.text;
  try {
    await _auth.signInUser(email,
password);
    Navigator.pop(context);
    Navigator.pop(context);
  } catch (e) {
    Navigator.pop(context);
    showDialog(
      context: context,
      builder: (_) {
        return AlertDialog(
          title: "Can't login",
          content: e.message,
        );
      },
    );
  }
}

void _gotoRegisterPage() {
  Navigator.push(
    context,
    PageTransition(
      duration:

```

```

Duration(milliseconds: 500),
      type:
PageTransitionType.rightToLeft,
      child: RegisterPage(),
    ),
  );
}
}

```

- Register Page

```

import
'package:flutter/material.dart';
import
'package:my_ecommerce/Pages/login_pa
ge.dart';
import
'package:my_ecommerce/models/user_ac
count.dart';
import
'package:my_ecommerce/services/auth_
service.dart';
import
'package:my_ecommerce/services/cart_
service.dart';
import
'package:my_ecommerce/services/user_
service.dart';
import
'package:my_ecommerce/widget/alert_e
rror.dart';
import
'package:firebase_auth/firebase_auth
.dart';

```

```

// TODO: Fix keyboard go up to block
input field (Done)

```

```

// TODO: remove data members of the
class

```

```

class RegisterPage extends
StatefulWidget {
  const RegisterPage({Key key}) :
super(key: key);

```

```

  @override
  _RegisterPageState createState()
=> _RegisterPageState();
}

```

```

class _RegisterPageState extends
State<RegisterPage> {
  final TextEditingController
_fullNameController =

```

```

TextEditingController();
  final TextEditingController
_emailController =
TextEditingController();
  final TextEditingController
_passwordController =
TextEditingController();
  final TextEditingController
_confirmPasswordController =
  TextEditingController();

  final AuthService _authService =
AuthService();

@override
void dispose() {
  // TODO: implement dispose
  super.dispose();
  _fullNameController.dispose();
  _emailController.dispose();
  _passwordController.dispose();

_confirmPasswordController.dispose()
;
}

@override
Widget build(BuildContext context)
{
  return Scaffold(
    body: SingleChildScrollView(
      scrollDirection:
Axis.vertical,
      child: Container(
        width:
MediaQuery.of(context).size.width,
        height:
MediaQuery.of(context).size.height,
        color: Colors.white,
        child: Column(
          children: <Widget>[
            _buildTopPart(),
            _buildBodyPart(),
          ],
        ),
      ),
    ),
  );
}

Widget _buildTopPart() {
  return Stack(

```

```

children: <Widget>[
  Container(
    height: 350,
    width: double.infinity,
    decoration: BoxDecoration(
      image:
DecorationImage(
        image:
AssetImage('assets/images/background
.png'),
        fit:
BoxFit.fill)),
    ),
  Positioned(
    top: 160,
    right: 50,
    left: 50,
    child: Text(
      "Register",
      style: TextStyle(
        fontWeight:
FontWeight.bold,
        fontSize: 50,
        color: Colors.white,
      ),
      textAlign:
TextAlign.center,
    ),
  ),
],
);
}

Widget _buildBodyPart() {
  return Container(
    padding:
EdgeInsets.symmetric(horizontal:
40),
    child: Column(
      children: <Widget>[
        Container(
          padding:
EdgeInsets.all(5),
          decoration:
BoxDecoration(
            color: Colors.white,
            borderRadius:
BorderRadius.circular(10),
            boxShadow: [
              BoxShadow(
                color:
Color.fromRGBO(143, 148, 251, .3),

```

```

        blurRadius: 20.0,
        offset: Offset(0,
10),
    ),
    ],
    ),
    child: _buildFormPart(),
    ),
    SizedBox(
        height: 40,
    ),
    _buildActionPart(),
    ],
    ),
    );
}

Widget _buildFormPart() {
    return Container(
        child: Column(
            children: <Widget>[
                _buildInput("Full Name",
controller: _fullNameController),
                Divider(),
                _buildInput("Email",
controller: _emailController),
                Divider(),
            ],
        ),
    );
}

Widget _buildInput(String
hintText, {@required
TextEditingController controller}) {
    return Container(
        padding:
EdgeInsets.symmetric(horizontal: 8),
        child: TextField(
            controller: controller,
            decoration: InputDecoration(
                border: InputBorder.none,
                hintText: hintText,

```

```

            hintStyle:
TextStyle(color: Colors.grey[400]),
        ),
    ),
    );
}

Widget _buildInputPassword(String
hintText,
    {@required
TextEditingController controller}) {
    return Container(
        padding:
EdgeInsets.symmetric(horizontal: 8),
        child: TextField(
            controller: controller,
            obscureText: true,
            decoration: InputDecoration(
                border: InputBorder.none,
                hintText: hintText,
                hintStyle:
TextStyle(color: Colors.grey[400]),
            ),
        ),
    );
}

Widget _buildActionPart() {
    return Container(
        child: Column(
            children: <Widget>[
                InkWell(
                    onTap: () async {
                        //TODO: Implement
                        signUp account here (Validate email
format)
                        this._register();
                    },
                ),
                child: Container(
                    height: 50,
                    decoration:
BoxDecoration(
                    borderRadius:
BorderRadius.circular(10),
                    gradient:
LinearGradient(
                        colors: [
                            Color.fromRGBO(143, 148, 251, 1),
                            Color.fromRGBO(143, 148, 251, 6),
                        ],

```



```

        showDialog(
          context: context,
          child: AlertDialog(
            title: "Can't create
User",
            content: e.message,
          ),
        );
      }
    } else {
      showDialog(
        context: context,
        child: AlertDialog(
          title: "Something
wrong",
          content: "Confirm
password must be the same Password",
        ),
      );
    }
  }
}

```

- Product Detail Page

```

import
'package:flutter/material.dart';
import
'package:my_ecommerce/models/cart.da
rt';
import
'package:my_ecommerce/models/categor
y.dart';
import
'package:my_ecommerce/models/chat.da
rt';
import
'package:my_ecommerce/models/product
.dart';
import
'package:my_ecommerce/models/user_ac
count.dart';
import
'package:my_ecommerce/pages/product/
edit_product.dart';
import
'package:my_ecommerce/pages/product/
product_reviews_page.dart';
import
'package:my_ecommerce/pages/product/
rate_review_product_page.dart';

```

