



**សាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រនិងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច**

Université Royale De Droit et des Sciences Économiques  
Royal University of Law and Economics



**សារណាបញ្ចប់ការសិក្សា**

**ការបង្កើត App សម្រាប់ការទិញទំនិញនៃ  
ហាង Nike Sneaker App Store**

ស្រាវជ្រាវពីថ្ងៃទី ០៣ ខែមេសា ឆ្នាំ ២០២៣ ដល់ថ្ងៃទី ៣១ ខែ ឧសភា ឆ្នាំ ២០២៣។

ស្រាវជ្រាវដោយ

និស្សិតឈ្មោះ: កញ្ញា **ស៊ុក អន្តាសិល**  
កញ្ញា **អ៊ុយ ស៊ីណាល**

ណែនាំដោយ

សាស្ត្រាចារ្យជំនួយបណ្ឌិត **ទិន ហេង**

ថ្នាក់បរិញ្ញាបត្រ **ព័ត៌មានវិទ្យា**

ជំនាន់ទី០២

ឆ្នាំចូលសិក្សា

ឆ្នាំសរសេរសារណា

២០១៩

២០២៣

**សេចក្តីថ្លែងអំណរគុណ**

ក្រុមសិក្សាស្រាវជ្រាវទាំងអស់គ្នាជានិស្សិត នៃសាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រនិងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច ជំនាន់ទី០២ នៃមហាវិទ្យាល័យ សេដ្ឋកិច្ចព័ត៌មានវិទ្យា ផ្នែកព័ត៌មានវិទ្យាដែលមានសមាជិកក្រុមក្រុមចំនួន០២រូបគឺ៖

- ១ ស៊ុក អង្គាសិល
- ២ អ៊ុច ស៊ីណាន

**សូមគោរពប្រណិបត្តិដោយសេចក្តីកត់ត្រាគាប់ចំពោះ**

លោកអ្នកជាមាតាបិតា និងអាណាព្យាបាល របស់ក្រុមសិក្សាស្រាវជ្រាវទាំងអស់គ្នា ដែលលោកបានផ្តល់នូវកំណើតការចិញ្ចឹមបីបាច់ថែរក្សាអប់រំទូន្មានប្រៀនប្រដៅនិងការឧបត្ថម្ភគ្រប់បែបគ្រប់យ៉ាងទាំងផ្នែកស្មារតី សម្ភារតាំងពី កុមារភាពរហូតដល់យើងទាំងអស់គ្នាបានបញ្ចប់ថ្នាក់បរិញ្ញាប័ត្រ ដោយរូបលោកទាំងអស់គ្នាបានជំនះរាល់ឧបសគ្គដោយមិនគិតពីការ នឿយហត់ឡើយ។

សូមថ្លែងអំណរគុណដ៏ជ្រាលជ្រៅចំពោះការទទួលស្វាគមន៍ដ៏រាក់ទាក់និងយ៉ាងកក់ក្តៅបំផុតក្នុងការជួយឧបត្ថម្ភខាងផ្នែកគំនិតស្មារតីនិងសម្ភាររបស់៖

- សាស្ត្រាចារ្យជំនួយលោក **ទិន ហេង** ជាសាស្ត្រាចារ្យនៃសាកលវិទ្យាល័យ ភូមិន្ទនីតិសាស្ត្រនិងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច ដែលបានផ្តល់ចំណេះដឹងនិងចំណាយពេលវេលាដ៏មានតម្លៃ ក្នុងការជួយផ្តល់នូវគំនិតយោបល់ និងគំនុះល្អៗព្រមទាំងបានផ្តល់នូវឯកសារសំខាន់ៗសម្រាប់ធ្វើការសិក្សាស្រាវជ្រាវបន្ថែមលើនេះទៀត។
- លោកព្រឹទ្ធបុរស និងលោកព្រឹទ្ធបុរសរងនៃមហាវិទ្យាល័យ សេដ្ឋកិច្ចព័ត៌មានវិទ្យា ដែលបានជួយបង្ហាត់បង្រៀនទាំងទ្រឹស្តី ការអនុវត្ត និងណែនាំនូវបទពិសោធន៍ជាមូលដ្ឋានគ្រឹះមួយចំនួន។

ជាទីបញ្ចប់ក្រុមសារណា ទាំងអស់គ្នាសូមប្រសិទ្ធិពរជ័យ ដល់លោកអ្នកមានគុណ លោកសាស្ត្រាចារ្យទាំងអស់ ព្រមទាំងលោកនាយក និងបុគ្គលិកទាំងអស់នៃសាកលវិទ្យាល័យ ភូមិន្ទនីតិសាស្ត្រនិងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច សូមអោយជួបតែពុទ្ធពរទាំង៤ប្រការគឺ អាយុ វណ្ណៈ សុខៈ ពលៈ កុំបីឃ្លាឃ្លាតឡើយ។

ភ្នំពេញ ថ្ងៃទី ខែ ឆ្នាំ២០២៣ ក្រុមនិស្សិត  
ស្រាវជ្រាវនៃសាកលវិទ្យាល័យភូមិន្ទនីតិសាស្ត្រ  
និងវិទ្យាសាស្ត្រសេដ្ឋកិច្ច

**សេចក្តីថា**

ប្រទេសកម្ពុជាជាប្រទេសកំពុងអភិវឌ្ឍន៍ដែលមានការរីកចម្រើនទៅលើគ្រប់វិស័យទាំងអស់ជាពិសេស វិស័យបច្ចេកវិទ្យាដែលកំពុងវាយលុកនិងពេញនិយមជាខ្លាំងនាសម័យបច្ចុប្បន្ន។ យើងសង្កេតឃើញថា ការរីកដុះដាលផ្នែកប្រព័ន្ធអ៊ីនធឺណេត ធ្វើអោយមានការកើនឡើងនៃអ្នកប្រើទូរស័ព្ទច្រើន ក្នុងគោលបំណង ផ្សេងៗដូចជា ធ្វើការផ្សព្វផ្សាយ, ការស្រាវជ្រាវ និងទំនាក់ទំនងគ្នាជាដើម ដែលជាហេតុធ្វើអោយបច្ចេកវិទ្យា ខាងទូរស័ព្ទនេះក្លាយជារសនិយម និងជាមធ្យោបាយដំបូងបង្អស់ក្នុងការបំពេញនូវតម្រូវការទាំងអស់នោះ។ យ៉ាងណាមិញវិស័យអាជីវកម្មតូចធំគ្រប់ប្រភេទទាំងអស់ក៏សុទ្ធសឹងតែប្រើប្រាស់ប្រព័ន្ធអ៊ីនធឺណេត ដើម្បី សម្រួលដល់ការទិញលក់ជូនដល់អតិថិជន ដែលបង្ហាញអោយឃើញថាការធ្វើអាជីវកម្មតាម App ទូរស័ព្ទ ដៃគឺជាជម្រើសនៃបច្ចេកវិទ្យាថ្មីនៅលើទីផ្សារផងដែរ។

ដោយស្របទៅលើការរីកចម្រើនខាងលើទើបធ្វើអោយក្រុមពួកយើងខ្ញុំសម្រេចលើកយកការសរសេរ បង្កើត App លក់ស្បែកជើងឈ្មោះថា Nike Sneaker App Store ដើម្បីមកធ្វើការបង្ហាញពីការទិញ លក់របស់ហាង Nike Sneaker App Store នៅលើ App ទូរស័ព្ទ។

# មាតិកា

## ទំព័រ

មាតិកា.....	i
បញ្ជីតារាង .....	iv
បញ្ជីរូបភាព .....	v

### សេចក្តីផ្តើម

១. មូលហេតុនៃការសិក្សា .....	1
២. ចំណោទបញ្ហានៃការសិក្សា.....	1
៣. គោលបំណងនៃការសិក្សា.....	2
៤. វិសាលភាពនៃការសិក្សា.....	2
៥.អត្ថប្រយោជន៍នៃការសិក្សា .....	2
៥.១ ចំពោះអ្នកលក់ .....	3
៥.២ ចំពោះអតិថិជន.....	3
៥.៣ ចំពោះសង្គមជាតិ .....	3

### ជំពូកទី១

#### ឯកសារ និង ទ្រឹស្តីដែលពាក់ព័ន្ធ

១.១. ឯកសារដែលពាក់ព័ន្ធ.....	4
១.២. ទ្រឹស្តីពាក់ព័ន្ធនឹង E-Commerce.....	4
១.៣. កម្មវិធី Microsoft និង Database.....	6
១.៣.១. ទ្រឹស្តីពាក់ព័ន្ធនឹង Microsoft .....	6
១.៣.២. ប្រវត្តិរបស់ Microsoft Visual Studio Code (VS Code).....	6
១.៣.៥. ទ្រឹស្តីពាក់ព័ន្ធនឹងផ្នែក Database .....	7
១.៣.៥.១. និយមន័យ Database .....	7
១.៣.៥.២. អ្វីទៅជា Database Management System(DBMS)?.....	8
១.៤. ទ្រឹស្តីពាក់ព័ន្ធនឹងភាសាបង្កើត Application.....	11
១.៤.១ Dart Programming .....	11
១.៤.២ តើអ្វីជា Flutter? .....	11

១.៤.២.១ Widgets .....	12
១.៤.២.២. Features of Flutter .....	13
១.៤.២.៣. MVC Architecture Pattern .....	13
១.៤.២.៤. State maintenance widgets.....	15
១.៤.៣ ទ្រឹស្តី Node.js.....	15
១.៤.៤. ទ្រឹស្តី PostgreSQL.....	18
១.៥ ទ្រឹស្តីពាក់ព័ន្ធនិងវិធីសាស្ត្រនៃការវិភាគប្រព័ន្ធ .....	21
១.៥.១. Work Flow Diagram.....	21
១.៥.២. Data Flow Diagram (DFD) .....	22

## **ជំពូកទី២**

### **វិធីសាស្ត្រនិងរចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ**

២.១ វិធីសាស្ត្រនៃការសិក្សាស្រាវជ្រាវ.....	24
២.១.១. System Planning.....	26
២.១.២. System Analysis.....	26
២.១.៣. System Design.....	26
២.១.៤. System Implementation .....	26
២.១.៥. System Maintenance.....	26
២.២ រចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ.....	26
២.២.១ សេចក្តីផ្តើម.....	27
២.២.២ ជំពូកទី១៖ រំលឹកទ្រឹស្តី ដែលពាក់ព័ន្ធ .....	28
២.២.៣ ជំពូកទី២៖ វិធីសាស្ត្រ និងរចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ.....	28
២.២.៤. ជំពូកទី៣៖ ការវិភាគ ការគ្រោង និងការអនុវត្តន៍ .....	28
២.២.៥. ជំពូកទី៤៖ សំយោគ សន្និដ្ឋាន និងការផ្តល់អនុសាសន៍ .....	29
២.៣ គំរោងពេលវេលានៃការសិក្សាស្រាវជ្រាវ.....	29

## **ជំពូកទី៣**

### **ការវិភាគ ការគ្រប់គ្រង និងការអនុវត្តន៍**

៣.១. ការវិភាគ.....	31
៣.១.១ ការងារជាក់ស្តែងទាក់ទងនឹងប្រព័ន្ធបច្ចុប្បន្ន .....	31

៣.១.២ ដំណើរការនៃការងាររបស់ប្រព័ន្ធបច្ចុប្បន្ន.....	31
៣.១.៣ ការវាយតម្លៃលើប្រព័ន្ធបច្ចុប្បន្ន.....	31
៣.១.៤ ការវិភាគលើប្រព័ន្ធសំណើរ .....	32
៣.១.៥ ដំណើរការលើប្រព័ន្ធសំណើរ .....	33
៣.២. ការគ្រោងប្រព័ន្ធ.....	33
៣.២.១. ការគ្រោងនៅលើ Process and User Interface.....	33
៣.២.១.១. Screen Design របស់ User.....	33
ក. Register& Login Screen .....	33
ខ. Home Page.....	35
គ. Search Screen.....	36
ឃ. Details Page.....	37
ង. Cart Page .....	38
ច. Order Screen.....	39
ឆ. Notification Screen .....	40
ជ. Profile Screen.....	41
៣.២.១.២. Process .....	42
៣.៣. ការអនុវត្តន៍ .....	43

**ជំពូកទី៤**

**សំយោគ សន្និដ្ឋាន និងការផ្តល់អនុសាសន៍**

៥.១. សំយោគ .....	45
៥.២. សេចក្តីសន្និដ្ឋាន.....	45

**ឯកសារយោង**

**ឧបសម្ព័ន្ធ**

## **បញ្ជីតារាង**

តារាង 1: និមិត្តសញ្ញា Work Flow Diagram.....	22
តារាង 2 : និមិត្តសញ្ញា Data Flow Diagram.....	23
តារាង 3:គំរោងពេលវេលានៃការសិក្សាស្រាវជ្រាវ.....	29

## បញ្ជីរូបតារាង

រូបភាព 1: Type of E-Commerce.....	5
រូបភាព 2: Important of using Database .....	8
រូបភាព 3: Type of DBMS .....	9
រូបភាព 4: Architecture of Flutter Application .....	12
រូបភាព 5: MVC Architecture Pattern .....	14
រូបភាព 6: Important of using Node.js.....	16
រូបភាព 7: How client side work with server side .....	19
រូបភាព 8: How Node.js work with PostgreSQL .....	20
រូបភាព 9: Phases of SDLC .....	25
រូបភាព 10: រចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ .....	27
រូបភាព 11 Context diagram level .....	33
រូបភាព 12 : Login Screen.....	34
រូបភាព 13: Register Screen.....	35
រូបភាព 14: Home Screen.....	36
រូបភាព 15 : Search Screen.....	37
រូបភាព 16 : Details Screen .....	38
រូបភាព 17 : Cart Screen .....	39
រូបភាព 18 : Order Screen.....	40
រូបភាព 19 : Notification Screen.....	41
រូបភាព 20: Profile Screen .....	42
រូបភាព 21 : Process របស់Appបញ្ជាទិញទំនិញ Nike Sneaker online store.....	43



**សេចក្តីផ្តើម**

ស្របតាមតម្រូវការបច្ចេកវិទ្យាកាន់តែទំនើបនាពេលបច្ចុប្បន្ន និងការកើនឡើងនូវតម្រូវការរបស់អតិថិជនក៏ មានការកើនឡើងច្រើន ធ្វើអោយវិស័យព័ត៌មានវិទ្យាដើរតួនាទីយ៉ាងសំខាន់ក្នុងជួយដល់ម្ចាស់អាជីវកម្មនិងអតិថិជន ដើម្បីដោះស្រាយនូវបញ្ហាខាងលើដូចជា បង្កើនលទ្ធភាពលក់តាមរយៈការផ្សព្វផ្សាយតាមបែបបច្ចេកវិទ្យាទំនើប កាត់បន្ថយភាពស្មុគស្មាញដែលកើតមានឡើងក្នុងការគ្រប់គ្រងអាជីវកម្ម កាត់បន្ថយកម្លាំងពលកម្មមនុស្ស និងកាត់បន្ថយការចំណាយពេលរបស់អតិថិជនក្នុងការស្វែងរកទំនិញ ។

នៅតាមក្រុមហ៊ុនម្ចាស់អាជីវកម្មខ្នាតយក្សទាំងក្នុងនិងក្រៅប្រទេសជាច្រើនដូចជា Amazon, eBay, Taobao,.....។ សុទ្ធសឹងតែដាក់បញ្ចូលអោយប្រើប្រាស់នូវ System បែបឌីជីថល ជាអ្នកគ្រប់គ្រងជំនួស អោយមនុស្ស វាផ្តល់នូវសេវាកម្មទិញលក់ទៅដល់អតិថិជនទូទាំងពិភពលោក និងជាអ្នកសម្របសម្រួល ដល់ការគ្រប់គ្រងរបស់ម្ចាស់អាជីវកម្មផងដែរ។

ដូច្នេះហើយទើបប្រធានបទនៃ **ការបង្កើត App សម្រាប់ការទិញទំនិញនៃ ហាង Nike Sneaker App Store** ត្រូវបានលើកយកមកសិក្សា ព្រោះវាបានបង្ហាញយើងពីការប្រើប្រាស់នូវយុទ្ធសាស្ត្រទិញលក់បែបឌីជីថលលើ App ទូរស័ព្ទដៃ” ពោលគឺរាល់ការទិញលក់គ្រប់ប្រភេទទំនិញទាំងអស់សុទ្ធតែដំណើរការលើប្រព័ន្ធ Online ” ដែលអាចបំពេញតម្រូវការរបស់អតិថិជនបានគ្រប់ទីកន្លែង រហ័ស និងមានសុវត្ថិភាព ។

**១. មូលហេតុនៃការសិក្សា**

សព្វថ្ងៃនេះកំណើនសេដ្ឋកិច្ចមានការកើនឡើង តម្រូវការរបស់អតិថិជនក៏មានការកើនឡើងឥតឈប់ឈរ ទើបធ្វើឲ្យការធ្វើអាជីវកម្មលើ App ទូរស័ព្ទមានការរីកដុះដាលជាកត់សម្គាល់។ យើងសង្កេតឃើញថាមានម្ចាស់ អាជីវកម្មជាច្រើនក៏ធ្វើការវិវត្តន៍ខ្លួនទៅកាន់ការប្រើប្រាស់Appដើម្បីពង្រីកអាជីវកម្មរបស់ខ្លួនទៅកាន់ទីផ្សារ និង ដើម្បីសម្របតាមតម្រូវការរបស់អតិថិជន។ ហេតុនេះហើយទើប Nike Sneaker App Store ត្រូវបានលើកមក សិក្សា និងផ្សព្វផ្សាយពីអត្ថប្រយោជន៍នៃបច្ចេកវិទ្យាថ្មីនាសម័យបច្ចុប្បន្ន។

**២. ចំណោទបញ្ហានៃការសិក្សា**

ការធ្វើអាជីវកម្មតាមបែបប្រពៃណីធ្វើឲ្យបាត់បង់នូវសក្តានុពលនៃទំនិញ (ពោលគឺពិបាកធ្វើការផ្សព្វផ្សាយពី ទំនិញទៅកាន់ទីផ្សារ) រីឯតម្រូវការរបស់មនុស្សក៏មានការកើនឡើងពីមួយថ្ងៃទៅមួយថ្ងៃឥតឈប់ឈរ ដែលធ្វើឲ្យ មានការខ្វះខាតរាល់ការផ្គត់ផ្គង់របស់ម្ចាស់អាជីវកម្មទៅដល់អតិថិជននៅមានកម្រិត។ ចំណែកឯអតិថិជន វិញ ពិបាកធ្វើការស្វែងរកទំនិញនៅលើទីផ្សារដែលត្រូវការ ការចំណាយពេលវេលាតិច និងមានទំនុកចិត្ត។