```

import
'package:my_ecommerce/pages/profile/
user_profile_page.dart';
import
'package:my_ecommerce/services/cart_
service.dart';
import
'package:my_ecommerce/services/categ
ory_service.dart';
import
'package:my_ecommerce/services/chat_
service.dart';
import
'package:my_ecommerce/services/messa
ge_service.dart';
import
'package:my_ecommerce/services/produ
ct_service.dart';
import
'package:my_ecommerce/services/user_
service.dart';
import
'package:my_ecommerce/widget/alert_e
rror.dart';
import
'package:my_ecommerce/widget/carouse
l.dart';
import
'package:my_ecommerce/widget/rounded
_corner_button.dart';
import
'package:percent_indicator/linear_pe
rcent_indicator.dart';
import
'package:rating_bar/rating_bar.dart'
;

import '../chat/message_page.dart';

class ProductDetailPage extends
StatefulWidget {
  final Product product;
  final UserAccount authUser;
  final String suffixHeroTag;

  const ProductDetailPage(
    {Key key,
    @required this.product,
    @required this.authUser,
    this.suffixHeroTag = ""})
    : super(key: key);

```

```

    @override
    _ProductDetailPageState
    createState() =>
    _ProductDetailPageState();
  }

  class _ProductDetailPageState
  extends State<ProductDetailPage> {
    int currentIndex = 0;
    List<dynamic> imageUrl;

    UserAccount _seller;

    //Review
    int _userReviewIndex;

    @override
    void initState() {
      super.initState();
      imageUrl =
      widget.product.imageUrl;
      widget.product.view++;
      ProductService.set(
        widget.product,
        onDone: () {},
      );
    }

    @override
    Widget build(BuildContext context)
    {
      _checkIfReviewed();
      return SafeArea(
        child: Scaffold(
          body: _buildBody(),
        ),
      );
    }

    Widget _buildBody() {
      return Container(
        child: SingleChildScrollView(
          scrollDirection:
          Axis.vertical,
          child: Column(
            children: <Widget>[
              _buildCarouselPart(),
              _buildDetail(),
              _buildYourReview(),
              _buildRatesBlock(),
              _buildDescription(),
              _buildActionPart(),
            ],
          ),
        ),
      );
    }
  }

```

```

    ],
  ),
);
}

Widget _buildCarouselPart() {
  return Container(
    child: Stack(
      children: <Widget>[
        Carousel(
          imageUrl:
          widget.product.imageUrl,
          suffixHeroTag:
          widget.suffixHeroTag,
        ),
        Positioned(
          top: 5,
          left: 5,
          child: IconButton(
            icon: Icon(
              Icons.arrow_back,
              color: Colors.white,
            ),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ),
        ),
      ],
    ),
  );
}

Widget _buildDetail() {
  return Container(
    padding:
    EdgeInsets.symmetric(horizontal:
    30),
    child: Column(
      crossAxisAlignment:
      CrossAxisAlignment.start,
      children: <Widget>[
        SizedBox(
          height: 20,
        ),
        Row(
          crossAxisAlignment:
          CrossAxisAlignment.center,
          mainAxisAlignment:
          MainAxisAlignment.spaceBetween,

```

```

        children: <Widget>[
            Container(
                width:
MediaQuery.of(context).size.width *
2 / 3,
                child: Text(
widget.product.name,
                    style: TextStyle(
                        color:
Colors.black,
                        fontSize: 35,
                        fontWeight:
FontWeight.bold,
                        fontFamily:
'Avenir',
                    ),
                    maxLines: 2,
                    overflow:
TextOverflow.visible,
                ),
            ),
            Text('\$
{widget.product.price}')
        ],
    ),
    _buildCategoryName(widget.product.ca
tegory.id),
    SizedBox(
        height: 10,
    ),
    _buildProfilePictureNameStream(widge
t.product.seller.id),
    Container(
        padding:
EdgeInsets.symmetric(vertical: 10),
        child:
Text("{widget.product.view}
Views"),
    ),
    Divider(),
    ],
    ),
    );
}

Widget _buildCategoryName(String
categoryId) {
    return StreamBuilder<Category>(
        stream:

```

```

CategoryService.getById(id:
categoryId),
        builder: (context, snapshot) {
            if (snapshot.connectionState
== ConnectionState.active) {
                return Text(
                    snapshot.data.categoryName,
                    style: TextStyle(color:
Colors.black),
                );
            } else {
                return Text("Loading");
            }
        },
    );
}

Widget
_buildProfilePictureNameStream(String
uid) {
    return
StreamBuilder<UserAccount>(
        stream: UserService.get(uid),
        builder: (context, snapshot) {
            if (snapshot.connectionState
== ConnectionState.active) {
                if (snapshot.hasData) {
                    this._seller =
snapshot.data;
                    return
_buildProfilePictureAndName(this._se
ller);
                } else {
                    return Container();
                }
            } else {
                return Center(
                    child:
CircularProgressIndicator(),
                );
            }
        },
    );
}

Widget
_buildProfilePictureAndName(UserAcco
unt user) {
    return Row(
        children: <Widget>[
            GestureDetector(

```



```

        onTap: () {
      _gotoUserProfilePage(user);
    },
    child: CircleAvatar(
      radius: 25,
      backgroundImage:
NetworkImage(user.profiles == null
?
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'
:
user.profiles.length > 0
?
user.profiles[0]
:
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'),
    ),
    ),
    SizedBox(
      width: 10,
    ),
    Column(
      crossAxisAlignment:
CrossAxisAlignment.start,
      children: <Widget>[
        Text(
          user.fullName,
          style: TextStyle(
            color: Colors.black,
            fontSize: 15,
            fontWeight:
FontWeight.bold,
          ),
          textAlign:
TextAlign.start,
        ),
        Text(
          "Phone:
${user.phone}",
          style:
TextStyle(color: Colors.grey),
        ),
      ],
    ),
  ),

```

```

    ],
  );
}

Widget _buildActionPart() {
  if (widget.authUser == null) {
    return Container();
  }
  return Container(
    padding:
EdgeInsets.symmetric(horizontal:
30),
    child:
widget.product.seller.id ==
widget.authUser.uid
? Column(
      children: [
        RoundedCornerRadiusButton(
          color:
Colors.blue,
          child: Text(
            "Edit",
            style:
TextStyle(fontWeight:
FontWeight.bold),
          ),
          onTap: () {
            _onTapEdit();
          },
        ),
        SizedBox(
          height: 10,
        ),
        RoundedCornerRadiusButton(
          color: Colors.red,
          child: Text(
            "Delete this
product",
            style:
TextStyle(fontWeight:
FontWeight.bold),
          ),
          onTap: () {
            _onTapDeleteProduct();
          },
        ),
        SizedBox(
          height: 10,
        ),
      ],
    ),
  ),
}

```

```

        : Column(
            children: <Widget>[
                _buildActionAddToCart(),
                SizedBox(
                    height: 10,
                ),
                RoundedCornerRadius(
                    color:
Colors.blue,
                    child: Text(
                        "Send Message",
                        style:
TextStyle(
                            fontSize: 15,
                            fontWeight:
FontWeight.bold,
                            color:
Colors.black,
                        ),
                    ),
                    onTap: () {
                        _onSendMessage();
                    },
                ),
                SizedBox(
                    height: 10,
                ),
            ],
        ),
    );
}

Widget _buildActionAddToCart() {
    return StreamBuilder<Cart>(
        stream:
CartService.getByUser(widget.authUser.uid),
        builder: (context, snapshot) {
            if (snapshot.hasData) {
                bool _existedInCart =
false;
                for (var product in
snapshot.data.products) {
                    if (product.documentID
== widget.product.reference.id) {
                        _existedInCart = true;
                    }
                }
            }
            if (_existedInCart) {
                return

```

```

RoundedCornerRadius(
                    color: Colors.red,
                    child: Text(
                        "Remove from Cart",
                        style: TextStyle(
                            fontSize: 15,
                            fontWeight:
FontWeight.bold,
                        ),
                    ),
                    onTap:
_onRemoveFromCart,
                );
            } else {
                return
RoundedCornerRadius(
                    color: Colors.blue,
                    child: Text(
                        "Add to Cart",
                        style: TextStyle(
                            fontSize: 15,
                            fontWeight:
FontWeight.bold,
                        ),
                    ),
                    onTap: _onAddToCart,
                );
            }
        } else {
            return
RoundedCornerRadius(
                    color: Colors.blue,
                    child: Text(
                        "Add to Cart",
                        style: TextStyle(
                            fontSize: 15,
                            fontWeight:
FontWeight.bold,
                        ),
                    ),
                    onTap: _onAddToCart,
                );
        }
    );
}

void _onSendMessage() {
    ChatService.newChat(
        userIds:
[widget.authUser.uid,
widget.product.seller.id],

```

```

        onDone: (chat) {
            return
        }
        _onPressSendMessage(
            chat: chat,
            authUser:
                widget.authUser,
            otherUser: this._seller,
        );
    },
    onExisted: (chat) {
        return
    }
    _onPressSendMessage(
        chat: chat,
        authUser:
            widget.authUser,
        otherUser: this._seller,
    );
}

void _goToMessagePage(
    {Chat chat, UserAccount
    otherUser, UserAccount authUser}) {
    print("Existed");
    Navigator.of(context).pop();
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
                MessagePage(
                    chat: chat,
                    otherUser: otherUser,
                    authUser: authUser,
                ),
    );
}

_onPressSendMessage(
    {Chat chat, UserAccount
    otherUser, UserAccount authUser}) {
    MessageService.sendMessage(
        chat: chat,
        content:
            widget.product.reference.id,
        from: authUser.uid,
        type: "product")
        .then((value) {
            _goToMessagePage(
                chat: chat,
                authUser: authUser,
                otherUser: otherUser,

```

```

        );
    }).catchError((error) {}));
}

Widget _buildRatesBlock() {
    double _average = 0;
    double _oneStar = 0;
    double _twoStar = 0;
    double _threeStar = 0;
    double _fourStar = 0;
    double _fiveStar = 0;
    for (Map rate in
        widget.product.rates) {
        _average += rate['star'];
        if (rate['star'] == 1) {
            _oneStar++;
        } else if (rate['star'] == 2)
        {
            _twoStar++;
        } else if (rate['star'] == 3)
        {
            _threeStar++;
        } else if (rate['star'] == 4)
        {
            _fourStar++;
        } else if (rate['star'] == 5)
        {
            _fiveStar++;
        }
    }
    if (widget.product.rates.length
    > 0) {
        _average /=
            widget.product.rates.length;
        _oneStar /=
            widget.product.rates.length;
        _twoStar /=
            widget.product.rates.length;
        _threeStar /=
            widget.product.rates.length;
        _fourStar /=
            widget.product.rates.length;
        _fiveStar /=
            widget.product.rates.length;
    }

    return Container(
        padding:
            EdgeInsets.symmetric(horizontal:
                30),
        child: Column(
            crossAxisAlignment:

```

```

CrossAxisAlignment.start,
  children: <Widget>[
    Divider(),
    Text(
      "Rates: ",
      style: TextStyle(
        fontSize: 20,
        fontWeight:
FontWeight.bold,
      ),
    ),
    InkWell(
      onTap:
_gotoToProductReviews,
      child: Row(
        mainAxisAlignment:
MainAxisAlignment.spaceBetween,
        children: <Widget>[
          Column(

crossAxisAlignment:
CrossAxisAlignment.start,
        children:
<Widget>[
          Text(

"${_average.roundToDouble()}",
          style:
TextStyle(fontSize: 60),
        ),
        Row(
          children:
<Widget>[
          RatingBar.readOnly(
            maxRating:
5,
            initialRating: _average,
            isHalfAllowed: true,
            filledIcon: Icons.star,
            filledColor: Colors.blue,
            halfFilledIcon: Icons.star_half,
            halfFilledColor: Colors.blue,
            emptyIcon:
Icons.star_border,
            size: 15,

```

```

          ),
        ],
      ),
    Text("(${widget.product.rates.length})"),
  ],
),
Container(
  child: Column(
    children:
<Widget>[
    LinearPercentIndicator(
      width:
MediaQuery.of(context).size.width /
2,
      lineHeight:
10,
      percent:
_fiveStar,
      progressColor: Colors.blue,
      leading:
Text("5"),
    ),
    LinearPercentIndicator(
      width:
MediaQuery.of(context).size.width /
2,
      lineHeight:
10,
      percent:
_fourStar,
      progressColor: Colors.blue,
      leading:
Text("4"),
    ),
    LinearPercentIndicator(
      width:
MediaQuery.of(context).size.width /
2,
      lineHeight:
10,
      percent:
_threeStar,
      progressColor: Colors.blue,
      leading:

```



```

        ? Container()
        : Container(
            padding:
EdgeInsets.symmetric(horizontal:
30),
            child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: <Widget>[
                    Text(
                        "Your Review",
                        style: TextStyle(
                            fontSize: 18,
                            fontWeight:
FontWeight.bold,
                        ),
                    ),
                    Row(
                        mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                        children:
<Widget>[
                            Row(
                                children:
<Widget>[
                                    CircleAvatar(
                                        radius:
25,
                                        backgroundImage: NetworkImage(widget
.authUser.profiles ==
null
?
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'
:
widget.authUser.profiles.length > 0
?
widget.authUser.profiles[0]
:
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'
),

```