**៣. គោលបំណងនៃការសិក្សា**

ក្រុមសារណាយើងខ្ញុំបានលើកយកប្រធានបទ “ ការបង្កើត App សម្រាប់ការទិញទំនិញនៃហាង Nike Sneaker App Store ” នេះមកធ្វើការសិក្សានិងបកស្រាយបង្ហាញពីដំណោះស្រាយនូវបញ្ហាខាងលើដែលជាជំនួយនៃការធ្វើអាជីវកម្មដោយការដាក់ឱ្យប្រើប្រាស់ App ដូចខាងក្រោម៖

- មានសុវត្ថិភាពជូនដល់អតិថិជនប្រើប្រាស់
- បង្កើតទីផ្សារលក់នៅលើ Online ដើម្បីស្វែងរកអតិថិជន និងបង្កើនលទ្ធភាពលក់
- មាន feature notification សម្រាប់ធ្វើការផ្សព្វផ្សាយ រាល់ការនាំចូលនៅផលិតផលថ្មី ក៏ដូចជាផ្សព្វផ្សាយពីផលិតផលដែលត្រូវ promotion
- មាន feature search សម្រាប់អតិថិជនស្វែងរកទំនិញបានរហ័សតាមតម្រូវការ
- មាន បែងចែកប្រភេទទំនិញ ដើម្បីសម្រួលដល់តម្រូវការរបស់អតិថិជន
- អតិថិជនអាចចូលទៅមើលលើ details របស់ផលិតផលមុនធ្វើការបញ្ជាទិញ
- សម្រាប់ម្ចាស់អាជីវកម្មក៏ទទួលបាន System ដែលជាអ្នកសម្របសម្រួលដល់ការគ្រប់គ្រងរាល់ការទិញលក់រវាងម្ចាស់អាជីវកម្មនិងអតិថិជន។

**៤. វិសាលភាពនៃការសិក្សា**

ការបង្កើត App សម្រាប់ការទិញទំនិញនៃហាង Nike Sneaker App Store ត្រូវបានចាប់ផ្តើមការស្រាវជ្រាវពីថ្ងៃទី ០៦ ខែមិនា ដល់ថ្ងៃទី ២១ ខែ សីហា ឆ្នាំ ២០២៣។ ដែល App នេះប្រើប្រាស់នូវបច្ចេកវិទ្យាថ្មីទាន់សម័យកាលដែលអាចបំពេញតម្រូវការដល់អតិថិជនបានរហ័សទាន់ចិត្ត និងជួយ សម្រួលដល់ការងារគ្រប់គ្រងរបស់ម្ចាស់អាជីវកម្មឱ្យកាន់តែមានភាពងាយស្រួលក្នុងការផ្សព្វផ្សាយទំនិញរបស់ខ្លួនទៅកាន់អតិថិជនថែមទៀតផង។

**៥. អត្ថប្រយោជន៍នៃការសិក្សា**

ការស្រាវជ្រាវបែបកុរកអំពីវត្ថុបំណងអ្វីមួយ យើងតែងតែមានគោលបំណងផ្សេងៗគ្នាទៅតាមតម្រូវការចង់បានរបស់ក្រុមសិក្សានីមួយៗ។ យ៉ាងណាមិញគោលបំណង នៃក្រុមសារណា របស់យើង ខ្ញុំចង់ផ្តោតសំខាន់លើការគ្រប់គ្រងនៃការលក់ផលិតផលនៅលើ App ទូរស័ព្ទ ដើម្បីឆ្លុះបញ្ចាំងពីលទ្ធផល និងការអនុវត្តន៍ជាក់ស្តែង។ ជាពិសេសទៅទៀតនោះ គឺក្រុមយើងខ្ញុំបានធ្វើការស្រាវជ្រាវដើម្បីសម្រេច បានគោលដៅក្នុងការបង្កើតជា App ទូរស័ព្ទថ្មីមួយ ដោយធ្វើយ៉ាងណាឱ្យ App របស់យើងមានភាពងាយស្រួលទៅកាន់អតិថិជនក្នុងការកម្រង់អីវ៉ាន់ពីខាងហាងយើងខ្ញុំផ្តល់ភាពងាយស្រួលក្នុងការបញ្ចូលទិន្នន័យ កែសម្រួល ឬលុប។

**៥.១ ចំពោះអ្នកលក់**

ផលប្រយោជន៍ចំពោះអ្នកលក់ងាយស្រួលក្នុងការគ្រប់គ្រងប្រព័ន្ធការទិញលក់កាត់បន្ថយបុគ្គលិកតិច ថែមទាំងបង្កើនទំនុកចិត្តអោយភ្ញៀវអោយកាន់តែទុកចិត្តមកលើហាងយើងផងដែរហើយផលិតផលរបស់ហាង និងមានភាពទូលំទូលាយជាងមុនមានគេស្គាល់ច្រើននិងផ្តល់នូវភាពងាយស្រួលចំពោះអតិថិជនក្នុងការស្វែង រកទំនិញពីខាងហាងផងដែរ។

**៥.២ ចំពោះអតិថិជន**

- ចំណាយពេលវេលាតិចចំណេញទាំងថវិការមិនពិបាកក្នុងការទៅទិញអីវ៉ាន់ដល់ហាងដែរឆ្ងាយពីទីតាំង ពួកគេ
- អោយអតិថិជនបានស្វែងយល់ពីប្រព័ន្ធអ៊ីនធឺណែតនៃបច្ចេកវិទ្យាទំនើប
- ទទួលបានព័ត៌មានថ្មីៗពីអីវ៉ាន់ខាងហាង។

**៥.៣ ចំពោះសង្គមជាតិ**

- ជួយបង្កើតធនធានមនុស្សតាមរយៈការសិក្សាស្រាវជ្រាវរបស់និស្សិត
- និស្សិតនឹងទទួលបានចំណេះដឹងថ្មីៗដើម្បីយកទៅធ្វើការអភិវឌ្ឍន៍ចំណុចដែលខ្វះខាតក៏ដូចជា សមត្ថភាពបន្ថែមដើម្បីដើរទាន់សម័យកាលបច្ចុប្បន្ន
- ទទួលបានឱកាសដុះខាត់លើវិស័យការងារ បង្កើតចំណេះជំនាញច្បាស់លាស់
- នាំឱ្យសង្គមមានភាពដុះដាលលើគ្រប់វិស័យ ជាកត្តានាំឱ្យជាតិមានការរីកចម្រើន។

## ជំពូកទី១

### ឯកសារ និង ទ្រឹស្តីដែលពាក់ព័ន្ធ

ក្រោយពីបានធ្វើការស្រាវជ្រាវពីការសរសេររបង្កើត App នៃការបញ្ជាទិញទំនិញរបស់ហាង “ Nike Sneaker App Store ” មួយរយៈកន្លងមក ពួកយើងបានធ្វើការស្វែងយល់ពីបញ្ហា និងទទួលបានទិន្នន័យ ក៏ដូចជាព័ត៌មានបន្ថែមមួយចំនួនដែលពាក់ព័ន្ធនឹងហាងលក់ស្បែកជើងនេះ។ យោងតាមការលើកឡើងនូវបញ្ហារបស់ប្រធានបទខាងលើគប្បីឲ្យយើងធ្វើការប្រមូលទិន្នន័យ និងឯកសារដែលពាក់ព័ន្ធ មកធ្វើការបកស្រាយចងក្រង និងអនុវត្តន៍ដើម្បីឲ្យការសិក្សារបស់ប្រធានបទខាងលើទទួលបានប្រសិទ្ធិភាពខ្ពស់។

#### ១.១. ឯកសារដែលពាក់ព័ន្ធ

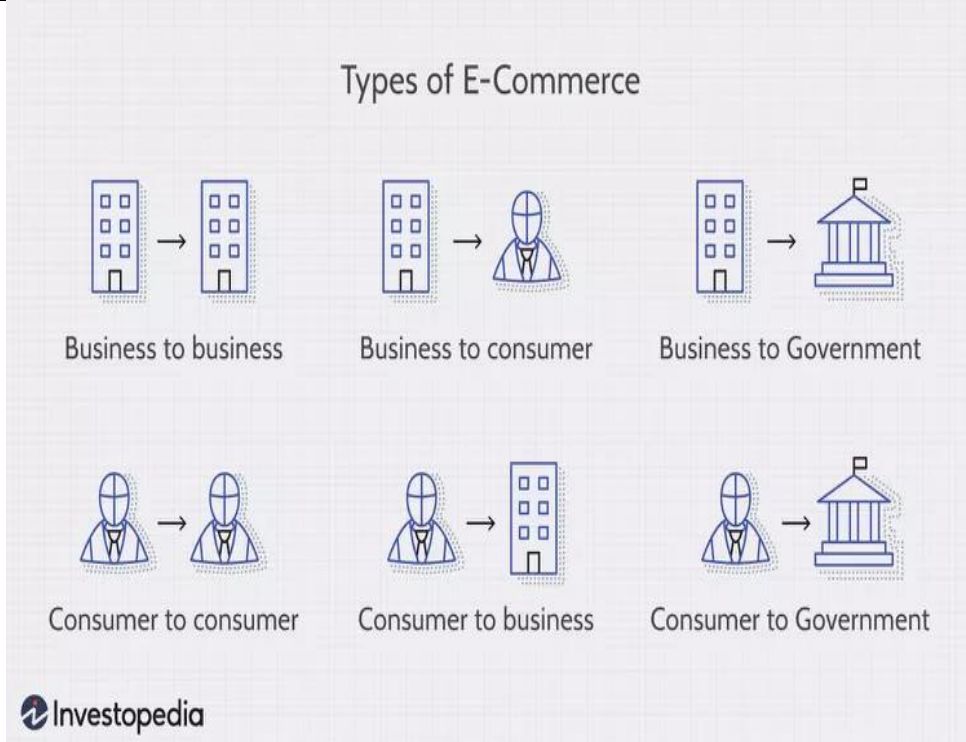
ទិន្នន័យដែលធ្វើការប្រមូលបានពី app លក់ស្បែកជើង Nike Sneaker App Store មានដូចខាងក្រោម៖

- ឈ្មោះរបស់ទំនិញ
- ប្រភេទរបស់ទំនិញ
- តម្លៃរបស់ទំនិញ
- ស្តុកទំនិញ , ការបញ្ចុះតម្លៃ
- របាយការណ៍លក់ផ្សេងៗ។

#### ១.២. ទ្រឹស្តីពាក់ព័ន្ធនឹង E-Commerce

<sup>1</sup>យោងតាមទ្រឹស្តីរបស់លោក Carter McNamara តាមរយៈសៀវភៅ Electronic commerce បានបោះពុម្ពផ្សាយនៅឆ្នាំ២០០១ បានបកស្រាយថា E-Commerce គឺសំដៅដល់ការទិញលក់ផលិតផលនិងសេវាកម្មរវាងម្ចាស់អាជីវកម្ម និងអ្នកទិញនៅលើប្រព័ន្ធអ៊ីនធឺណេត។ វាត្រូវបានដាក់អោយដំណើរការនៅលើប្រព័ន្ធ digital ជាច្រើនដូចជា: Computer, tablets, smartphones ជាដើម។

<sup>1</sup> Carter McNamara , Electronic commerce ,( បោះពុម្ពផ្សាយនៅឆ្នាំ២០០១ )។



រូបភាព 1: Type of E-Commerce

<sup>2</sup>តាមរូបភាពទី១ ខាងលើបានបង្ហាញអោយឃើញថា E-commerce មាន៦ប្រភេទធំៗដែលតម្រូវអោយក្រុមពួកយើងលើកយកតែ ៤ ចំណុចដែលសំខាន់ប៉ុណ្ណោះយកមកបកស្រាយ:

- Business to Business(B2B): ជាប្រភេទអាជីវកម្មដែលក្រុមហ៊ុនអាចជា អ្នកលក់ឬអ្នកទិញដូចគ្នាដែលអាចជា ម្ចាស់អាជីវកម្មផ្សេងទៀតមានគោលបំណង លក់ទំនិញនៅលើប្រព័ន្ធអ៊ីនធឺណែត។
- Business to Consumer(B2C): ជាប្រភេទអាជីវកម្ម ដែលអ្នកលក់ លក់ទំនិញទៅកាន់អតិថិជនដែលជាអ្នកប្រើផ្ទាល់នៅលើប្រព័ន្ធអ៊ីនធឺណែត។
- Consumer to Consumer(C2C): ជាប្រភេទអាជីវកម្ម ដែលអ្នកលក់និងអ្នកទិញសុទ្ធតែជាអ្នកប្រើប្រាស់ដូចគ្នា។
- Consumer to Business (C2B): ជាប្រភេទអាជីវកម្ម ប្រើប្រាស់ធ្វើការផ្គត់ផ្គង់ផលិតផលឬសេវាកម្មទៅកាន់ក្រុមហ៊ុន។

<sup>2</sup>Andrew Bloomenthal, E-Commerce, <https://www.investopedia.com/terms/e/ecommerce.asp#toc-types-of-e-commerce>(8/15/2023) ។

**១.៣. កម្មវិធី Microsoft និង Database**

**១.៣.១. ទ្រឹស្តីពាក់ព័ន្ធនឹង Microsoft**

Microsoft គឺជាក្រុមហ៊ុនផលិតSoftware ធំបំផុតនៅលើសកលលោក។ផលិតផលដែលបង្កើតឡើងដោយ Microsoft មានដូចជា : window operating system, Internet Explorer, Microsoft office, Visual studio code និងកម្មវិធីបង្កើតApplicationផ្សេងៗជាច្រើនទៀត។ ក្នុងនោះ: <sup>3</sup>Visual studio code ត្រូវបានបង្កើតឡើងដើម្បីឱ្យ developer អាចធ្វើការ develop Application បានដោយខ្លួនឯងនៃផ្នែកFront end របស់Application។ នៅក្នុង Visual studio code អាចឱ្យ developerប្រើភាសាជាច្រើនដូចជា : C, C++, C#, Java, JavaScript, Node.js, Flutter ជាដើម។ ក្នុងនោះដែរ Flutter ត្រូវបានជ្រើសរើសមកសរសេរបង្កើតប្រព័ន្ធសំណើនេះឡើងដោយប្រើប្រាស់ភាសា Programming Dart ។ ចំពោះទិន្នន័យផ្នែក Backend វិញត្រូវបានប្រើភាសា Node.js សរសេរលើកម្មវិធី PostgreSQL ដើម្បីរក្សាសុវត្ថិភាពនៃទិន្នន័យរបស់Userទៅក្នុង Database។

ដើម្បីអាចធ្វើការជាមួយFlutter ហើយនឹង Dartបាន យើងត្រូវការកម្មវិធីចំនួន៣ចាំបាច់គឺ:

- Visual Studio Code
- Android Studio
- Xcode

**១.៣.២. ប្រវត្តិរបស់ Microsoft Visual Studio Code ( VS Code )**

Vs Code គឺជា Source code editor ដែលបង្កើតឡើងដោយMicrosoft ជាមួយនឹងElectron Framework ហើយអាចប្រើបានជាមួយ platform ជាច្រើនដូចជា: Windows, Linux and macOS ។ Visual Studio Code ត្រូវបានប្រកាសអោយប្រើប្រាស់ដំបូងដោយ Microsoft នៅថ្ងៃទី29 April 2015។ នៅថ្ងៃទី 18 November 2015 source របស់Visual Studio code ត្រូវបានដាក់ក្រោមការគ្រប់គ្រងរបស់ MIT License នឹងអាចធ្វើការផ្ទុក Code នៅលើ GitHub បានផងដែរ។ Extension ដែលSupportក៏ត្រូវបានប្រកាសអោយប្រើប្រាស់នៅថ្ងៃទី 14 April 2015 ហើយបច្ចុប្បន្នVS Code មាន Version1.78។

**១.៣.៣. អំពី Android Studio**

Android Studio គឺជា IDE (Integrated Development Environment) ដែលបង្កើតឡើងដើម្បីធ្វើការអភិវឌ្ឍន៍ទៅលើកម្មវិធី Android ដែលបង្កើតឡើងដោយ googleដែលវាផ្អែកសំខាន់លើIntelliJ IDEA និង

<sup>3</sup>From Wikipedia, Visual Studio Code, (last update on 15 October 2023), [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code,\(8/15/2023\)](https://en.wikipedia.org/wiki/Visual_Studio_Code,(8/15/2023))

Java រួមបញ្ចូលគ្នាសម្រាប់សរសេរកម្មវិធី។វាអាចប្រើនៅលើPlatformជាច្រើនដូចជា: Windows, Linux and macOS។ Android Studio មានFeaturesជាច្រើនដូចជា៖

- មានbuilt-in emulatorដែលអនុញ្ញាតអោយdeveloperធ្វើការតេស្ត app ដោយអាច run app នៅលើVirtual deviceបាន
- មានLinters ដែលអាចជួយយើងក្នុងការស្វែងរក និង Fix errorនៅក្នុងCode
- មានCode Suggests ពេលយើងធ្វើការ type
- មានdebugger ដែលអនុញ្ញាតអោយអ្នកអាចមើលពីប្រតិបត្តិការអំពី Code របស់អ្នក។

**១.៣.៤. អំពី Xcode**

Xcode គឺជាការធ្វើការអន្តរកម្មរបស់ Apple development environment (IDE) សម្រាប់ macOS។ វាគឺរាប់បញ្ចូលទាំងអស់ដែលអ្នកអភិវឌ្ឍន៍ត្រូវការដើម្បីទៅបង្កើតកម្មវិធីដ៏ល្អសម្រាប់ Mac, iPhone, iPad, Apple TV, and Apple Watch។ Xcode បានផ្តល់អោយអ្នកអភិវឌ្ឍន៍កម្មវិធីនូវ workflow សម្រាប់ user interface design, coding, testing and debugging។