```

                SizedBox(
                    width: 10,
                ),
                Column(
                    crossAxisAlignment:
CrossAxisAlignment.start,
                    children:
<Widget>[
                        Text(widget.authUser.fullName),
                        Row(
                            children: <Widget>[
                                RatingBar.readOnly(
                                    initialRating: widget
.product.rates[_userReviewIndex]['st
ar']
                                    .toDouble(),
                                    filledIcon: Icons.star,
                                    emptyIcon: Icons.star_border,
                                    size: 15,
                                ),
                            ],
                        ),
                    ],
                ),
                PopupMenuButton(
                    icon:
Icon(Icons.more_vert),
                    offset:
Offset(0, 50),
                    itemBuilder:
(context) {
                        return [
                            PopupMenuItem(
                                value:
'delete',
                                child:
Text("Delete"),
                            ),
                        ];
                    }
                );

```

```

        },
        onSelect:
    (val) {
        _deleteRateReview();
    },
    onCancel: ()
    {}),
    ],
    ),
    SizedBox(
        height: 10,
    ),
    Text(widget.product.rates[_userReviewIndex]['review']),
    FlatButton(
        padding:
    EdgeInsets.all(0),
        child: Text("Edit
    your review"),
        splashColor:
    Colors.blue[100],
        textColor:
    Colors.blue,
        onPressed:
    _gotoRateAndReviewProduct,
    ),
    ],
    );
}

void
_gotoUserProfilePage(UserAccount
user) {
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
    UserProfilePage(
        userProfile: user,
        authUser: widget.authUser,
    ),
    ),
    );
}

void _gotoRateAndReviewProduct() {
    Navigator.push(
        context,

```

```

        MaterialPageRoute(
            builder: (context) =>
    RateReviewProductPage(
        product: widget.product,
        authUser: widget.authUser,
    ),
    ),
    );
}

void _gotoProductReviews() {
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
    ProductReviewsPage(
        authUser: widget.authUser,
        product: widget.product,
    ),
    ),
    );
}

void _deleteRateReview() {
    widget.product.rates.removeAt(_userReviewIndex);
    ProductService.set(
        widget.product,
        onDone: () {
            setState(() {
                _userReviewIndex = null;
            });
        },
    );
}

void _checkIfReviewed() {
    for (int i = 0; i <
    widget.product.rates.length; i++) {
        if
    (widget.product.rates[i]['user'].documentID ==
    widget.authUser.reference.id) {
            _userReviewIndex = i;
        }
    }
}

void _onAddToCart() {
    CartService.addProductToCart(

```

```

        uid: widget.authUser.uid,
        productRef:
        widget.product.reference,
      ).catchError((e) {
        showDialog(
          context: context,
          child: AlertDialog(
            title: "Error",
            content: e.message,
          ),
        );
      });
    }

    void _onRemoveFromCart() {
      CartService.removeProductFromCart(
        uid: widget.authUser.uid,
        productRef:
        widget.product.reference,
      );
    }

    void _onTapDeleteProduct() {
      showDialog(
        context: context,
        builder: (_) {
          return AlertDialog(
            content: Container(
              height: 70,
              child: Center(
                child:
                CircularProgressIndicator(),
              ),
            ),
          );
        },
      );
      ProductService.removeProduct(
        product: widget.product,
        onDone: () {
          Navigator.pop(context);
          Navigator.pop(context);
        },
      );
    }

    void _onTapEdit() {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) =>

```

```

EditProduct(
          product: widget.product,
        ),
      ),
    ).then((value) {
      setState(() {});
    });
  }
}

• User Profile Page

import 'dart:io';

import
'package:flutter/cupertino.dart';
import
'package:flutter/material.dart';
import
'package:my_ecommerce/models/chat.dart';
import
'package:my_ecommerce/pages/chat/message_page.dart';
import
'package:my_ecommerce/pages/profile/edit_cover_image_page.dart';
import
'package:my_ecommerce/pages/profile/edit_profile_image_page.dart';
import
'package:my_ecommerce/uihelper/uihelper_product.dart';
import
'package:my_ecommerce/services/chat_service.dart';
import 'package:toast/toast.dart';
import
'package:image_picker/image_picker.dart';
import
'package:cloud_firestore/cloud_firestore.dart';
import
'package:my_ecommerce/pages/product/add_product.dart';
import
'package:my_ecommerce/pages/profile/edit_user_info_page.dart';
import
'package:my_ecommerce/pages/view_image_screen.dart';
import

```



```

'package:my_ecommerce/widget/rounded
_corner_button.dart';
import
'package:my_ecommerce/models/user_ac
count.dart';
import
'package:my_ecommerce/models/product
.dart';
import
'package:my_ecommerce/services/produ
ct_service.dart';

class UserProfilePage extends
StatefulWidget {
  final UserAccount userProfile;
  final UserAccount authUser;

  UserProfilePage({Key key,
this.userProfile, this.authUser}) :
super(key: key);

  @override
  _UserProfilePageState
createState() =>
UserProfilePageState();
}

class _UserProfilePageState extends
State<UserProfilePage> {
  final ImagePicker _picker =
ImagePicker();

  /*File _image;
String _imageUrl;

  Future _getImage() async {
    final pickedFile = await
_picker.getImage(source:
ImageSource.gallery);
    setState(() {
      _image =
File(pickedFile.path);
    });
  }*/

  @override
  void dispose() {
    super.dispose();
  }

  @override
  Widget build(BuildContext context)
{
  return SafeArea(
    child: Scaffold(
      body: _buildBody(),
      floatingActionButton:
_buildFAB(),
    ),
  );
}

_buildBody() {
  return SingleChildScrollView(
    scrollDirection:
Axis.vertical,
    child: Container(
      child: Column(
        children: <Widget>[
_buildCoverProfilePart(),
_buildBio(),

_buildSendMessageButton(),
Divider(
  thickness: 10,
),
_buildUserInfo(),
Divider(thickness: 10),

_buildProductListStream(),
],
      ),
    ),
  );
}

_buildCoverProfilePart() {
  return Stack(
    overflow: Overflow.visible,
    alignment: Alignment.center,
    children: <Widget>[
_buildCover(),
_buildProfilePart(),
_buildCloseButton(),
],
  );
}

_buildCloseButton() {
  return Positioned(
    top: 0,
    left: 0,
    child: IconButton(

```

```

        icon: Icon(
          Icons.arrow_back,
        ),
        color: Colors.white,
        onPressed: () {
          Navigator.pop(context);
        },
      ),
    );
  }

  _buildProfilePart() {
    return Positioned(
      bottom: -50,
      left: 20,
      child: Row(
        crossAxisAlignment:
CrossAxisAlignment.end,
        children: <Widget>[
          _buildProfile(),
          Padding(
            padding: const
EdgeInsets.only(left: 10, bottom:
15),
            child: Text(

widget.userProfile.fullName,
              style: TextStyle(
                fontWeight:
FontWeight.bold,
                fontSize: 20,
              ),
            ),
          ],
        ),
      );
    }

    _buildProfile() {
      if (widget.userProfile.profiles
== null ||

widget.userProfile.profiles.length
== 0) {
        return InkWell(
          borderRadius:
BorderRadius.circular(70),
          onTap:
_showModalBottomSheetProfile,
          child: CircleAvatar(
            radius: 70,

```

```

          backgroundImage:
NetworkImage(
            'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266',
          ),
        ),
      );
    }
    return InkWell(
      borderRadius:
BorderRadius.circular(70),
      onTap:
_showModalBottomSheetProfile,
      child: Hero(
        tag:
widget.userProfile.profiles[widget.u
serProfile.profiles.length - 1],
        child: CircleAvatar(
          radius: 70,
          backgroundImage:
NetworkImage(

widget.userProfile.profiles == null
?
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'
          : widget.userProfile

.profiles[widget.userProfile.profile
s.length - 1],
        ),
      ),
    );
  }

  _buildCover() {
    if (widget.userProfile.covers ==
null ||

widget.userProfile.covers.length ==
0) {
      return InkWell(
        onTap:
_showModalBottomSheetCover,

```

```

        child: Container(
          height: 300,
          width:
MediaQuery.of(context).size.width,
          child: Container(
            decoration:
BoxDecoration(
              gradient:
LinearGradient(
                begin:
Alignment.topLeft,
                end:
Alignment.bottomRight,
                colors: [
Colors.black.withOpacity(.3),
Colors.black.withOpacity(.0)
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
  return InkWell(
    onTap:
_showModalBottomSheetCover,
    child: Hero(
      tag:
widget.userProfile.covers[widget.use
rProfile.covers.length - 1],
      child: Material(
        child: Container(
          height: 300,
          width:
MediaQuery.of(context).size.width,
          decoration:
BoxDecoration(
            image:
DecorationImage(
              image: NetworkImage(
                widget
.userProfile.covers[widget.userProfi
le.covers.length - 1],
              ),
              fit: BoxFit.cover,
            ),
          ),
          child: Container(
            decoration:

```

```

BoxDecoration(
  gradient:
LinearGradient(
  begin:
Alignment.topLeft,
  end:
Alignment.bottomRight,
  colors: [
Colors.black.withOpacity(.3),
Colors.black.withOpacity(.0)
  ],
),
),
),
),
),
),
),
);
}

_buildBio() {
  return Column(
    children: <Widget>[
      SizedBox(
        height: 60,
      ),
      (widget.userProfile.bio ==
null ||
widget.userProfile.bio.trim().isEmpty
)
        ? Container()
        : Text(
            "(
${widget.userProfile.bio} )",
            style:
TextStyle(color: Colors.grey,
fontSize: 15),
          ),
      SizedBox(
        height: 20,
      ),
    ],
  );
}

_buildSendMessageButton() {
  //if client logged in as
anonymous
  if (widget.authUser == null) {

```

```

        return Container();
    } else {
        // if it's own profile
        if (widget.authUser.uid ==
            widget.userProfile.uid) {
            return Container();
        }
    }
    return Container(
        padding: EdgeInsets.only(left:
            30, right: 30, top: 0, bottom: 30),
        child: RoundedCornerRadiusButton(
            color: Colors.blue,
            child: Text(
                "Message",
                style: TextStyle(
                    fontSize: 15,
                    fontWeight:
                        FontWeight.bold,
                    decoration:
                        TextDecoration.none,
                    color: Colors.black,
                ),
                textAlign:
                    TextAlign.center,
            ),
            onTap: _onMessageTap,
        ),
    );
}

_buildUserInfo() {
    return Container(
        width: double.infinity,
        padding: EdgeInsets.only(left:
            20, right: 20, top: 20),
        child: Column(
            children: <Widget>[
                Row(
                    children: <Widget>[
                        Icon(Icons.home),

                        (widget.userProfile.province == null
                            ||
                            widget.userProfile.province.isEmpty)
                            ? Text("No
                                province")
                            : Text("Live in
                                ${widget.userProfile.province}")
                    ],
                ),
            ],
        ),
    );
}

```

```

        SizedBox(
            height: 10,
        ),
        Row(
            children: <Widget>[
                Icon(Icons.phone_iphone),

                widget.userProfile.phone == null ||
                widget.userProfile.phone.trim().isEmpty
                    ? Text("No Phone
                        number")
                    :
                Text(widget.userProfile.phone)
            ],
        ),
        SizedBox(
            height: 10,
        ),
        Container(
            padding:
                EdgeInsets.only(bottom: 15),
            child: Row(
                crossAxisAlignment:
                    CrossAxisAlignment.start,
                children: <Widget>[
                    Icon(Icons.location_on),
                    Container(
                        width:
                            MediaQuery.of(context).size.width -
                            65,
                        child:
                            (widget.userProfile.address == null
                                ||
                                widget.userProfile.address.isEmpty)
                                ? Text("No
                                    Address")
                                : Text(
                                    "${widget.userProfile.address}",
                                    maxLines:
                                        3,
                                ),
                    ),
                ],
            ),
        ),
    ),
    widget.authUser == null

```

```

                ? Container()
                : widget.authUser.uid
    == widget.userProfile.uid
                ? FlatButton(
                    child: Text(
                        "Edit",
                        style:
    TextStyle(color: Colors.blue),
                    ),
                    onPressed: ()
    {
        _gotoEditUserInfoPage();
    },
                )
                : Container(),
            ],
        ),
    );
}

_buildProductListStream() {
    return Container(
        child:
    StreamBuilder<QuerySnapshot>(
        stream:

    ProductService.getBySeller(seller:
    widget.userProfile.reference),
        builder: (context,
    snapshots) {
            if
    (snapshots.connectionState ==
    ConnectionState.active) {
                return
    _buildProductList(snapshots.data.doc
    s);
            } else {
                return Center(
                    child:
    CircularProgressIndicator(),
                );
            }
        },
    ),
    );
}

_buildProductList(List<DocumentSnap
shot> documents) {
    List<Widget> _children = [];

```

```

        /*_children.add(Text(
            "Products",
            style: TextStyle(fontSize: 20,
    fontWeight: FontWeight.bold),
        ));*/
        for (DocumentSnapshot doc in
    documents) {
            _children.add(_buildProductItem(Prod
    uct.fromSnapshot(doc));
        }
        _children.add(Divider(thickness:
    10));
    }
    return Column(
        children: _children,
    );
}