Xcode នៅក្នុង flutter គឺប្រើសម្រាប់សាងសង UI នៅក្នុង simulator ដែលអនុញ្ញាតអោយ developer អាចធ្វើការតេស្ត App នៅលើ virtual device នោះបាន។

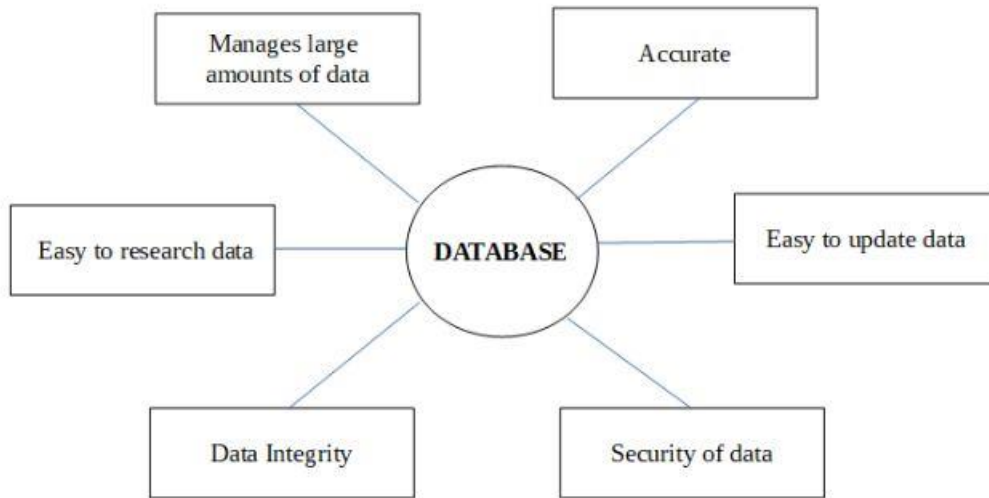
**១.៣.៥. ទ្រឹស្តីពាក់ព័ន្ធនឹងទិន្នន័យ Database**

**១.៣.៥.១. និយមន័យ Database**

Database គឺជាបណ្តុំនៃទិន្នន័យដែលបង្កើតឡើងក្នុងទម្រង់ជាTableដែលកើតចេញពី Row និង Column ដែលវាមានសារសំខាន់និងផ្តល់ភាពងាយស្រួលជាច្រើនក្នុងការគ្រប់គ្រងទិន្នន័យដូចជា ការទៅទាញយកទិន្នន័យនិងប្រើប្រាស់ទិន្នន័យ ឬការធ្វើការលុប (Delete), កែប្រែ (Update) ទៅលើទិន្នន័យណាមួយប្រកបដោយប្រសិទ្ធិនិងសុវត្ថិភាពហើយវាត្រូវបានរក្សាទុកនៅលើប្រព័ន្ធកុំព្យូទ័រ។

<sup>4</sup>Databaseផ្តល់ផលប្រយោជន៍ដូចខាងក្រោម៖

<sup>4</sup> TutorailsPoint, Database, (Copyright 2023),<https://www.tutorialspoint.com/Why-do-we-need-a-Database>(8,15,2023)



រូបភាព 2: Important of using Database

តាមរូបភាពទី២ ខាងលើនឹងបង្ហាញយើងពីសារៈប្រយោជន៍សំខាន់នៃការប្រើប្រាស់ប្រព័ន្ធជាមួយ database មានដូចខាងក្រោម៖

- គ្រប់គ្រងនិងផ្ទុកទិន្នន័យបានច្រើន
- មានការត្រួតពិនិត្យបានត្រឹមត្រូវរាល់ទិន្នន័យ ដែលប្រាកដថាគ្រប់ទិន្នន័យដែលរក្សាទុក ឬយកទៅប្រើប្រាស់គឺត្រឹមត្រូវពិតប្រាកដ
- ងាយស្រួលធ្វើការUpdate និង Research ទិន្នន័យដោយប្រើ Languages SQL
- មានសុវត្ថិភាព មុននឹងអាចចូលទៅប្រើប្រាស់ទិន្នន័យក្នុងdatabaseបានត្រូវតម្រូវអោយ Userមានសិទ្ធិLoginជាមុនសិនទើបអនុញ្ញាតអោយuserប្រើប្រាស់ទិន្នន័យនោះបាន
- Data integrity មានន័យថា database មានមុខងារជាច្រើនដូចជា backup, distribution and redundancy ដែលប្រាកដបានថាមិនមានការបាត់បង់នូវទិន្នន័យណាមួយឡើយ។

**១.៣.៥.២. អ្វីទៅជា Database Management System( DBMS ) ?**

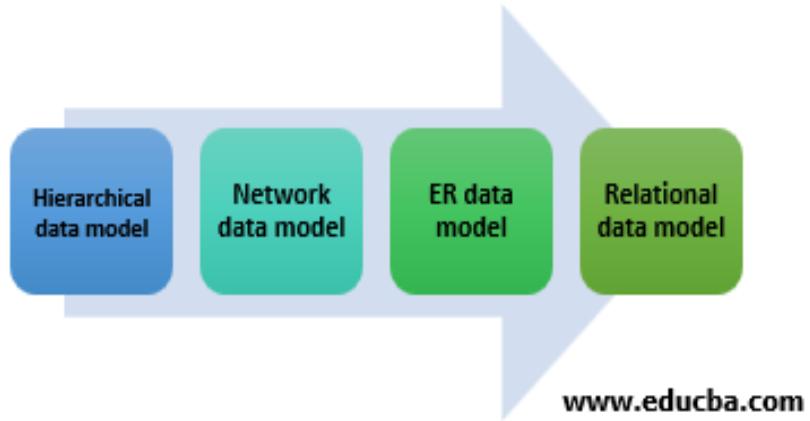
Database Management System (DBMS) សំដៅចំពោះ Software Application ដែលត្រូវបានប្រើសម្រាប់បង្កើតនិងថែរក្សា Database ព្រមទាំងផ្តល់នូវការត្រួតពិនិត្យទៅលើដំណើរការរបស់ database។ Database management System ដែលយើងអាចប្រើរួមមាន Microsoft Access, Ms SQL Server, MySQL, Oracle និង PostgreSQLជាដើម។

DBMS មាន៤ប្រភេទគឺ

- Hierarchical database management system



- Relational database management system
- Object oriented database management system
- Network database management system។



រូបភាព 3: Type of DBMS

តាមរូបភាពទី៣ ខាងលើនឹងធ្វើការបង្ហាញអ្នកអំពីប្រភេទសំខាន់ៗដែលមានបែងចែកក្នុង database management system មានដូចខាងក្រោម:

- relational database: គឺជាការរួមបញ្ចូលជាមួយនឹង table ជាច្រើននៃទិន្នន័យដែលកើតឡើងពី Colum and Row ដែលមានទំនាក់ទំនងជាមួយគ្នានឹងភ្ជាប់គ្នាដោយ special key នៃ fields ណាមួយ។ វាគឺមានភាពបត់បែនបានល្អ (flexible) ហើយវាមានផ្តល់ជា Functionality សម្រាប់ reading, creating, updating និងអាចធ្វើការ deleting ទិន្នន័យបានថែមទៀតផង។ Relational databases ប្រើប្រាស់ Structured Query Language (SQL)សម្រាប់ធ្វើការបង្កើត database ដែលផ្តល់ជាភាសា interface ដ៏ងាយស្រួលមួយសម្រាប់ធ្វើការទំនាក់ទំនងជាមួយ database ។
- hierarchical database model: គឺស្រដៀងនឹងរចនាសម្ព័ន្ធមែកធាង ដែលទំនាក់ទំនងរវាង record ទាំងអស់គឺមានលក្ខណៈ pre-defined ក្នុងទម្រង់ one to one manner (between parent and child nodes) ដែលគេតម្រូវអោយអ្នកប្រើប្រាស់ត្រូវប្រតិបត្តិតាមលំដាប់លំដោយដើម្បីចូលទៅប្រើទិន្នន័យដែលត្រូវការ។
- Network database models: ក៏មានទម្រង់ដូចនឹងរចនាសម្ព័ន្ធជាមែកធាងរបស់ hierarchical ដែរ ។ទោះជាត្រូវប្រើប្រាស់ជំនួសដោយ single-parent tree hierarchy ក៏ដោយក៏ model នេះ

supports many to many relationships ផងដែរ ដូច child tables អាចមានទំនាក់ទំនងជាមួយ parent table បានច្រើនទៀត។

- object-oriented databases (ER data model): ព័ត៌មាននៃទិន្នន័យទាំងអស់មានទម្រង់ជា object ។ ជាមួយនឹងប្រភេទផ្សេងៗគ្នានៃទំនាក់ទំនងអាចកើតមានរវាងពីរប្រភេទច្រើនជាងពីរ objects វា គឺដូចទៅនឹង databases ដែលប្រើប្រាស់ an object-oriented programming language សម្រាប់ ការអភិវឌ្ឍន៍(development) ។

DBMS មានតួនាទីនិងទំនួលខុសត្រូវដូចខាងក្រោម:

- Data Integrity: ប្រាកដថាទិន្នន័យនៅក្នុង database ទាំងអស់គឺជាទិន្នន័យត្រឹមត្រូវពិតប្រាកដ។
- Data Security: មានតួនាទីការពារសុវត្ថិភាពរបស់ database ដោយតម្រូវអោយតែអ្នកមានសិទ្ធិ ប៉ុណ្ណោះទើបត្រូវបានអនុញ្ញាតអោយចូលទៅប្រើប្រាស់ database នេះបាន ។ ចំពោះអ្នកប្រើ ប្រាស់នោះត្រូវមានការផ្ទៀងផ្ទាត់ទិន្នន័យជាមួយ Username និង password ជាមុនសិនទើប អាចចូលប្រើបាន បើអ្នកដែលមិនមានសិទ្ធិមិនត្រូវបានអនុញ្ញាតអោយចូលប្រើប្រាស់ database បានទេ ព្រោះវាជាបំពាន។
- Better data integration: វាផ្តល់នូវភាពងាយស្រួលក្នុងការគ្រប់គ្រងនិងតាមដានទិន្នន័យ របស់ម្ចាស់អាជីវកម្មទៅលើប្រព័ន្ធ។
- Minimized Data Inconsistency: កាត់បន្ថយភាពមិនស៊ីសង្វាក់គ្នារបស់ទិន្នន័យណាមួយ ដែលវាបញ្ជាក់បានថាមិនមានវត្តមានតែមិនត្រូវបានប្រើទៀតឡើយនៅក្នុង database ។ នៅ ពេល database មានការផ្លាស់ប្តូរទិន្នន័យ ទិន្នន័យនៅខាងផ្នែក user ទាំងអស់ក៏ត្រូវបាន ផ្លាស់ប្តូរភ្លាមៗផងដែរ។
- Faster Data Access: The database management system ជួយដល់អ្នកប្រើប្រាស់ ទាំងអស់នូវការធ្វើការឆ្លើយតបទៅនឹងតម្រូវការរបស់អតិថិជនបានឆាប់រហ័ស ដោយធ្វើការទៅ ទាញយកទិន្នន័យមកប្រើប្រាស់បានត្រឹមត្រូវនិងរហ័ស។
- Recovery and Backup: DBMS នឹងធ្វើការ recovery and backup ដោយស្វ័យប្រវត្តិ ទៅលើ database ដោយមិនមានការស្នើសុំពីUsersឡើយ ។ម៉្យាងវិញទៀតប្រសិនបើប្រព័ន្ធ មានបញ្ហាឬក៏គាំងវាមានតួនាទីទៅស្តារ(restore)ទិន្នន័យរបស់ប្រព័ន្ធពីមុនមកវិញ។

**១.៤. ទ្រឹស្តីពាក់ព័ន្ធនឹងភាសាបង្កើត Application**

**១.៤.១ Dart Programming**

<sup>5</sup>ភាសាសរសេរកម្មវិធី Dart ត្រូវបានបង្កើតឡើងដោយ Google នៅឆ្នាំ២០១១ដែលជាភាសាសរសេរកម្មវិធីទូទៅ ដែលដំបូងឡើងគោលដៅបង្កើតសម្រាប់សរសេរ Web, Server, Desktop ដែលក្រោយមកគេបានយកប្រើជាមួយនឹង Flutter SDK ។ បច្ចុប្បន្នភាសា Dart បាន Upgrade ដល់ version Null Safe ដែលអាចទប់មិនអោយមាន Null ទៅលើទិន្នន័យ ដែលអាចកាត់បន្ថយកុំអោយមាន Bug កើតមានក្នុងពេល Runtime។

គោលដៅរបស់វាគឺដើម្បីផ្តល់ជូននូវភាសាសរសេរកម្មវិធីដែលមានផលិតភាពបំផុតសម្រាប់ការអភិវឌ្ឍន៍ពហុធីទិក (Cross Platform) សម្រាប់បង្កើតកម្មវិធីដែលមានល្បឿនលឿននៅលើplatformណាមួយ។

**១.៤.២ តើអ្វីជា Flutter ?**

ជាទូទៅ ការបង្កើតកម្មវិធីទូរស័ព្ទគឺជាកិច្ចការដ៏ស្មុគស្មាញ និងពិបាក។ មាន framework ជាច្រើនដែលអាចរកបានដើម្បីអភិវឌ្ឍកម្មវិធីទូរស័ព្ទ។ ប្រព័ន្ធប្រតិបត្តិការ Android ផ្តល់នូវ framework ដើមដោយផ្អែកលើភាសា Java ហើយប្រព័ន្ធប្រតិបត្តិការ iOS ផ្តល់នូវ framework ដើមដោយផ្អែកលើភាសា Objective-C / Swift ។ ដូចនេះហើយបាន Flutter ដើរតួនាទីយ៉ាងសំខាន់ដែលជា Framework សម្រាប់បង្កើតកម្មវិធីទូរស័ព្ទបានទាំង២ OS ដោយផ្អែកទៅលើភាសាសរសេរកូដDart។

<sup>6</sup>Flutter គឺជា Open-Source SDK ដែលសម្រាប់សរសេរកម្មវិធីជាលក្ខណៈ អន្តរប្រតិបត្តិការ (Cross Platform) លើ IOS និងAndroid ដែលបង្កើតឡើងដោយGoogleនៅឆ្នាំ២០១៧។

Flutter មានសារសំខាន់ជាច្រើនដូចជា:

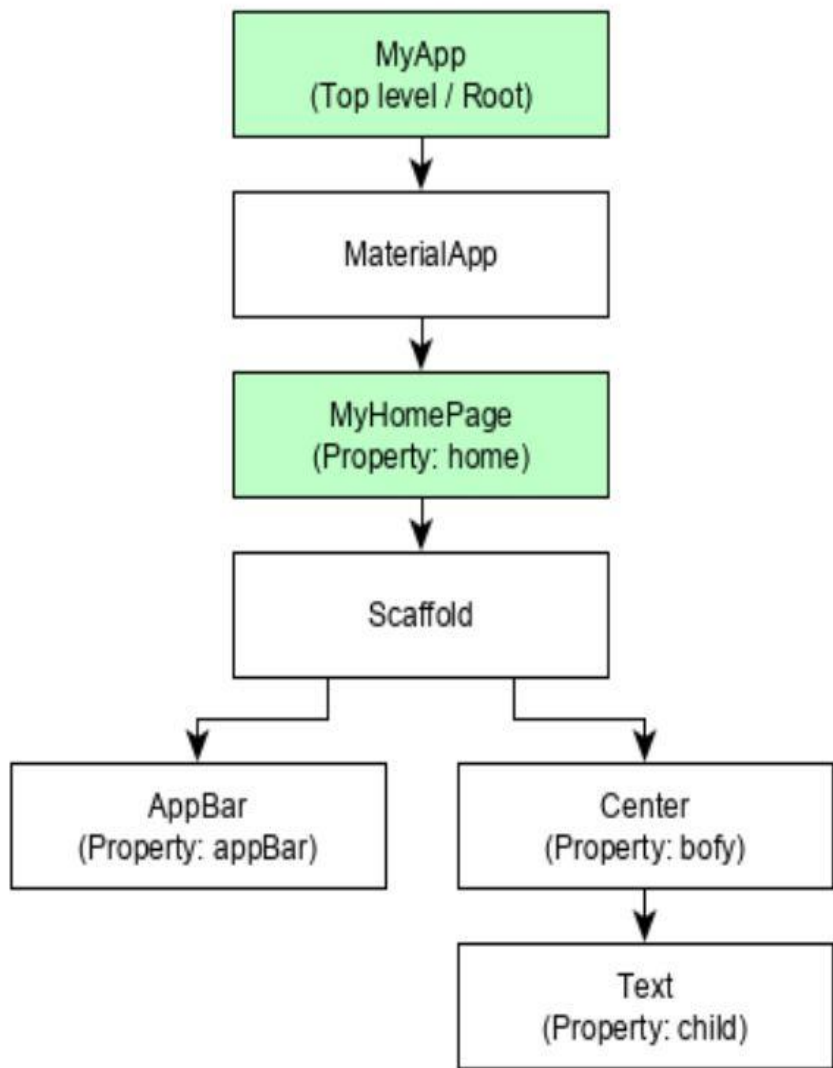
- អាចបង្កើតApp ជាលក្ខណៈកូដដើម(native code)សម្រាប់ IOS, Android, WebនិងDesktop App(MacOS, Window, Linux)
- ងាយស្រួលសរសេរកូដសម្រាប់ UI ដែលប្រើទម្រង់រចនា Material Design ជាងកូដដើម
- មាន Hot reload ដែលអាចផ្លាស់ប្តូរចំណុចតូចណាមួយដោយមិនចាំបាច់ rebuild
- អ្វីគ្រប់យ៉ាងគឺសុទ្ធតែកើតឡើងពី Widget។

<sup>5</sup> Oum Saokosal, Dart Language with Null Safety (ឆ្នាំ២០១០), ទំព័រទី២។

<sup>6</sup> TutorialsPoint , Flutter E-Book,( Copyright ឆ្នាំ២០១៩) ។

១.៤.២.១ Widgets

គោលគំនិតស្នូលនៃ Framework Flutter គឺនៅក្នុង Flutter អ្វីគ្រប់យ៉ាងគឺជា Widget ។ Widget ជាមូលដ្ឋានរបស់ User Interface components (សមាសធាតុរបស់UI) ដែលប្រើសម្រាប់បង្កើត User Interface(UI) របស់កម្មវិធី។ នៅក្នុងការបង្កើតកម្មវិធីរបស់ Flutter គឺវាកើតចេញពី widget របស់វាផ្ទាល់ដោយធ្វើការសាងសង់ឡើងដោយប្រើប្រាស់ children (Widget)មួយឬច្រើន។សម្រាប់ការរួមបញ្ចូលមុខងារទាំងនេះបានជួយយើងក្នុងការបង្កើតនូវ User Interface មួយដែលមានភាពទាក់ទាញ ។