Widget _buildProductItem(Product
product) {
    return Container(
        padding:
    EdgeInsets.symmetric(horizontal:
    20),
        child: Column(
            children: <Widget>[
                Container(
                    width: double.infinity,
                    child: Column(
                        mainAxisAlignment:
    MainAxisAlignment.start,
                        children: <Widget>[
                            SizedBox(
                                height: 10,
                            ),
                            Align(
                                alignment:
    Alignment.centerLeft,
                                child: Text(
                                    product.name,
                                    style:
    TextStyle(
                                        color:
    Colors.black,
                                        fontSize: 18,
                                        fontWeight:
    FontWeight.bold,
                                        decoration:
    TextDecoration.underline,
                                    ),
                                ),
                            ),
                        ],
                    ),
                ),
            ],
        ),
    );
}

```

```

        ),
        SizedBox(height:
10),
        Align(
            alignment:
Alignment.centerLeft,
            child:
Text(product.description),
        ),
        SizedBox(
            height: 10,
        ),
    ],
),
SizedBox(height: 10),
// Image
buildProductItem(
    context: context,
    product: product,
    authUser:
widget.authUser,
    suffixHeroTag:
"profile",
),
),
);
}

void onTapEditProfile() {
    Navigator.pop(context);
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
EditProfileImagePage(
                authUser: widget.authUser,
            ),
        ).then((value) {
            setState(() {});
        });
}

void onTapEditCover() {
    Navigator.pop(context);
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
EditCoverImagePage(

```

```

                authUser: widget.authUser,
            ),
        ).then((value) {
            setState(() {});
        });
    }

    void
_showModalBottomSheetProfile() {
        showModalBottomSheet(
            context: context,
            builder: (context) {
                return
_buildBottomSheetProfile();
            },
        );
    }

    Widget _buildBottomSheetProfile()
    {
        return Container(
            child: Wrap(
                children: <Widget>[
                    ListTile(
                        leading:
Icon(Icons.image),
                        title: Text("View
Profile"),
                        onTap:
_viewProfileImage,
                    ),
                    widget.authUser == null
                    ? Container()
                    :
widget.userProfile.uid ==
widget.authUser.uid
                    ? ListTile(
                        leading:
Icon(Icons.edit),
                        title:
Text("Edit Profile"),
                        onTap: ()
                    )
                    : Container(),
                ],
            ),
        );
    }

```

```

}

_showModalBottomSheetCover() {
  showModalBottomSheet(
    context: context,
    builder: (context) {
      return
_buildBottomSheetCover();
    },
  );
}

Widget _buildBottomSheetCover() {
  return Container(
    child: Wrap(
      children: <Widget>[
        ListTile(
          leading:
Icon(Icons.image),
          title: Text("View
Cover"),
          onTap: _viewCoverImage,
        ),
        widget.authUser == null
          ? Container()
          :
widget.userProfile.uid ==
widget.authUser.uid
          ? ListTile(
            leading:
Icon(Icons.edit),
            title:
Text("Edit Cover"),
            onTap: ()
async {
              await
onTapEditCover();
            },
          )
          : Container(),
      ],
    ),
  );
}

Widget _buildFAB() {
  if (widget.authUser == null ||
widget.authUser.uid ==
widget.userProfile.uid) {
    return FloatingActionButton(
      child: Icon(Icons.add),
      backgroundColor:

```

```

Colors.blue,
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) =>
AddProduct(
              user:
widget.authUser,
            ),
          ),
        );
      },
    );
  } else {
    return Container();
  }
}

void _gotoEditUserInfoPage() {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) =>
EditUserInfoPage(
        user: widget.authUser,
      ),
    ),
  );
}

void _viewCoverImage() {
  Navigator.pop(context);
  if (widget.userProfile.covers !=
null &&
widget.userProfile.covers.length >
0) {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
ViewImageScreen(
          initialPage: 0,
          imageUrls:
widget.userProfile.covers.reversed.t
oList(),
        ),
      ),
    );
  } else {
    Toast.show(

```

```

        "No Image",
        context,
        duration:
Toast.LENGTH_SHORT,
        gravity: Toast.BOTTOM,
    );
}
}

void _viewProfileImage() {
    Navigator.pop(context);
    if (widget.userProfile.profiles
!= null &&
widget.userProfile.profiles.length >
0) {
        Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) =>
ViewImageScreen(
                    initialPage: 0,
                    imageUrls:
widget.userProfile.profiles.reversed
.toList(),
                ),
            ),
        );
    } else {
        Toast.show(
            "No Image",
            context,
            duration:
Toast.LENGTH_SHORT,
            gravity: Toast.BOTTOM,
        );
    }
}

void _onMessageTap() async {
    await ChatService.newChat(
        userIds: [widget.authUser.uid,
widget.userProfile.uid],
        onDone: (chat) {
            return _goToMessagePage(
                chat: chat,
                authUser: widget.authUser,
                otherUser:
widget.userProfile,
            );
        },
        onExisted: (chat) {

```

```

            return _goToMessagePage(
                chat: chat,
                authUser: widget.authUser,
                otherUser:
widget.userProfile,
            );
        },
    );
}

_goToMessagePage({Chat chat,
UserAccount otherUser, UserAccount
authUser}) {
    print("Existed");
    Navigator.of(context).pop();
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
MessagePage(
                chat: chat,
                otherUser: otherUser,
                authUser: authUser,
            ),
        ),
    );
}
}

```

- Chat Page

```

import
'package:flutter/material.dart';
import
'package:cloud_firestore/cloud_firestore.dart';
import
'package:my_ecommerce/models/chat.dart';
import
'package:my_ecommerce/models/user_account.dart';
import
'package:my_ecommerce/pages/chat/message_page.dart';
import
'package:my_ecommerce/pages/chat/new_chat_page.dart';
import
'package:my_ecommerce/services/chat_service.dart';
import

```



```

'package:page_transition/page_transi
tion.dart';

class ChatPage extends
StatefulWidget {
  final UserAccount user;

  ChatPage({@required this.user});

  @override
  _ChatPageState createState() =>
  _ChatPageState();
}

class _ChatPageState extends
State<ChatPage> {
  Stream<List<Chat>>
chatsListStream;

  @override
  void initState() {
    super.initState();
    this.chatsListStream =

ChatService.getChatsByUserRef(userId
: widget.user.uid);
  }

  @override
  void dispose() {
    super.dispose();
  }

  @override
  Widget build(BuildContext context)
{
  return Scaffold(
    appBar: _buildAppBar(),
    body: _buildBody(),
  );
}

Widget _buildAppBar() {
  return AppBar(
    backgroundColor: Colors.black,
    elevation: 0,
    automaticallyImplyLeading:
false,
    title: Container(
      padding: EdgeInsets.all(5),
      child: Row(
        crossAxisAlignment:

```

```

CrossAxisAlignment.center,
    children: <Widget>[
      CircleAvatar(
        backgroundColor:
Colors.white,
        backgroundImage:
NetworkImage(
  widget.user.profiles == null
?
'https://firebasestorage.googleapis.
com/v0/b/e-commerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'
:
widget.user.profiles[widget.user.pro
files.length - 1],
        scale: 1),
    ),
    SizedBox(
      width: 15,
    ),
    Text(
      "Chats",
      style:
TextStyle(color: Colors.white,
fontSize: 20),
    ),
  ],
),
actions: <Widget>[
  CircleAvatar(
    backgroundColor:
Colors.white,
    child: IconButton(
      icon:
Icon(Icons.create),
      color: Colors.black,
      onPressed: () {
        //List<Chat>
_chatList;
        this.chatsListStream.first.then((lis
t) => list);
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context)
=> NewChatPage(
              user:

```

```

widget.user,
        ),
    ),
    );
    },
    tooltip: "New chat",
    ),
),
IconButton(
    icon: Icon(Icons.cancel),
    color: Colors.red,
    onPressed: () {
Navigator.of(context).pop();
    },
    tooltip: "Exit to Home
Page",
    ),
    SizedBox(
        width: 15,
    ),
    ],
);
}

Widget _buildBody() {
    return Container(
        child:
        StreamBuilder<List<Chat>>(
            stream:
            this.chatsListStream,
            builder: (context, snapshot)
        {
            if
            (snapshot.connectionState ==
            ConnectionState.active) {
                if (snapshot.hasData) {
                    return
                    _buildChatList(snapshot.data);
                } else {
                    return Container();
                }
            } else {
                return Center(
                    child:
                    CircularProgressIndicator(),
                );
            }
        },
    );
}

```

```

Widget _buildChatList(List<Chat>
chats) {
    return Container(
        child: ListView.builder(
            itemCount: chats.length,
            itemBuilder: (context,
index) {
                return
                _buildChatItemStream(chats[index]);
            },
        ),
    );
}

Widget _buildChatItemStream(Chat
chat) {
    String userIds0 =
chat.userIds[0].toString();
    String userIds1 =
chat.userIds[1].toString();
    String otherUserId =
widget.user.uid == userIds0 ?
userIds1 : userIds0;

    return
    StreamBuilder<DocumentSnapshot>(
        stream:
        FirebaseFirestore.instance
        .collection("users")
        .doc(otherUserId)
        .snapshots(),
        builder: (context, snapshot) {
            if (snapshot.connectionState
            == ConnectionState.active) {
                Future.delayed(Duration(milliseconds
: 1000));
                return
                _buildChatItem(UserAccount.fromSnaps
hot(snapshot.data), chat);
            } else {
                //TODO: Implement skeleton
                return
                _buildChatItemSkeleton();
            }
        },
    );
}

Widget _buildChatItem(UserAccount
user, Chat chat) {

```

```

return ListTile(
  leading: CircleAvatar(
    radius: 25,
    backgroundColor:
Colors.black,
    backgroundImage:
NetworkImage(user.profiles == null
?
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'
: user.profiles.length >
0
? user.profiles[0]
:
'https://firebasestorage.googleapis.
com/v0/b/ecommerce-app-
ba448.appspot.com/o/profiles%2Fno-
profile.png?alt=media&token=6e4b5ccc
-3d93-44bf-8296-cd492018d266'),
  title: Text(
    user.fullName,
    style: TextStyle(
      fontWeight:
chat.lastMessageFrom == user.uid
? FontWeight.bold
:
FontWeight.normal),
  ),
  subtitle: chat.lastMessageFrom
== user.uid
? Text("Last sent by:
${user.fullName}",
style:
TextStyle(fontWeight:
FontWeight.bold))
: Text("You sent last
message"),
  onTap: () {
    Navigator.push(
      context,
      PageTransition(
        type:
PageTransitionType.fade,
        duration:
Duration(milliseconds: 300),
        child: MessagePage(
          chat: chat,
          authUser: widget.user,

```

```

        otherUser: user,
      ),
    ),
  );
},
);
}

Widget _buildChatItemSkeleton() {
return ListTile(
  leading: CircleAvatar(
    radius: 25,
    backgroundColor:
Colors.black,
  ),
  title: Text("Username"),
  subtitle: Text("....."),
);
}
}

• Forget Password Page

import
'package:firebase_auth/firebase_auth
.dart';
import
'package:flutter/material.dart';
import
'package:my_ecommerce/services/auth_
service.dart';
import
'package:my_ecommerce/widget/alert_e
rror.dart';
import
'package:my_ecommerce/widget/rounded
_corner_button.dart';

class ForgetPasswordPage extends
StatelessWidget {
  final TextEditingController
_emailController =
TextEditingController();
  final _auth =
FirebaseAuth.instance;

  @override
  Widget build(BuildContext context)
{
  return Scaffold(
    body: SingleChildScrollView(
      scrollDirection:

```

```

Axis.vertical,
  child: Container(
    width:
MediaQuery.of(context).size.width,
    height:
MediaQuery.of(context).size.height,
    color: Colors.white,
    child: Column(
      children: <Widget>[
        _buildTopPart(),
        _buildBodyPart(),
      ],
    ),
  ),
);
}

_buildTopPart() {
  return Stack(
    children: <Widget>[
      Container(
        height: 350,
        width: double.infinity,
        decoration: BoxDecoration(
          image: DecorationImage(
            image:
AssetImage('assets/images/background
.png'),
            fit: BoxFit.fill),
        ),
      Positioned(
        top: 160,
        right: 50,
        left: 50,
        child: Text(
          "Reset Password",
          style: TextStyle(
            fontWeight:
FontWeight.bold, fontSize: 35,
            color: Colors.white),
          textAlign:
TextAlign.center,
        ),
      ),
    ],
  );
}

```

```

_buildBodyPart() {
  return Padding(
    padding: EdgeInsets.only(left:
40, right: 40, bottom: 20),
    child: _buildInputPart(),
  );
}

_buildInputPart() {
  return Container(
    padding: EdgeInsets.all(5),
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius:
BorderRadius.circular(10),
      boxShadow: [
        BoxShadow(
          color:
Color.fromRGBO(143, 148, 251, .3),
          blurRadius: 20.0,
          offset: Offset(0, 10))
      ],
    ),
    child: Column(
      children: <Widget>[
        Container(
          padding:
EdgeInsets.all(8.0),
          child: TextField(
            controller:
_emailController,
            decoration:
InputDecoration(
              border:
InputBorder.none,
              hintText: "Email or
UserName",
              hintStyle:
TextStyle(color: Colors.grey[400]),
            ),
          ),
        ],
      ),
    );
}

_buildActionPart(context) {
  return Container(
    padding: EdgeInsets.only(left:
40, right: 40, bottom: 40),
    child: Column(

```