រូបភាព 4: Architecture of Flutter Application

តាមរយៈរូបភាពទី4 ជា Architecture of Flutter Application គឺបានបង្ហាញពី ការកំណត់សម្គាល់ ខ្លះៗពី Diagram ខាងលើដូចជា:

- MyApp: គឺអ្នកប្រើប្រាស់ជាអ្នកបង្កើត widget នេះហើយវាត្រូវបានសាងសង់ឡើងដើម្បីប្រើ (native widget MaterialApp) ក្នុងFlutter ។
- MaterialApp: នៅក្នុងMaterialAppមាន home property ដែលសម្រាប់បញ្ជាក់មុខងាររបស់user Interface នៃ homepage (មានន័យថាhome ជាpropertyនៅក្នុង MaterialApp ហើយជាកន្លែងសម្រាប់ដាក់ class Homepage) ដែល widget MyHomePage គឺត្រូវបានសរសេរឡើងដោយ developer។
- My Homepage គឺត្រូវបានសាងសង់ឡើងដោយ native widget ផ្សេងៗទៀតនៅក្នុង Flutter នោះគឺ widget Scaffold ។ Scaffold មានproperties ចំនួន២សំខាន់ៗគឺ AppBar និង body។ body គឺប្រើសម្រាប់បញ្ជាក់ចំណុចគោលរបស់user interface ហើយ AppBar គឺប្រើសម្រាប់បញ្ជាក់អោយ Header របស់user interface។ បន្ទាប់មកទៀត Header Ui គឺត្រូវសាងសង់ឡើងដោយប្រើ flutter native widget AppBar ហើយ Body UI គឺត្រូវសាងសង់ឡើងដោយប្រើ Center widget។ Center widget មានproperty Child ដែលអាចដាក់ widget ផ្សេងៗទៀតដូចជា diagram ខាងលើគឺប្រើ Text widget ។

**១.៤.២.២. Features of Flutter**

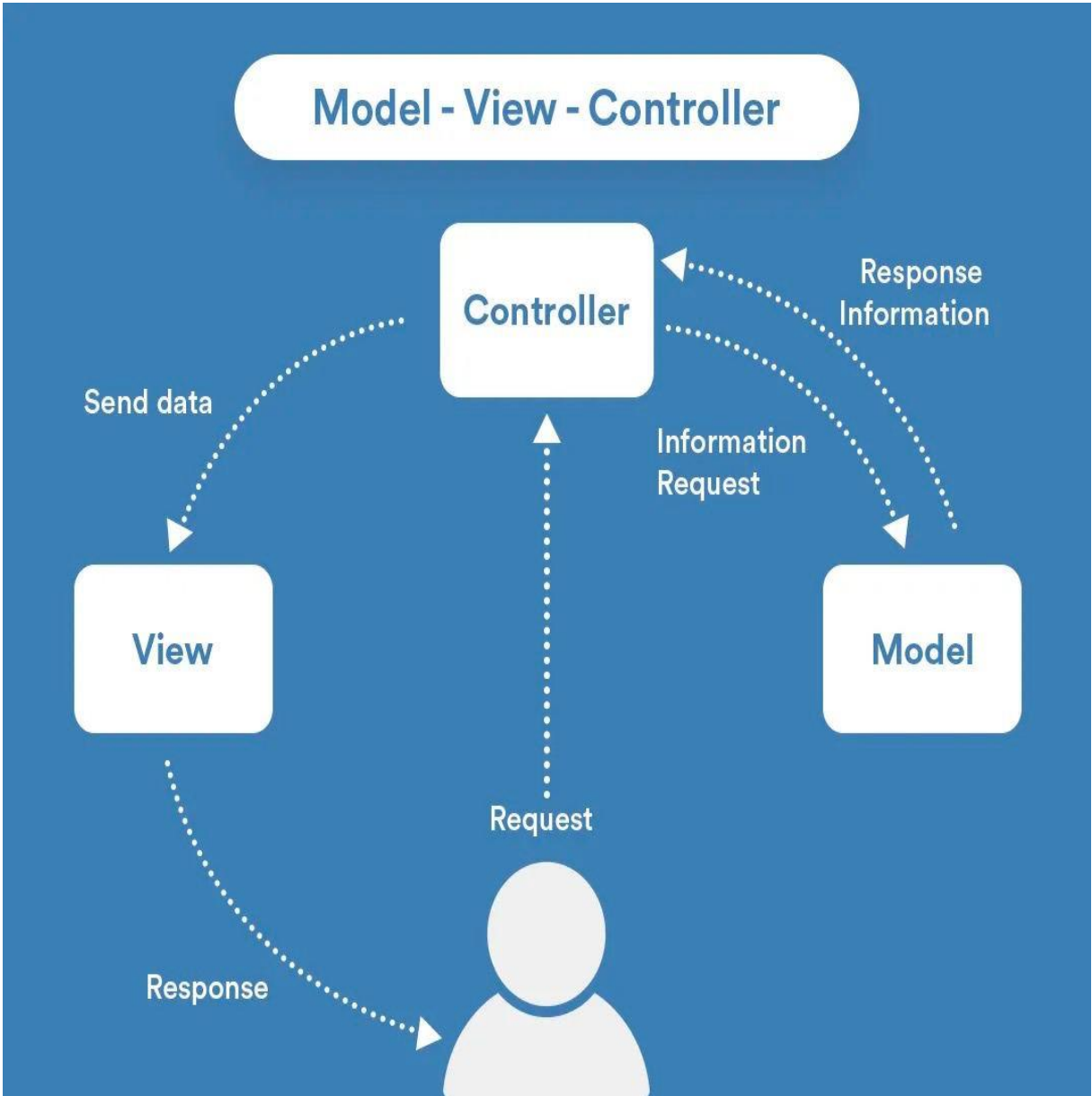
Flutter framework ផ្តល់នូវ feature ជាច្រើនដល់ developers ដូចជា៖

- ប្រើប្រាស់ភាសាDart ហើយវាគឺងាយស្រួលរៀន
- មាន framework ទំនើបៗ
- ទទួលបានការអភិវឌ្ឍន៍ឆាប់រហ័ស(Fast development)
- មានfeature សម្រាប់សាងសង់ UI ដ៏ស្រស់ស្អាតនិងទាក់ទាញ
- មានWidgetជាច្រើនសម្រាប់ប្រើក្នុងការសរសេរកូដ
- Runs ជាមួយ UI តែមួយសម្រាប់ platforms ជាច្រើនទៀត
- ជាកម្មវិធីមាន High performance។

**១.៤.២.៣. MVC Architecture Pattern**

Model-View-Controller (MVC) គឺជាសំណង់ Pattern ដែលបានបំបែកកម្មវិធីជា៣ Layers។

Model, View, and Controller គឺជា 3 layers នៃកម្មវិធី 1 components ទាំងអស់នោះធ្វើការគ្រប់គ្រងទៅលើផ្នែកជាក់លាក់នៃការអភិវឌ្ឍន៍កម្មវិធី។



រូបភាព 5: MVC Architecture Pattern

តាមរយៈរូបភាពទី៥: MVC Architecture Pattern បានបកស្រាយថាពេលអ្នកប្រើប្រាស់(User)ធ្វើការ sent request ទិន្នន័យវាត្រូវឆ្លងកាត់តាមរយៈ Controller ដែល Controller មានតួនាទីបញ្ជូនព័ត៌មាន ទិន្នន័យដែល request នោះទៅកាន់ផ្នែក Model បន្ទាប់ដើម្បីធ្វើការផ្ទៀងផ្ទាត់ទិន្នន័យ ហើយ Model ធ្វើការ

បញ្ជូនទិន្នន័យទាំងអស់នោះទៅ Controller វិញពេលដែលផ្ទៀងផ្ទាត់ត្រឹមត្រូវ Controller ក៏ធ្វើការបញ្ជូនទិន្នន័យនោះបន្តទៅកាន់ផ្នែក View ដោយ View ជាអ្នកបញ្ជូនទិន្នន័យមកអោយ User វិញ។

- Model: គឺជាស្នូលនៃលំហូរទិន្នន័យនៅក្នុងសំណង់កម្មវិធី។
- View: ការងាររបស់Viewគឺយកការបញ្ជូនទិន្នន័យពីUserហើយសម្រាប់បង្ហាញUI ទៅកាន់User។
- Controller: ធ្វើការងារគ្រប់គ្រងទៅលើlogicនៃbusiness ជាមួយការគ្រប់គ្រងលើការInput ទិន្នន័យ។

**១.៤.២.៤. State maintenance widgets**

នៅក្នុង Flutter គឺWidget ទាំងអស់គឺត្រូវទទួលបានចេញពីការ Class Stateless Widget or Stateful Widget ។ Widget ដែលប្រើក្នុង Stateless Widget គឺមិនមានការផ្លាស់ប្តូរទិន្នន័យណាមួយនោះទេ។ដើម្បីអោយកម្មវិធីមានភាព dynamic ទៅតាមធម្មជាតិរបស់កម្មវិធីយើងត្រូវប្រើប្រាស់នៅក្នុង Stateful Widget ព្រោះវាមាន state changes កំឡុងពេលធ្វើការផ្លាស់ប្តូរទិន្នន័យដែលវាអាចប្តូរទិន្នន័យបានភ្លាមៗដោយមិនចាំបាច់ ធ្វើការ re-rendering ម្តងទៀតទេ។

សរុបសេចក្តីមក Flutter Framework គឺជាFramework មួយដែលធ្វើការងារបានល្អទៅលើការបង្កើតកម្មវិធីទូរស័ព្ទដែលពិតជា Platformដែលមានគោលការណ៍ឯករាជ្យ។ តាមរយៈការផ្តល់នូវភាពសាមញ្ញក្នុងដំណើរការអភិវឌ្ឍន៍ (development process ), high performance នៅក្នុងកម្មវិធីទូរស័ព្ទដែលពាក់ព័ន្ធនឹង both user interface Android and IOS Platform។ Flutter Framework នឹងទៅបង្កើនអ្នកអភិវឌ្ឍន៍កម្មវិធីទូរស័ព្ទជាច្រើនបន្ថែមទៀតដើម្បីធ្វើការបង្កើតកម្មវិធីទូរស័ព្ទដែលមានលក្ខណៈ high performance និងមានមុខងារថ្មីបន្ថែមទៀតនាពេលអនាគតដ៏ខ្លីខាងមុខនេះ។

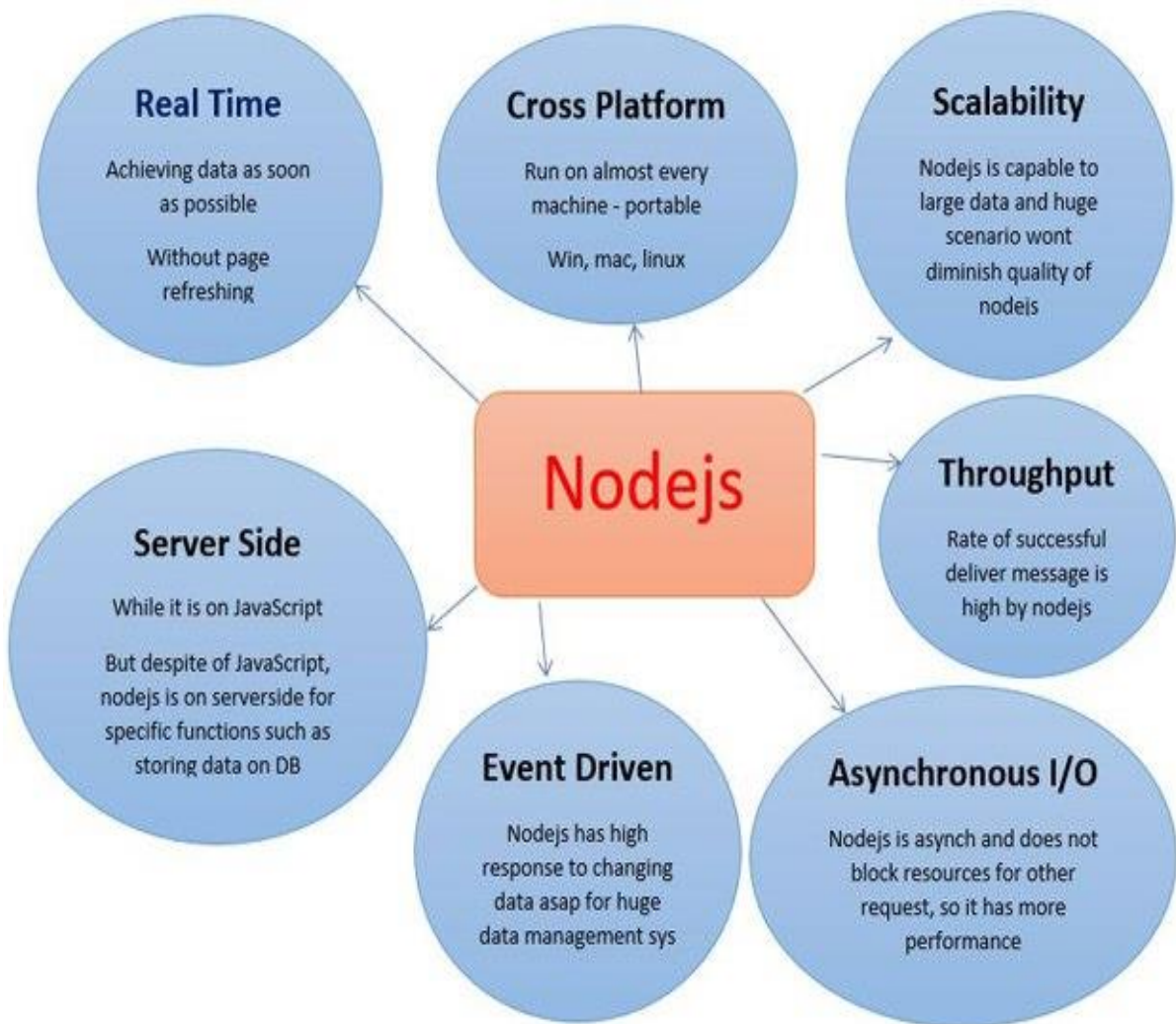
**១.៤.៣ ទ្រឹស្តី Node.js**

Node.js គឺជា Open-source ដែលមាន runtime environment ជាអន្តរកម្មប្រតិបត្តិការ (Cross-platform) សម្រាប់ executing JavaScript code ។ វាអាចសរសេរបង្កើតកម្មវិធីនៅលើ Platform ជាច្រើនដូចជា: Windows, Linux, Unix, macOS...។

Node.js គឺប្រើប្រាស់ សម្រាប់សរសេរភាសាផ្នែក Server ដែលវាអាចអោយ developers ប្រើភាសា JavaScript ទៅសរសេរផ្នែកផ្នែកកម្មវិធីបានទាំងផ្នែក client-side និង server-side ដោយមិនចាំបាច់ត្រូវរៀនភាសាថ្មីបន្ថែម។

Node.js គឺជាដំណើរការនៃផ្នែកខាងក្រោយ (back-end side) របស់ភាសាសរសេរកូដ JavaScript ។ ដែលវាធ្វើប្រតិបត្តិការនៅលើ V8 JavaScript Engine ហើយនិងធ្វើការទាញយកប្រតិបត្តិការរបស់កូដ JavaScript នោះមកបង្ហាញនៅលើ Web browser ។

ដោយហេតុយើងបានដឹងនិងស្គាល់រួចមកហើយថានៅក្នុងភាសាសរសេរឃ្លើត web មានភាសា ៣ ដែល developer ទាំងអស់ត្រូវតែស្គាល់គឺ HTML, CSS and JavaScript ។ JavaScript ត្រូវបានសម្គាល់ថាជា សកម្មភាព (behavior) ទាំងឡាយនៃប្រតិបត្តិការរបស់ web pages ។ JavaScript មានតួនាទីធ្វើអោយ Website មានភាពទាក់ទាញនិងធ្វើអោយ web ក្លាយជា web dynamic។ ហើយវាក៏មានតួនាទីយ៉ាងសំខាន់ធ្វើជា back-end side នៅក្នុង Node.js ដូចខាងលើផងដែរ។ ដូច្នេះហើយ Node.js មិនមែនជាភាសាសរសេរកូដ(Programming)ទេ គឺវាប្រើប្រាស់ភាសាសរសេរកូដ JavaScript សម្រាប់សរសេរឃ្លើតផ្នែក server side។



រូបភាព 6: Important of using Node.js



តាមរយៈរូបភាពទី៦ ខាងលើនឹងធ្វើការប្រាប់បង្ហាញពីចំណុចសំខាន់ៗមួយចំនួននៃការប្រើប្រាស់ Node.js

នៅក្នុងប្រព័ន្ធសំណើរខាងលើដូចខាងក្រោម៖

- Real-Time-Communication

Real Time web solution គឺជា web technologies ដែលអាចអោយ Client អាចទទួលបានព័ត៌មានបានភ្លាមៗពីអ្នកប្រើប្រាស់នៅពេលមានទិន្នន័យថ្មីឬអត់មានដោយគ្រាន់តែធ្វើការ refreshing page។

- Cross Platform

Cross platform មានន័យថា Nodejs គឺជា platform ដែលជាទូទៅឯករាជ្យហើយអាចធ្វើការតម្លើងបានគ្រប់ម៉ាស៊ីនដែលមានប្រព័ន្ធដំណើរការផ្សេងៗគ្នាដូចជា Windows, Linux, Macintosh។ដូច្នោះ Nodejs មិនត្រូវការការរៀបចំជាក់លាក់ទេ ហើយអាចចល័តបាន។