```

        children: <Widget>[
          SizedBox(height: 20),
          Container(
            color:
Color.fromRGBO(143, 148, 251, 1),
            child:
RoundedCornerRadiusButton(
              onTap: () {
                _onSubmit(context);
              },
              child: Text(
                "Submit",
                style: TextStyle(
                  color:
Colors.white,
                  fontSize: 20,
                  fontWeight:
FontWeight.bold,
                ),
              ),
              color:
Color.fromRGBO(143, 148, 251, 1),
            ),
          SizedBox(height: 20),
          Container(
            child: InkWell(
              onTap: () {
                _gotoLoginPage(context);
              },
              child: Center(
                child: Text(
                  "Return to Login",
                  style: TextStyle(
                    color:
Colors.grey[400],
                    fontWeight:
FontWeight.bold,
                  ),
                ),
              ),
            ),
          ],
        ),
      );
    }

    void _gotoLoginPage(context) {
      Navigator.pop(context);
    }

```

```

    void _onSubmit(context) {
      _auth.sendPasswordResetEmail(email:
_emailController.text).then((_) {
        showDialog(
          context: context,
          builder: (_) {
            return AlertDialog(
              title: "Done",
              content: "Reset link was
sent",
            );
          },
        );
      }).catchError((error) {
        print(error.message);
        showDialog(
          context: context,
          builder: (_) {
            return AlertDialog(
              title: "Can't Reset",
              content: error.message,
            );
          },
        );
      });
    }

```

- Category Page

```

import
'package:flutter/cupertino.dart';
import
'package:flutter/material.dart';
import
'package:my_ecommerce/pages/cart_
page.dart';
import
'package:my_ecommerce/pages/profi
le/user_profile_page.dart';
import
'package:my_ecommerce/models/cate
gory.dart';
import
'package:my_ecommerce/models/prod
uct.dart';

```

```

import
'package:my_ecommerce/models/user_account.dart';
import
'package:my_ecommerce/pages/product/search_product_page.dart';
import
'product/product_detail_page.dart';
import
'package:my_ecommerce/services/user_service.dart';
import
'file:///E:/FlutterProject/my_ecommerce/lib/uihelper/uihelper_product.dart';
import
'package:my_ecommerce/services/product_service.dart';
import
'package:page_transition/page_transition.dart';
import
'package:cloud_firestore/cloud_firestore.dart';

class CategoryPage extends StatefulWidget {
  final Category category;
  final UserAccount authUser;

  const CategoryPage(
    {Key key, @required
this.category, @required
this.authUser})
    : super(key: key);

  @override
  CategoryPageState createState()
=> CategoryPageState();
}

class CategoryPageState extends State<CategoryPage> {
  @override
  Widget build(BuildContext context) {

```

```

return NestedScrollView(
  scrollDirection:
Axis.vertical,
  headerSliverBuilder:
(BuildContext context, bool
innerBoxIsScrolled) {
    return <Widget>[
      _buildSliverAppBar(),
    ];
  },
  body: _buildBody(),
);
}

_buildSliverAppBar() {
  return SliverAppBar(
    expandedHeight: 300.0,
    floating: false,
    pinned: true,
    backgroundColor:
Colors.black,
    elevation: 0,
    actions: <Widget>[
      IconButton(
        icon: Icon(
          Icons.search,
          color: Colors.white,
        ),
        onPressed: () {
          Navigator.push(
            context,
            PageTransition(
              type:
PageTransitionType.fade,
              duration:
Duration(milliseconds: 300),
              child:
SearchProductPage(
                authUser:
widget.authUser,
              ),
            ),
          );
        },
      ),
    ],
  );
}
}

```

```

        Icons.shopping_cart,
        color: Colors.white,
      ),
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder:
(context) => CartPage(
              authUser:
widget.authUser,
            ),
          ),
        );
      },
    ],
    flexibleSpace:
FlexibleSpaceBar(
  centerTitle: true,
  title: Text(
    widget.category.categoryName,
    style: TextStyle(
      color: Colors.white,
      fontSize: 16.0,
    ),
  ),
  background: Hero(
    tag:
widget.category.imageUrl,
    child: Material(
      child: Container(
        decoration:
BoxDecoration(
          image:
DecorationImage(
            NetworkImage(widget.category.imag
eUrl),
            fit:
BoxFit.cover,
          ),
        ),
        child: Container(
          decoration:
BoxDecoration(

```

```

          gradient:
LinearGradient(
            begin:
Alignment.bottomRight,
            colors: [
Colors.black.withOpacity(.8),
Colors.black.withOpacity(.1),
            ],
          ),
        ),
      ),
    ),
  ),
  forceElevated: true,
);

_buildBody() {
  return Material(
    child:
SingleChildScrollView(
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: <Widget>[
          Container(
            padding:
EdgeInsets.all(20),
            child: Text(
              "New Product",
              style: TextStyle(
                color:
Colors.black,
                fontSize: 20,
                fontWeight:
FontWeight.bold,
              ),
            ),
          Divider(),
        ],
      ),
    ),
  ),
  _buildProductListStream(),
),

```

```

    ),
  );
}

_buildProductListStream() {
  return
  StreamBuilder<QuerySnapshot>(
    stream:
    ProductService.getByCategory(cate
    gory: widget.category.reference),
    builder: (context,
    snapshots) {
      if
      (snapshots.connectionState ==
      ConnectionState.active) {
        return
        _buildProductList(snapshots.data.
        docs);
      } else {
        return Center(
          child:
          CircularProgressIndicator(),
        );
      }
    },
  );
}

Widget
_buildProductList(List<DocumentSn
apshot> documents) {
  List<Widget> _children = [];
  for (DocumentSnapshot doc in
  documents) {

    // _children.add(_buildProductItem
    (Product.fromSnapshot(doc)));

    _children.add(_buildProduct(Produ
    ct.fromSnapshot(doc)));

    _children.add(Divider(thickness:
    10));
  }
  return Column(
    children: _children,
  );
}

```

```

}

Widget _buildProduct(Product
product) {
  return Container(
    padding:
    EdgeInsets.symmetric(vertical:
    10, horizontal: 20),
    child: Column(
      children: [
        Container(
          width:
          double.infinity,
          child: Column(
            mainAxisAlignment:
            MainAxisAlignment.start,
            children: <Widget>[

              buildProfilePictureNameStream(
                context,
                widget.authUser,
                product.seller.id),
                SizedBox(
                  height: 10,
                ),
                Align(
                  alignment:
                  Alignment.centerLeft,
                  child:
                  Text(product.description),
                ),
                SizedBox(
                  height: 10,
                ),
                buildProductItem(
                  context:
                  context,
                  authUser:
                  widget.authUser,
                  product:
                  product,
                  suffixHeroTag: "category")
            ],
          ),
        ),
      ],
    ),
  );
}

```



```
}  
  }  
);
```