- Server-Side Application

នៅក្នុងស្ថានភាពមួយដែលមិនអាចទៅរួចនោះគឺ Client មិនអាចធ្វើការផ្តល់ទិន្នន័យដោយផ្ទាល់ទៅអោយគ្នាបានទេ។ ក្នុងករណីនេះយើងត្រូវការរក្សាទុកទិន្នន័យរបស់យើងនៅក្នុង database ឬធ្វើការបង្កើតដំណើរការពិសេសមួយចំនួននៅលើ program ហើយយើងត្រូវដាក់ functionalities ទាំងអស់នោះនៅលើ Server ដែលអាចធ្វើការផ្លាស់ប្តូរទិន្នន័យគ្នាពី Client ទៅ server ទៅតាមគោលបំណងជាក់ស្តែង។ ហេតុនេះហើយបាន Nodejs បានដើរតួនាទីសំខាន់ធ្វើជាផ្នែកserver ដើម្បីអាចផ្លាស់ប្តូរទិន្នន័យពី client side បាន។

- Event Driven Architecture

ប្រសិនបើយើងធ្វើការជាមួយ data management system មួយចំនួនដែលមានការកំណត់លក្ខខណ្ឌទៅតាម status នីមួយៗ។ ដែលពេលយើងត្រូវការស្វែងរកទិន្នន័យនៃ status ណាមួយវានឹងឆ្លើយតបបានត្រឹមត្រូវ។ ដូច្នោះបើមានទិន្នន័យណាមួយដែល sold នោះទិន្នន័យទាំងនោះនឹងត្រូវផ្លាស់ប្តូរ Status ដោយខ្លួនវា។

- Asynchronous I/O

គឺមានវិធីសាស្ត្រចំនួន២ សម្រាប់ I/O ដែលជាធម្មតាគឺជា វិធីសាស្ត្រដែលប្រើសម្រាប់ Blocks resources and progress រហូតដល់ទំនាក់ទំនងត្រូវបានបំពេញរួចរាល់ដែលអាចចំណាយពេលវេលារងចាំច្រើនជាពិសេសប្រសិនបើយើងមាន I/O ច្រើន។ មានវិធីសាស្ត្រមួយទៀតដែលអនុញ្ញាតអោយ Critical operations ធ្វើការងាររបស់ពួកគេនៅកំឡុងពេលវាកំពុងរងចាំI/O។

- Throughput

នៅក្នុង scenario ដែលយើងត្រូវការផ្ញើនិងទទួលសារ (send and receive messages) វាមានភាពដែលអាចទៅរួចដែលសារមួយចំនួនដែលមិនអាចផ្ញើចេញបាន ហើយសារដែលផ្ញើចេញបានលុះត្រាតែជាសារដែលទទួលបានត្រឹមត្រូវ ដែលវាត្រូវបានហៅចេញនៅលើ Channel Throughput។

- Scalability

Scalability គឺជាសមត្ថភាពរបស់កម្មវិធីដែលពង្រីកបាននៅពេលដែលជួបជាមួយ scenario ធំៗមានដូចជា ចំនួនទិន្នន័យដែលផ្ទុកច្រើន ឬក៏ app មានប្រើ async await ច្រើនយ៉ាងណាក៏ដោយក៏វាមិនធ្វើអោយ app slow ដែរ។ ដូច្នេះការកើនឡើងដែលមិនរំពឹងទុកពីបរិមាណនឹងមិនថយចុះគុណភាពនៃប្រព័ន្ធទេ។

**១.៤.៤. ទ្រឹស្តី PostgreSQL**

<sup>7</sup> តាមរយៈសៀវភៅ Beginning Databases with PostgreSQL 2<sup>nd</sup> edition របស់លោក Neil Matthew and Richard Stones បានបោះពុម្ពផ្សាយនៅឆ្នាំ២០០៥ បានបង្ហាញថា PostgreSQL គឺជា DBMS (Database Management System) ដែលមានការរួមបញ្ចូលជាមួយនឹង relational model សម្រាប់បង្កើតdatabase របស់វាដោយប្រើប្រាស់ភាសា SQL(standard query language )។ វា run នៅលើ Platform ជាច្រើនដូចជា Window, Linux , macOS...។ វាមានសមត្ថភាពគ្រប់គ្រងទិន្នន័យបានធំទូលាយនិងប្រើប្រាស់ដោយឥតគិតថ្លៃថែមទាំងមាន featured ជាច្រើនដែលផ្តល់ភាពងាយស្រួលក្នុងការប្រើប្រាស់សម្រាប់ developer ជាពិសេស។

PostgreSQL ត្រូវបានគេស្គាល់ថាអាចប្រើប្រាស់ជាមួយភាសាសរសេរកូដជាច្រើនដូចជា C, C++, Perl, Python, java, Node.js, PHP និងភាសាសរសេរកូដជាច្រើនទៀតផងដែរ។

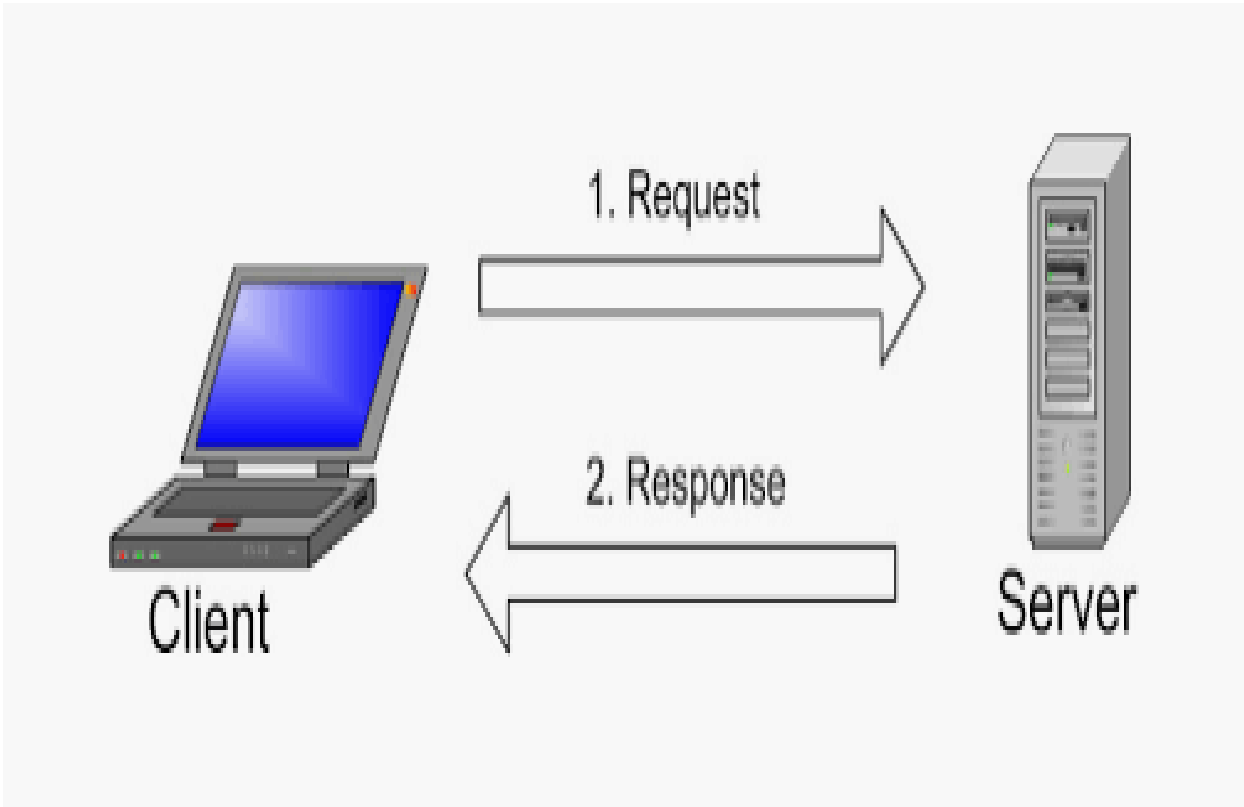
- **How does PostgreSQL work ?**

PostgreSQL ប្រើប្រាស់ client-server model ដែលផ្នែកclient និងផ្នែក server អាចផ្លាស់ប្តូរទំហំគ្នានៅលើhostsផ្សេងៗគ្នាក្នុងដំណាក់កាលរបស់ប្រព័ន្ធណីតិវិធី។ផ្នែកភាសាខាង server ធ្វើការគ្រប់គ្រងទៅលើ files របស់ database និងធ្វើការទទួលនូវការភ្ជាប់ទិន្នន័យទៅ database តាមរយៈ client application មានន័យថាភាសាដែលសរសេរបង្កើតផ្នែកServerមានតួនាទីគ្រប់គ្រងទៅលើ files ទាំងអស់ដែលបានបង្កើតនៅក្នុង database និងមានតួនាទីសំខាន់ៗទៀតគឺធ្វើការភ្ជាប់ទំនាក់ទំនងរវាង database ទៅកាន់ Client side

<sup>7</sup> Neil Matthew និង Richard Stones, Beginning Databases with PostgreSQL 2<sup>nd</sup> edition (2005) ។

នៅពេលដែលមានព័ត៌មានឬទិន្នន័យមកពីផ្នែក Client Application វាត្រូវបានធ្វើការបញ្ជូនទិន្នន័យឬព័ត៌មានទាំងនោះទៅកាន់ផ្នែក server side ដែលមាន database ជាអ្នកគ្រប់គ្រងនិងរក្សាទុកទិន្នន័យនៅទីនោះ ដើម្បីប្រាកដថាទិន្នន័យរក្សាបាននូវសុវត្ថិភាពល្អ។

PostgreSQL មានសមត្ថភាពដោះស្រាយបញ្ហានៅពេលដែលមានទិន្នន័យជាច្រើនធ្វើការបញ្ជូនស្របពេលគ្នាពីផ្នែក Client ដោយវិធីសាស្ត្រ Forking ទិន្នន័យទាំងនោះ មានន័យថាវានឹងទៅបង្កើតដំណើរការថ្មីផ្សេងៗទៀតដើម្បីទៅទទួលទិន្នន័យទាំងនោះពីផ្នែក Client ទៅតាមទិន្នន័យនីមួយៗ។

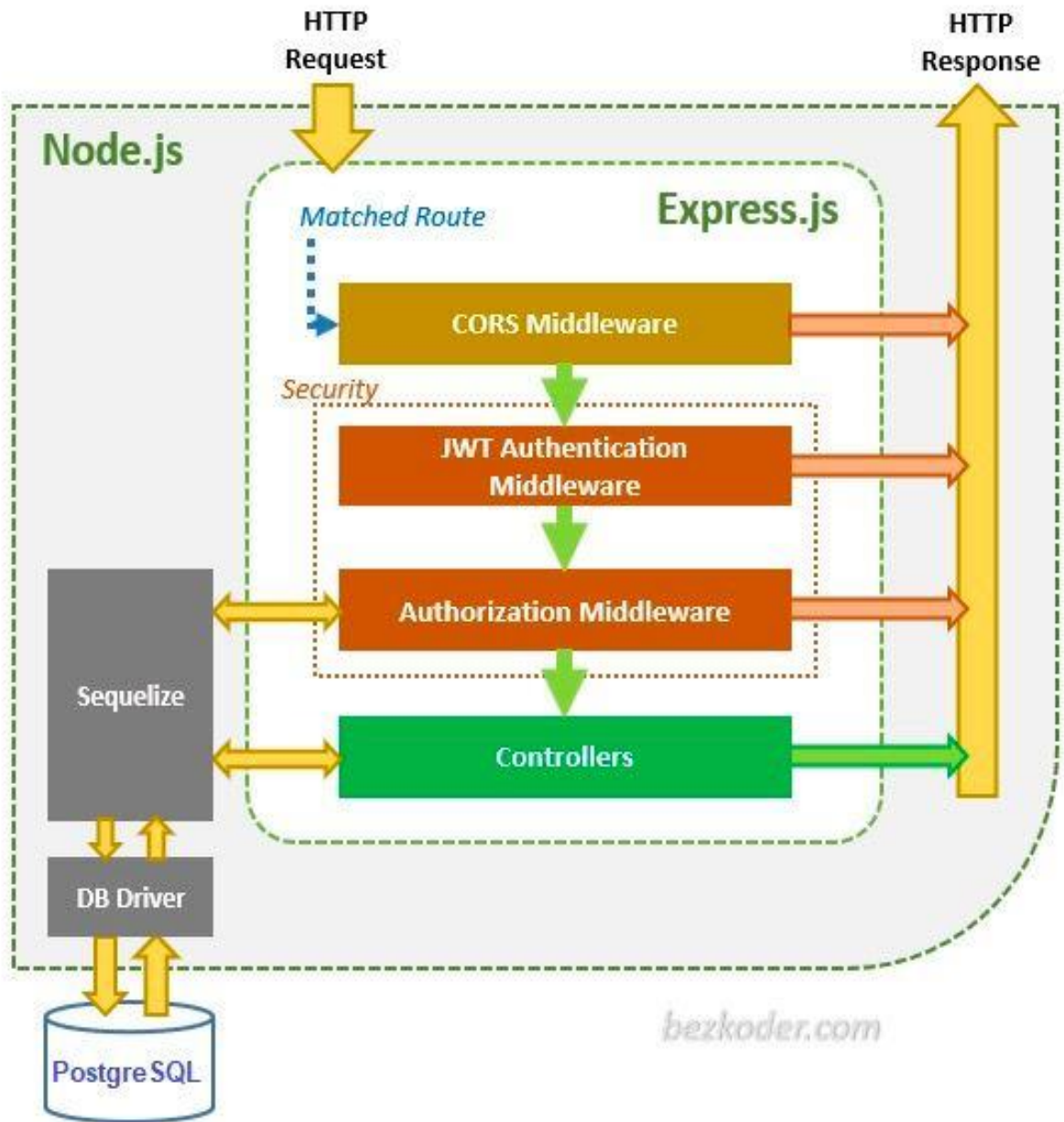


រូបភាព 7: How client side work with server side

តាមរយៈរូបភាពទី៧ ខាងលើនេះបានបង្ហាញពីដំណើរការរវាង Client side និង Server side នៅពេលដែលមានការធ្វើ request ពីផ្នែក client ទៅកាន់Database PostgreSQL ដើរតួនាទីជាអ្នកនាំ request ពី client side ទៅផ្នែកdatabase នឹងធ្វើការបញ្ជូនទិន្នន័យមកវិញ (response) ពីផ្នែក database ទៅកាន់ផ្នែក Client វិញ ។

<sup>8</sup> Dade2, PostgreSQL, (June 17, 2020), [https://dade2.net/what-is-postgresql-how-does-postgresql-work/\(8/15/2023\)](https://dade2.net/what-is-postgresql-how-does-postgresql-work/(8/15/2023))

ដំណើរការរវាង Client side (Node.js) ប្រើប្រាស់និង Server side (PostgreSQL)



រូបភាព 8: How Node.js work with PostgreSQL

<sup>9</sup>តាមរយៈរូបភាពទី៨ ខាងលើដែលធ្វើការបង្ហាញអំពីដំណើរការរបស់ Node.js និង PostgreSQL បានរៀបរាប់ថា: HTTP Request ត្រូវបានបញ្ជូនទៅផ្នែករបស់ Matches Route ដែលផ្នែកនេះត្រូវបានធ្វើការត្រួតពិនិត្យដោយ CORS Middleware មុនពេលត្រូវបានបញ្ជូនទៅបន្តមកដល់ Security layer ។ នៅក្នុង Security layer មានស្រទាប់layer ចំនួន២គឺ:

1. JWT Authentication Middleware: វាមានតួនាទី verify SignUp, verify token
2. Authorization Middleware: វាមានតួនាទីត្រួតពិនិត្យតួនាទីរបស់ User ជាមួយ recode នៅក្នុង database។









ប្រសិនបើ middlewares មានបញ្ហា error នឹងមានការបញ្ជូនសារជា HTTP response ទៅកាន់ client វិញ។ ក្នុងករណីដែល middleware ត្រឹមត្រូវវានឹងបញ្ជូនទៅឲ្យ controller ធ្វើការផ្ទៀងផ្ទាត់ជាមួយនឹង PostgreSQL Database តាមរយៈ Sequelize ហើយធ្វើការផ្ញើជា HTTP response (token, user information, data based on roles...) ទៅកាន់ client ។

### ១.៥ ទ្រឹស្តីពាក់ព័ន្ធនិងវិធីសាស្ត្រនៃការវិភាគប្រព័ន្ធ

#### ១.៥.១. Work Flow Diagram

Workflow diagram ត្រូវបានគេស្គាល់ថាជា គំនូសតាងលំហូរដំណើរការ ឬគំនូសតាងលំហូរការងារ គឺជាការបង្ហាញរូបភាពនៃលំដាប់នៃជំហាន ឬសកម្មភាពដែលពាក់ព័ន្ធនឹងការបំពេញកិច្ចការ ដំណើរការ របស់គម្រោង។ វាផ្តល់នូវទិដ្ឋភាពទូទៅច្បាស់លាស់ និងមានរចនាសម្ព័ន្ធអំពីរបៀបដែលការងារដំណើរការពីដំណាក់កាលមួយទៅដំណាក់កាលមួយទៀត ដោយគូសបញ្ជាក់ពីទំនាក់ទំនងរវាងកិច្ចការផ្សេងៗដែលទាក់ទងនឹងការសម្រេចចិត្ត និងលទ្ធផលជាដើម។

<sup>9</sup> Tien Nguyen, Node.js Express Login example with PostgreSQL, (last update on 27 May 2021), [https://dev.to/tienbku/node-js-express-login-example-with-postgresql-5676\(8/15/2023\)](https://dev.to/tienbku/node-js-express-login-example-with-postgresql-5676(8/15/2023))





	<p>It is used to indicate the start and the end of the flowchart.</p>
	<p>It is used to represent a data input (reading).</p>
	<p>It is used to represent processes, such as arithmetic operations, value assignments in memory, etc.</p>
	<p>It is used to indicate the flow direction of the diagram.</p>
	<p>It is used to indicate a decision process. The condition to be compared is described within the form.</p>
	<p>It is used to indicate the output (printing) of data.</p>
	<p>It is used to indicate a cycle of repetitions of an action.</p>
	<p>It is used to indicate the continuation of flow of the diagram within a same page.</p>

តារាង 1: និមិត្តសញ្ញា Work Flow Diagram

១.៥.២. Data Flow Diagram (DFD)

Data flow diagram (DFD) គឺជាឧបករណ៍ក្រាហ្វិកដែលអនុញ្ញាតឱ្យអ្នកវិភាគប្រព័ន្ធ (និងអ្នកប្រើប្រាស់ប្រព័ន្ធ) ពិពណ៌នាអំពីលំហូរទិន្នន័យនៅក្នុងប្រព័ន្ធព័ត៌មាន។ DFD គឺជាវិធីសាស្ត្រមួយក្នុងចំណោមវិធីសាស្ត្រដែលអ្នកវិភាគប្រព័ន្ធប្រើប្រាស់ដើម្បីប្រមូលព័ត៌មានចាំបាច់ដើម្បីកំណត់តម្រូវការប្រព័ន្ធព័ត៌មាន។

# DFD Symbols and Definitions

<b>Process</b>		<ul style="list-style-type: none"> <li>• Process - performs some action on data, such as creates, modifies, stores, delete, etc. Can be manual or supported by computer.</li> </ul>
<b>Data store</b>		<ul style="list-style-type: none"> <li>• Data store - information that is kept and accessed. May be in paper file folder or a database.</li> </ul>
<b>External Entity</b>		<ul style="list-style-type: none"> <li>• External entity - is the origin or destination of data. Entities are external to the system.</li> </ul>
<b>Data flow</b>		<ul style="list-style-type: none"> <li>• Data flow - the flow of data into or out of a process, data store or entity</li> </ul>

តារាង 2 : និមិត្តសញ្ញា Data Flow Diagram

- Process : ជាអ្នកដំណើរការលើសកម្មភាពរបស់ទិន្នន័យដូចជា Creates, Modifies, Stores និង Deleting ទិន្នន័យជាដើម។
- Data Store: ជាកន្លែងដែលរក្សាទុកព័ត៌មាននិងជាកន្លែងដែល User ចូលទៅទាញយកទិន្នន័យមកប្រើប្រាស់ជាពិសេសក្នុង database។
- External Entity: ជាអង្គភាពដែលមានពាក់ព័ន្ធនឹងការផ្តល់នូវទិន្នន័យ។
- Data flow: មានតួនាទីប្រាប់ពីទិសដៅរបស់ទិន្នន័យទៅតាមដំណើរការទិន្នន័យមួយទៅទិន្នន័យផ្សេងៗទៀត។

## ជំពូកទី២

### វិធីសាស្ត្រនិងចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ

ក្នុងជំពូកនេះត្រូវបានលើកយកនូវវិធីសាស្ត្រនៃការសិក្សាស្រាវជ្រាវ មកសិក្សាបង្ហាញដើម្បីបន្ថែមលក្ខណៈ ច្បាស់លាស់ និងដើម្បីឲ្យដំណើររបស់ប្រធានបទខាងលើឲ្យកាន់រលូន និងមានភាពល្អប្រសើរ។

#### ២.១ វិធីសាស្ត្រនៃការសិក្សាស្រាវជ្រាវ

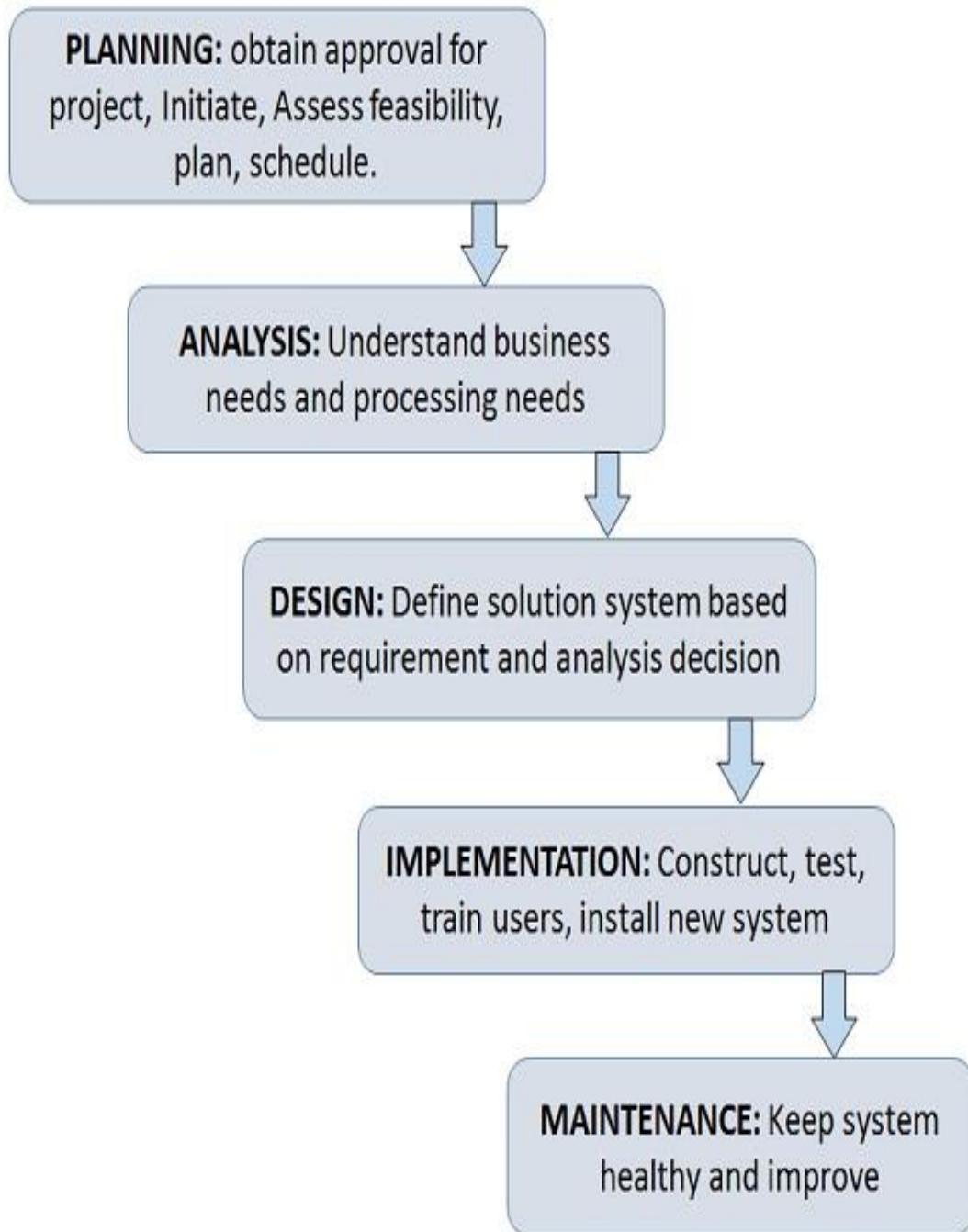
វដ្តនៃការអភិវឌ្ឍន៍ប្រព័ន្ធដែលមានប្រសិទ្ធភាព (SDLC) គួរតែជាលទ្ធផលនៃប្រព័ន្ធដែលមានគុណភាពខ្ពស់ ដែលបំពេញតាមការរំពឹងទុករបស់អតិថិជន ឈានដល់ការបញ្ចប់ក្នុងរយៈពេលកំណត់ និងវាយតម្លៃការចំណាយ ហើយដំណើរការប្រកបដោយប្រសិទ្ធភាព និងប្រសិទ្ធភាពនៅក្នុងហេដ្ឋារចនាសម្ព័ន្ធបច្ចេកវិទ្យាព័ត៌មានបច្ចុប្បន្ន និង ដែលបានគ្រោងទុក។

ដើម្បីឲ្យកាន់តែយល់ច្បាស់អំពីដំណើរការរបស់ប្រព័ន្ធ ការសិក្សាត្រូវបានគោរពតាមវិធីសាស្ត្រ <sup>10</sup> SDLC (System Development Life Cycle) តាមរយៈ website របស់ Tutorials Point បាននិយាយថា System Development Life Cycle គឺជាគំរូគំនិតដែលរួមបញ្ចូលទាំងគោលការណ៍ និងនីតិវិធីសម្រាប់ការអភិវឌ្ឍ ឬការកែប្រែ ប្រព័ន្ធពេញជារដ្ឋនៃការអភិវឌ្ឍន៍ប្រព័ន្ធ។ SDLC ត្រូវបានប្រើប្រាស់ដោយអ្នកវិភាគដើម្បីបង្កើតប្រព័ន្ធព័ត៌មាន (Information System) ។ ដូច្នេះសរុបសេចក្តីមក Systems Development Life Cycle គឺជាវិធីសាស្ត្រជាប្រព័ន្ធដែល បំបែកការងារយ៉ាងច្បាស់លាស់ទៅជាដំណាក់កាលដែលត្រូវឱ្យអនុវត្តលើប្រព័ន្ធព័ត៌មានថ្មី ឬកែប្រែប្រព័ន្ធព័ត៌មាន នោះ។

តាមរយៈរូបភាពទី៩ ជា Phases of SDLC ខាងក្រោមនេះនឹងបង្ហាញអ្នកអំពីដំណាក់កាលនីមួយៗរបស់ SDLC :

<sup>10</sup>Tutorials point, System Development Life Cycle, (Copyright 2023), [https://www.tutorialspoint.com/system\\_analysis\\_and\\_design/system\\_analysis\\_and\\_design\\_development\\_life\\_cycle.htm](https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_development_life_cycle.htm)(8/15/2023)





រូបភាព 9: Phases of SDLC

**២.១.១. System Planning**

ជាដំណាក់កាលដែលប្រព័ន្ធត្រូវធ្វើគម្រោងដើម្បីដាក់ស្នើទៅលើប្រព័ន្ធថ្មី ក្នុងគោលបំណងដោះស្រាយនូវបញ្ហាដែលប្រព័ន្ធបច្ចុប្បន្នកំពុងជួបប្រទះ ដោយត្រូវមានការរៀបចំពេលវេលា និងគោលបំណងឲ្យបានត្រឹមត្រូវទៅតាមទំហំប្រព័ន្ធដែលត្រូវសិក្សា។

**២.១.២. System Analysis**

ជាដំណាក់កាលដែលយើងត្រូវធ្វើការប្រមូលទិន្នន័យយកមកវិភាគនិងផ្ទៀងផ្ទាត់ឲ្យបានម៉ត់ចត់ទៅលើគោលបំណងរបស់អ្នកប្រើប្រាស់ឲ្យបានច្បាស់លាស់ដើម្បីវាយតម្លៃទៅលើអាទិភាពនៃតម្រូវមួយណាដែលចាំបាច់ ដែលផ្តល់ភាពងាយស្រួលក្នុងការកំណត់នូវតម្រូវការនិងគំរូសម្រាប់ប្រព័ន្ធថ្មី។ ក្នុងដំណាក់កាលនេះគឺចង់ឲ្យយើងយល់និងដឹងច្បាស់ពីគោលបំណងរបស់ប្រព័ន្ធដែលត្រូវសិក្សា និង ធ្វើការវិភាគពីដំណើរការរបស់ប្រព័ន្ធទៅតាមតម្រូវរបស់អ្នកប្រើប្រាស់ទាំងនោះជាមុនសិនមុននឹងបង្កើតប្រព័ន្ធ ។

**២.១.៣. System Design**

ជាដំណាក់កាលកំណត់នូវដំណោះស្រាយដល់ប្រព័ន្ធដោយផ្ដោតទៅលើតម្រូវការនិងការវិភាគសម្រេចរបស់ System Analysis ក្នុងគោលបំណងលើកយកករណីភាគរបស់ប្រព័ន្ធមកបង្ហាញជាកំរូបភាពគ្រោង Interface Design ។

**២.១.៤. System Implementation**

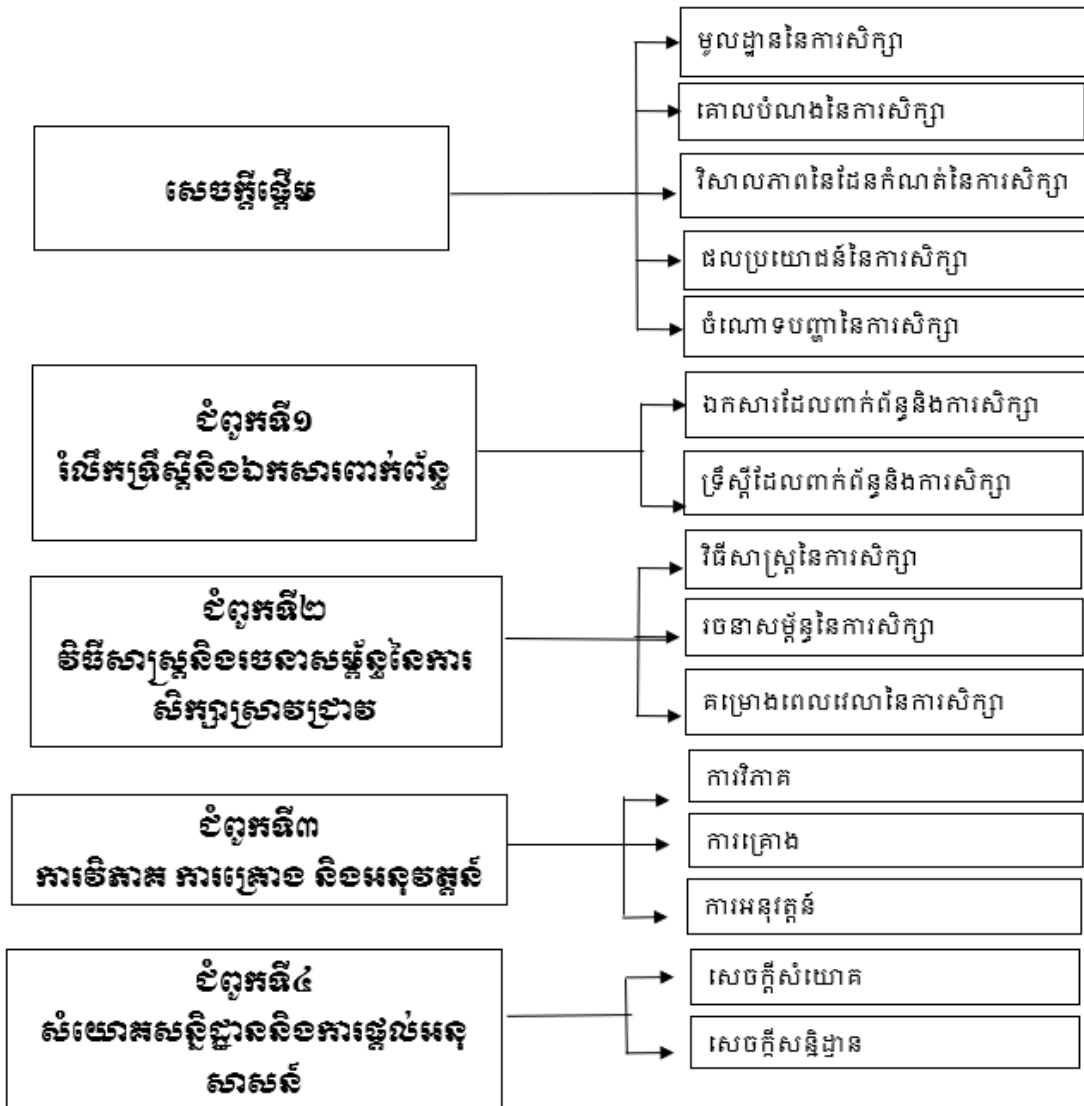
ជាដំណាក់កាលដែល Application ត្រូវបានបង្កើតឡើង និងមានការតេស្តសាកល្បង ដើម្បីស្វែងរកកំហុសឆ្គងមុននឹងដាក់ឲ្យ User ប្រើប្រាស់។ ក្នុងដំណាក់កាលនេះប្រព័ន្ធត្រូវបានបង្កើតឡើងទៅតាមគម្រោងនៃការស្នើរខាងលើនិងត្រូវដាក់អោយប្រើប្រាស់ជាសាធារណៈបានបន្ទាប់ពីក្រុមការងារបានដំឡើង Application នោះនៅលើ Play Store or AppStore។

**២.១.៥. System Maintenance**

ជាដំណាក់កាលចុងក្រោយរបស់ Application គ្រាន់តែធ្វើការត្រួតពិនិត្យបន្ថែមកុំឲ្យមានបញ្ហាកើតឡើងពេលដាក់អោយប្រើប្រាស់ និងធ្វើការ Improve បន្ថែមផងដែរសិនបើមានបញ្ហាណាមួយកើតមានដើម្បីធានាសុវត្ថិភាពរបស់អ្នកប្រើប្រាស់និងប្រសិទ្ធភាពរបស់ App។

**២.២ រចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ**

រចនាសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវលើ App លក់ស្បែកជើងឈ្មោះថា “Nike sneaker App Store” ត្រូវបែងចែកជា៣ជំពូកដូចខាងក្រោម៖



រូបភាព 10: របបសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ

ចំពោះរបបសម្ព័ន្ធដែលបានបង្ហាញតាមរយៈ រូបភាពទី10 បានបញ្ជាក់អោយយើងឃើញពីអត្ថន័យសង្ខេប ដូចខាងក្រោម៖

**២.២.១ សេចក្តីផ្តើម**

ក្នុងជំពូកនេះបានបង្ហាញពីអត្ថប្រយោជន៍ជាច្រើននៃការប្រើប្រាស់ប្រព័ន្ធខ្លីដីថលនៅក្នុងវិស័យអាជីវកម្មគ្រប់ ប្រភេទ ដែលវាជាមធ្យោបាយមួយដែលសម្របសម្រួលបញ្ហារបស់អ្នកធ្វើអាជីវកម្មនិងបំពេញនូវតម្រូវការរបស់ អតិថិជនបានយ៉ាងល្អប្រសើរ។

- មូលហេតុនៃការសិក្សា: ព្រោះការផ្សព្វផ្សាយរបស់ហាងនៅមានកម្រិត មិនទាន់មានភាពធំទូលាយដែលធ្វើអោយបាត់បង់ឧកាសក្នុងការកែស្វែងរកអតិថិជននិងប្រាក់ចំណូល។
- ចំណោទបញ្ហានៃការសិក្សា: គឺបង្ហាញពីផលវិបាក និងកង្វះខាតរបស់ប្រព័ន្ធដែលមានភាពយឺតយ៉ាវក្នុងការផ្តល់សេវាកម្មដល់អតិថិជន និងមិនទទួលបានសុវត្ថិភាពក្នុងការគ្រប់គ្រងអាជីវកម្មដែលធ្វើអោយមានភាពស្មុគស្មាញ និងពិបាកក្នុងការស្រាវជ្រាវសារឡើងវិញ។
- គោលបំណងនៃការសិក្សា: ចង់ផ្តោតសំខាន់លើចំណុចសំខាន់ៗនៃការប្រើប្រាស់ឌីជីថលទៅលើប្រព័ន្ធដែលវាបានដាក់អោយប្រើប្រាស់នៅលើកម្មវិធីទូរស័ព្ទដៃដែលវាងាយស្រួលប្រើជាមួយនឹងផ្ទាំងរចនារូបភាពមានភាពទាក់ទាញថែមទាំងជាភាពពេញនិយមសម្រាប់អ្នកប្រើប្រាស់ទូរស័ព្ទដៃផងដែរ។
- ទំហំ និងវិសាលភាពនៃប្រព័ន្ធ: ដែលបង្ហាញពីលទ្ធភាពរបស់ប្រព័ន្ធមកលើអាជីវកម្មនាសម័យបច្ចុប្បន្ននិងទទួលបានអត្ថប្រយោជន៍និងមានការរីកចម្រើនទៅមុខ។
- អត្ថប្រយោជន៍នៃការសិក្សាប្រព័ន្ធ: នឹងប្រាប់អ្នកពីភាពដែលអាចដោះស្រាយបញ្ហាដែលកើតមានក្នុងគ្រប់វិស័យទាំងអស់ដោយប្រើប្រាស់ប្រព័ន្ធឌីជីថល ជាពិសេសវិស័យអាជីវកម្មដែលនឹងបញ្ជាក់អោយយើងឃើញពីភាពល្អតូលាស់របស់ប្រព័ន្ធឌីជីថលនាពេលខាងមុខ វាជាកត្តានាំឲ្យប្រទេសជាតិមានការរីកចម្រើនថែមទៀត។

**២.២.២ ជំពូកទី១៖ រំលឹកទ្រឹស្តី ដែលពាក់ព័ន្ធ**

- ការប្រមូលឯកសារ: ដើម្បីអាចសិក្សាទៅលើប្រព័ន្ធបាន យើងត្រូវធ្វើការប្រមូលឯកសារដែលពាក់ព័ន្ធជាមុនសិន។
- ទ្រឹស្តីពាក់ព័ន្ធ: លើកយកទ្រឹស្តីដែលទាក់ទងនឹងភាសា programming និងកម្មវិធីដែលប្រើសម្រាប់ការសរសេរបង្កើតប្រព័ន្ធ ពីសៀវភៅនានា,បណ្ណាល័យ និងតាមប្រព័ន្ធអេឡិចត្រូនិចជាដើម។

**២.២.៣ ជំពូកទី២៖ វិធីសាស្ត្រ និងវេយសម្ព័ន្ធនៃការសិក្សាស្រាវជ្រាវ**

- វិធីសាស្ត្រសិក្សា: ការជ្រើសរើសយកវិធីសាស្ត្រមកបង្កើតប្រព័ន្ធហៅថា SDLC(System Development Life Cycle)ដើម្បីដំណើរការប្រព័ន្ធ។
- វេយសម្ព័ន្ធ: បង្ហាញពីដំណាក់កាលសិក្សារបស់ជំពូកនីមួយៗ។
- គម្រោងនិងពេលវេលានៃការសិក្សា: មានការកំណត់តារាងពេលវេលាទៅតាមគោលដៅ ដែលត្រូវធ្វើការសិក្សាលើប្រព័ន្ធដើម្បីសម្រួលដល់ដំណើរការសិក្សាឲ្យបានរលូន។

**២.២.៤. ជំពូកទី៣៖ ការវិភាគ ការគ្រោង និងការអនុវត្ត**

- ការវិភាគ: គឺជាដំណាក់កាលនៃការវិភាគទៅលើកង្វះខាតរបស់ប្រព័ន្ធបច្ចុប្បន្ន និងធ្វើការវិភាគកែលម្អទៅលើប្រព័ន្ធថ្មីជាមួយនឹងដំណើរការដែលប្រព័ន្ធបង្កើតឡើង។
- ការគ្រោង: ជាដំណាក់កាលដែលបង្ហាញពីការគ្រោងរបស់ប្រព័ន្ធបន្ទាប់ពីការវិភាគទៅលើប្រព័ន្ធរួចមក ។
- ការអនុវត្តន៍: ជាដំណាក់កាលដែលលើកយកការគ្រោងរបស់ប្រព័ន្ធមកធ្វើអោយចេញជារូបរាងពិតប្រាកដ ដោយប្រើប្រាស់កម្មវិធីសរសេរកូដ ក៏ដូចជាភាសាសរសេរកូដមួយចំនួនទៀតផងដែរ។

**២.២.៥. ជំពូកទី៤៖ សំយោគ សន្និដ្ឋាន និងការផ្តល់អនុសាសន៍**

- សំយោគសេចក្តី: ជាការនិយាយបកស្រាយពីសំយោគសេចក្តីរបស់ប្រព័ន្ធមុនពេលនិងក្រោយការបង្កើតប្រព័ន្ធថ្មី។
- សេចក្តីសន្និដ្ឋាន: ជាការនិយាយបកស្រាយពីអត្ថប្រយោជន៍និងប្រសិទ្ធិរបស់ការបង្កើតប្រព័ន្ធដែលបានបង្កើតថ្មី។

**២.៣ គំរោងពេលវេលានៃការសិក្សាស្រាវជ្រាវ**

គំរោងពេលវេលានៃការសិក្សាស្រាវជ្រាវនៃការបង្កើតប្រព័ន្ធគ្រប់គ្រងព័ត៌មានត្រូវបានកំណត់ដូចខាងក្រោម៖

តារាង 3: គំរោងពេលវេលានៃការសិក្សាស្រាវជ្រាវ

ដំណាក់កាលនៃការសិក្សា	២០២៣																							
	មិថុនា				មេសា				ឧសភា				មិថុនា				កក្កដា				សីហា			
	១	២	៣	៤	១	២	៣	៤	១	២	៣	៤	១	២	៣	៤	១	២	៣	៤	១	២	៣	៤
ធ្វើការកំណត់បញ្ហាលើប្រព័ន្ធការងារនិងគោលបំណង	■	■																						
ធ្វើការប្រមូលព័ត៌មាន			■	■																				
ធ្វើការវិភាគលើប្រព័ន្ធ					■	■	■	■																
ធ្វើការរៀបចំគំរោងប្រព័ន្ធ									■	■	■	■												
ការអនុវត្តន៍លើការសរសេរកូដ									■	■	■	■	■	■	■	■								
ធ្វើការសាកល្បងនិងកែតម្រូវ													■	■	■	■								
ការប្រើប្រាស់និងវាយតម្លៃទៅលើប្រព័ន្ធ																	■	■	■	■				

ក្នុងដំណាក់កាលនីមួយៗបានកំណត់ដូចខាងក្រោម៖

- ខែមិថុនា: ក្រុមចាប់ផ្តើមគត់សម្គាល់ទៅលើភាពរីកចម្រើនរបស់ប្រព័ន្ធបច្ចេកវិទ្យាទំនើបនិង បញ្ហាដែលតែងតែកើតមានឡើងលើដំណើរការអាជីវកម្មដែលមានភាពខ្វះខាតក្នុងការប្រើប្រាស់ប្រព័ន្ធបច្ចេកវិទ្យា ដែលបានជម្រុញអោយក្រុមធ្វើការសម្រេចជ្រើសរើសស្រាវជ្រាវលើប្រធានបទ App លក់ស្បែកជើង "Nike Sneaker App Store" ដើម្បីធ្វើការដោះស្រាយបញ្ហាដែលប្រព័ន្ធប្រធានបទ។
- ខែមេសា: ប្រមូលឯកសារ និងទ្រឹស្តីដែលពាក់ព័ន្ធដែលមាននៅសកលវិទ្យាល័យនានា និងសៀវភៅមួយចំនួនលើប្រព័ន្ធអ៊ីនធឺណិតដែលមានទាក់ទងនឹងការធ្វើអាជីវកម្មបែប Online ។
- ខែឧសភា: ធ្វើការវិភាគលើប្រព័ន្ធរៀបចំគ្រោងប្រព័ន្ធ បន្ទាប់មកចាប់ផ្តើមអនុវត្តន៍លើគម្រោង និងចងក្រងសៀវភៅជាដើម ។
- ខែមិថុនា ដល់ខែសីហា: ក្រុមសិក្សាចាប់ផ្តើមធ្វើការសាកល្បងនិងកែតម្រូវហើយនិងធ្វើការវាយតម្លៃទៅលើប្រព័ន្ធទាំងមូល។

### ជំពូកទី៣

## ការវិភាគ ការគ្រប់គ្រង និងការអនុវត្ត

### ៣.១. ការវិភាគ

មុននឹងឈានដល់ការបង្កើតប្រព័ន្ធច្រើនឡើង យើងត្រូវឆ្លងកាត់ការវិភាគអំពីប្រព័ន្ធបច្ចុប្បន្ន និងដំណើរការនៃ ផ្នែកមួយចំនួនដែលជាចំណុចដែលត្រូវវិភាគនិងមានទាក់ទងនឹងប្រព័ន្ធច្រើនជាមុនសិនផងដែរ។

#### ៣.១.១ ការងារជាក់ស្តែងទាក់ទងនឹងប្រព័ន្ធបច្ចុប្បន្ន

នៅក្នុងចំណុចនេះក្រុមសារណាពួកខ្ញុំ បានលើកយកផ្នែកមួយចំនួននៃការគ្រប់គ្រងផ្នែក ទិញលក់របស់ ហាងមកធ្វើការស្រាយបង្ហាញ មានដូចជា៖

- ការចេញវិក័យប័ត្រ
- ការបញ្ជាទិញទំនិញពីអតិថិជន
- ការផ្សព្វផ្សាយទំនិញទៅកាន់អតិថិជន
- ការគ្រប់គ្រងទិន្នន័យរបស់អតិថិជនមិនមានសុវត្ថិភាពល្អ។

#### ៣.១.២ ដំណើរការនៃការងាររបស់ប្រព័ន្ធបច្ចុប្បន្ន

- ការចេញវិក័យប័ត្រដោយប្រើដៃ
- ការបញ្ជាទិញទំនិញដោយការប្រើប្រាស់តាមការហៅទូរស័ព្ទ
- ការផ្សព្វផ្សាយទំនិញទៅកាន់អតិថិជន ដោយប្រើក្រដាសកាសែត មិនទូលំទូលាយ និងមានបញ្ហាក្នុង ការរក្សាទុកទិន្នន័យរបស់អតិថិជន មិនមានសុវត្ថិភាព និងពិបាកស្វែងរកមកប្រើប្រាស់វិញ។

#### ៣.១.៣ ការវាយតម្លៃលើប្រព័ន្ធបច្ចុប្បន្ន

ក្រោយពីបានស្វែងយល់ពីការវិភាគរបស់ប្រព័ន្ធការងារខាងលើបានធ្វើឲ្យក្រុមសារណាពួកខ្ញុំបានធ្វើការវាយ តម្លៃទៅលើហាងបានតាមរយៈចំណុចដូចខាងក្រោម៖

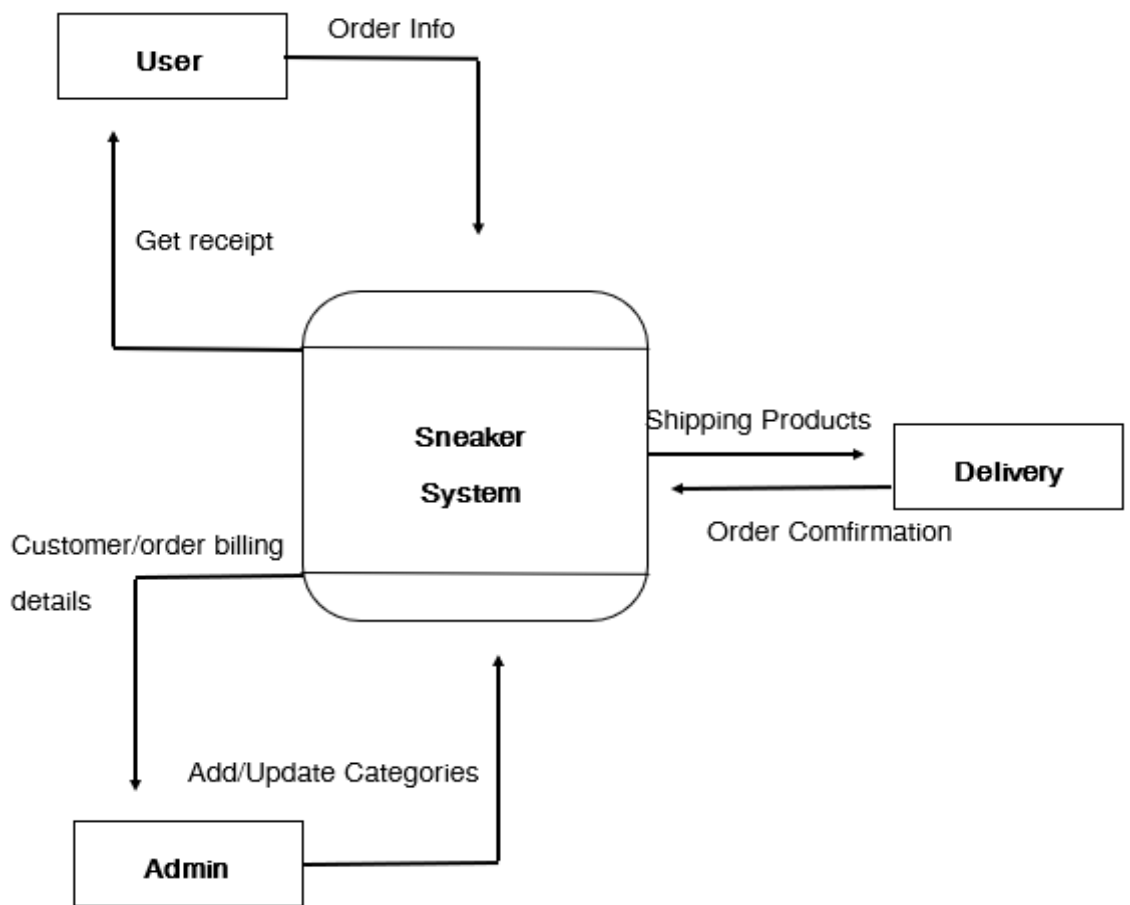
- ពុំទាន់មានកម្មវិធីលើប្រព័ន្ធទូរស័ព្ទដៃឲ្យអតិថិជនប្រើប្រាស់
- ការគ្រប់គ្រងទិន្នន័យដោយការកត់ត្រាដោយដៃ
- មិនទាន់មានវិធីសាស្ត្រផ្សព្វផ្សាយឲ្យបានទូលំទូលាយ
- មានភាពយឺតយ៉ាវក្នុងការបំពេញតម្រូវការរបស់ភ្ញៀវ
- ភ្ញៀវពិបាកស្វែងរកទំនិញដោយខ្លួនឯង។

**៣.១.៤ ការវិភាគលើប្រព័ន្ធសំណើរ**

ចំពោះការបង្កើតប្រព័ន្ធសំណើរថ្មីនេះគឺធ្វើការឆ្លើយតបទៅចំណុចខ្លះខាតដែលប្រព័ន្ធបច្ចុប្បន្នកំពុងជួបប្រទះដូចជា៖

- ការបង្កើតឲ្យមានការគ្រប់គ្រងទិន្នន័យនៅលើ database system
- ការបង្កើតកម្មវិធីដើម្បីធ្វើការគ្រប់គ្រងទៅលើការបញ្ជាទិញរបស់ហាង
- បង្កើត Feature search ដើម្បីងាយស្រួលឲ្យអតិថិជនស្វែងរកទំនិញបានឆាប់រហ័ស
- បង្កើត Feature notification ដើម្បីងាយស្រួលផ្សព្វផ្សាយឲ្យអតិថិជនដឹងពីព័ត៌មានទាក់ទងនឹងទំនិញថ្មីៗ និងពេលមាន promotion ជាដើម
- មានបែងចែកប្រភេទទំនិញ ងាយស្រួលស្វែងរកនិងធ្វើការបញ្ជាទិញ
- មានចេញវិក័យបំត្រភ្លាមៗ ពេលអតិថិជនធ្វើការបញ្ជាទិញរួចរាល់។

➢ Context diagram level





រូបភាព 11 Context diagram level

តាមរយៈរូបភាពទី11 Context diagram level នៅក្នុងប្រព័ន្ធសំណើរ Context diagram របស់ប្រព័ន្ធសំណើរ Nike Sneaker គឺមានទំនាក់ទំនងនឹង External Entity ចំនួន៣គឺ៖ Admin, User(Customer) និង Delivery។

**៣.១.៥ ដំណើរការលើប្រព័ន្ធសំណើរ**

- ប្រព័ន្ធគ្រប់គ្រងដោយ database គឺទិន្នន័យទាំងអស់របស់ User ត្រូវបានStoreទុកនៅក្នុង database (PostgreSQL engine)
- មានការកំណត់សិទ្ធិរបស់ User
- ធ្វើការបញ្ចូលទិន្នន័យរបស់ទំនិញចូលក្នុងប្រព័ន្ធ
- ការគ្រប់គ្រងដំណើរការទទួលកម្មវត្ថុរបស់អតិថិជន
- បញ្ចប់ដំណើរការកម្មវត្ថុរបស់អតិថិជនជាមួយវិក័យបត្របានត្រឹមត្រូវនិងរហ័ស។

**៣.២. ការគ្រោងប្រព័ន្ធ**

នៅក្នុងចំណុចនេះគឺពួកយើងនឹងលើកបង្ហាញពីការបង្កើតឡើងរបស់ប្រព័ន្ធតាមរយៈ Screen design, Process របស់ User ក្នុងការបញ្ជាទិញទំនិញរបស់ហាង Nike sneaker online store ក្រោយពេលត្រូវបានធ្វើការវិភាគរួចមក។

**៣.២.១. ការគ្រោងនៅលើ Process and User Interface**

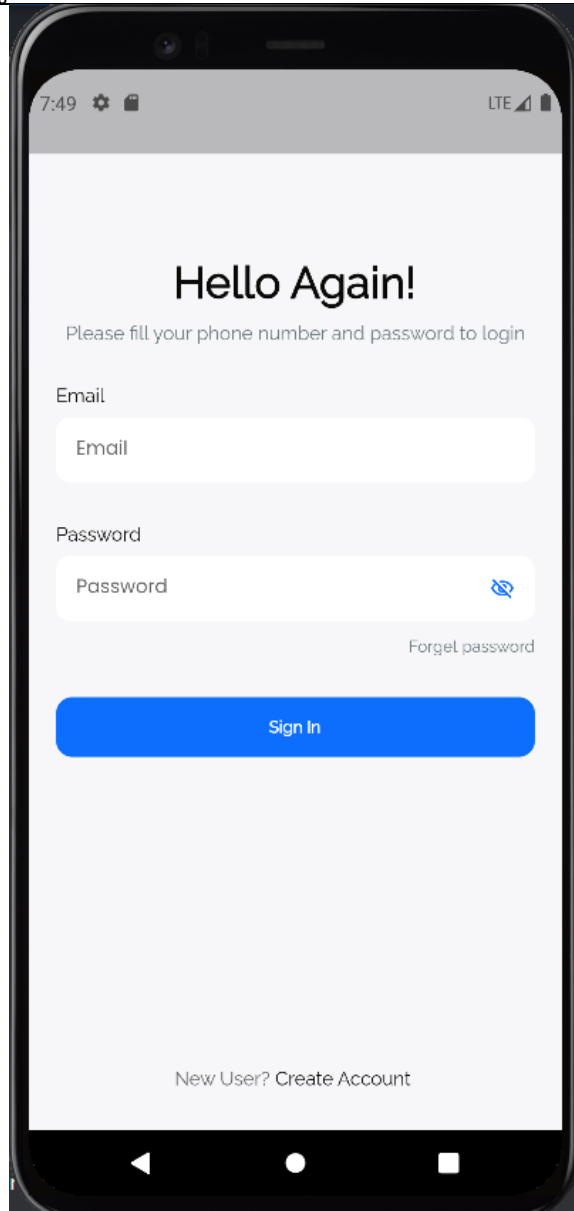
**៣.២.១.១. Screen Design របស់ User**

សម្រាប់ចំណុចនេះនឹងបង្ហាញពី Screen Design សំខាន់មួយចំនួនរបស់អ្នកប្រើប្រាស់ដែលប្រើនៅក្នុងប្រព័ន្ធសំណើរខាងលើ៖

**ក. Register & Login Screen**

- Login Screen

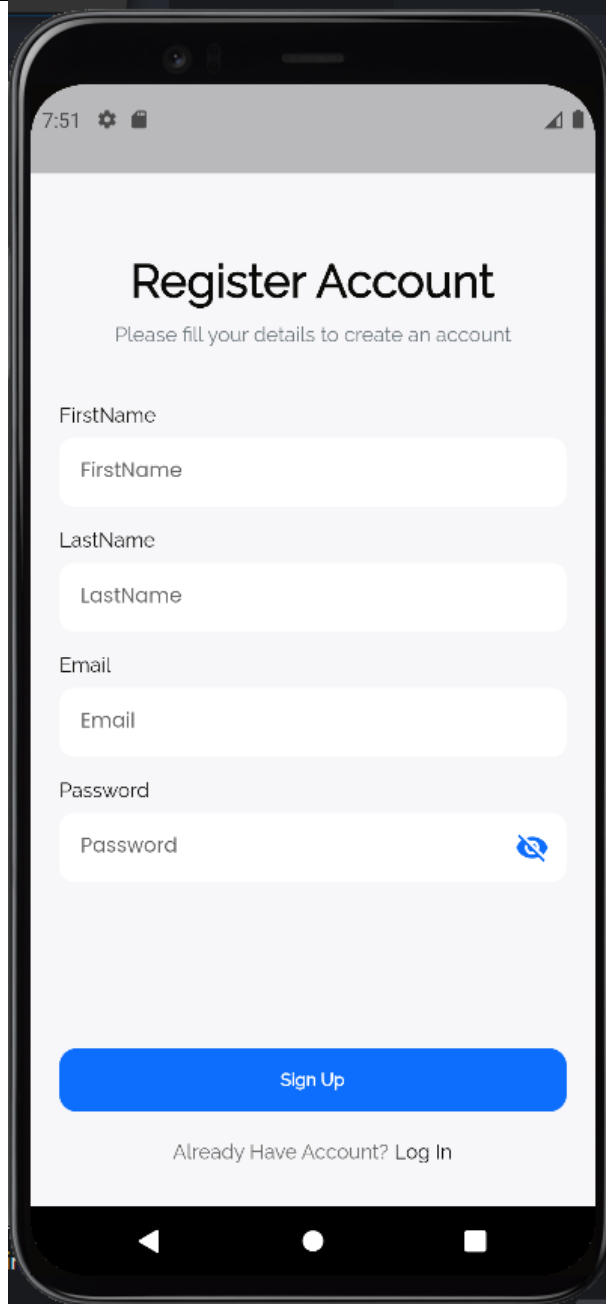
តាមការបង្ហាញនៃរូបភាពទី12 : Login Screen បានបញ្ជាក់ថា User ត្រូវមាន Email address account ជាមួយនឹង password សម្រាប់ធ្វើការ Login ជាមុនសិន។ប្រសិនបើ User ពុំទាន់មាន User Login ទេ User ត្រូវចូលទៅកាន់ Screen Register ដើម្បីធ្វើការ Register account ឬក៏ត្រូវធ្វើការ Create account ជាមុនសិនទើបមាន User អាច Login បាន។



រូបភាព 12 : Login Screen

- Register Screen

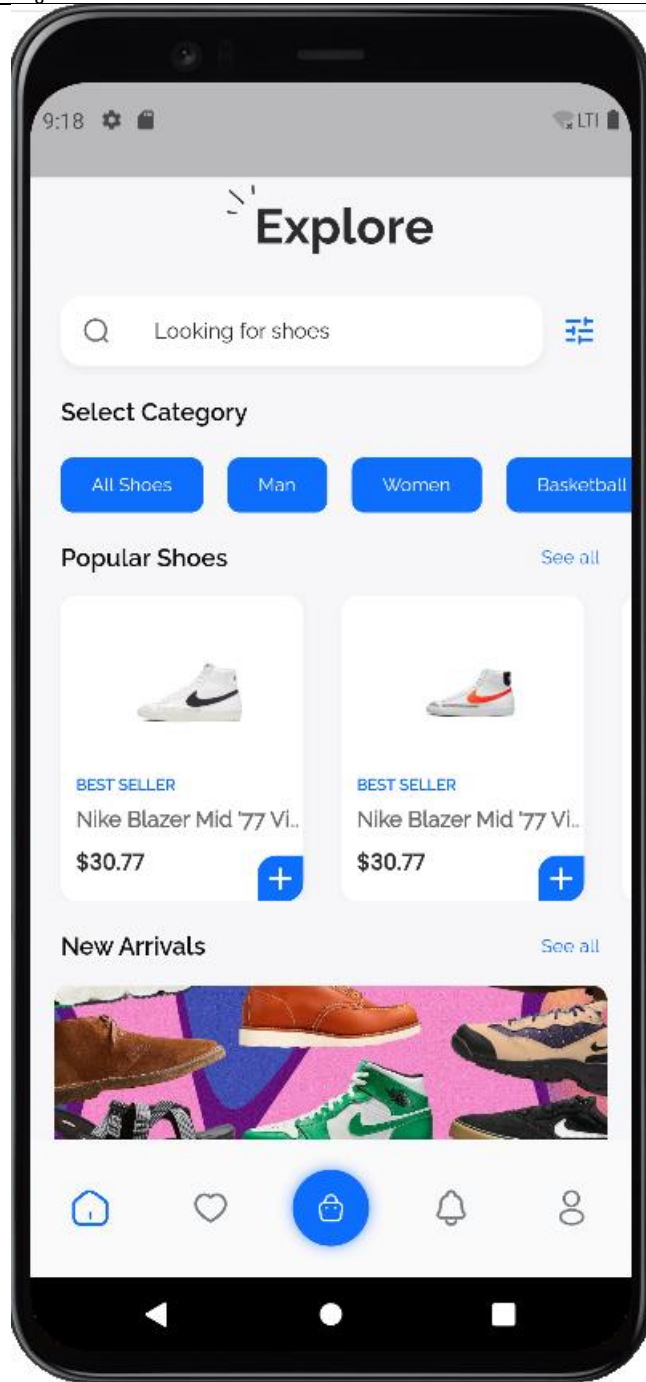
តាមការបង្ហាញនៃរូបភាពទី13 : Register Screen ជា Screen ដែល User ត្រូវចូលទៅ បង្កើត account user នៅពេលដែល User មិនទាន់មាន account សម្រាប់ login។



រូបភាព 13: Register Screen

ខ. Home Page

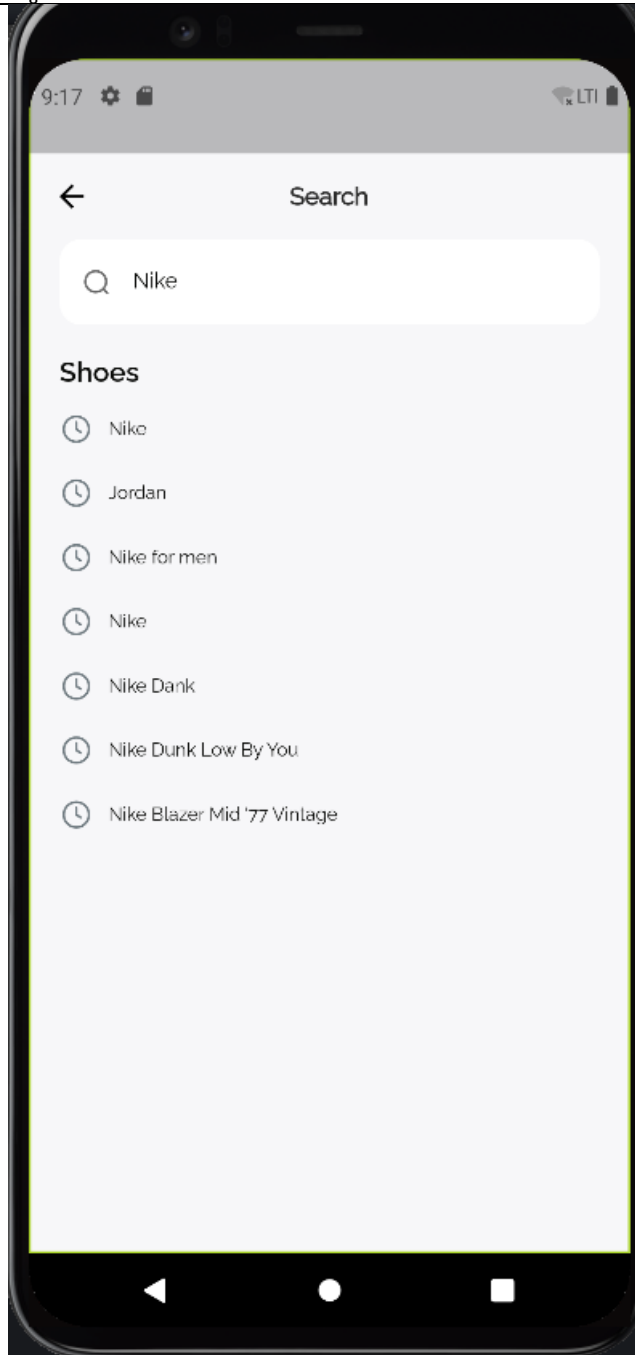
តាមការបង្ហាញនៃរូបភាពទី១៤ : Home Screen ជា Screen ដំបូងគេបន្ទាប់ពី Screen login និង Screen register ។ ក្នុង Screen នេះដែរអាចអោយយើងធ្វើការ research products , View Product details, ប្រភេទទំនិញ និងបង្ហាញ Products មួយចំនួននៅក្នុង Screen នោះផងដែរ។



រូបភាព 14: Home Screen

គ. Search Screen

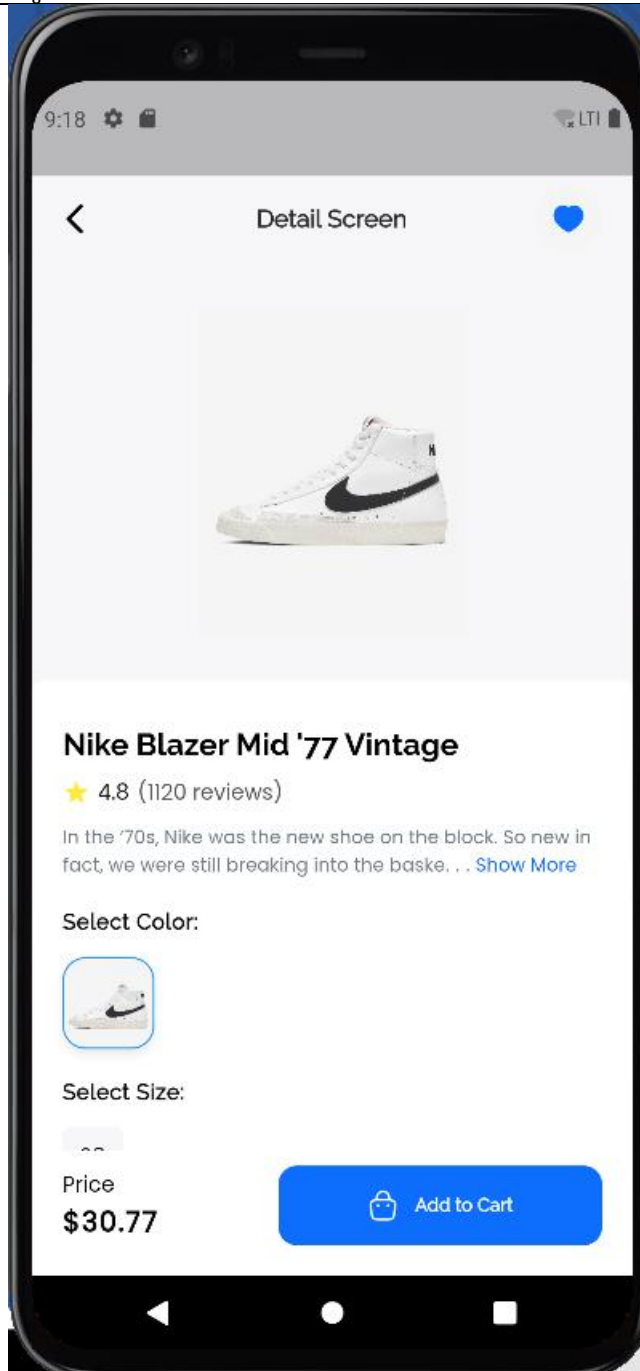
តាមការបង្ហាញនៃរូបភាពទី15 : Search Screen ជា Screen នេះនឹងបង្ហាញពីការស្វែងរកទំនិញ របស់អតិថិជនទៅតាមឈ្មោះរបស់ទំនិញ ហើយវានឹងទៅបង្ហាញនូវរាល់ទិន្នន័យដែលមាន។



រូបភាព 15 : Search Screen

២៥៥. Details Page

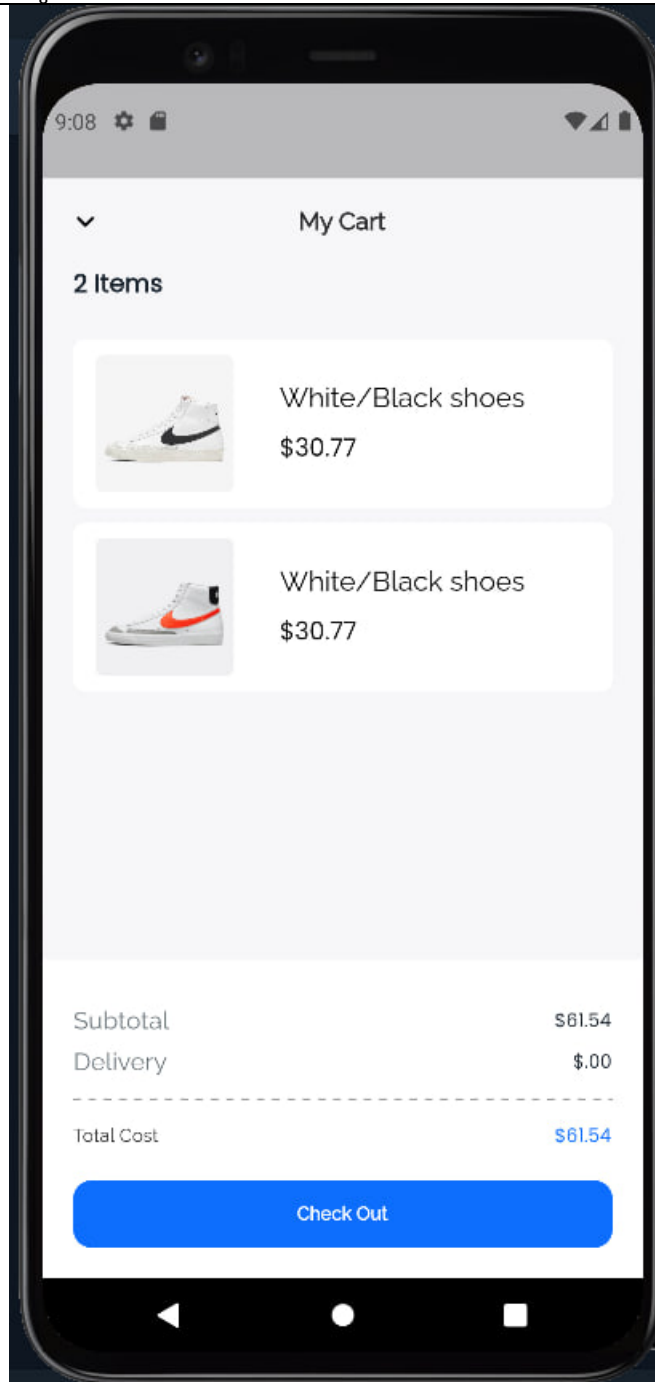
តាមការបង្ហាញនៃរូបភាពទី១៦: Details Screen ជា Screen ដែលអាចបង្ហាញពីព័ត៌មានលម្អិតរបស់ Product នីមួយៗ ។ ក្នុងនោះដែលមានបង្ហាញពី ឈ្មោះរបស់ទំនិញ, ព័ត៌មានបន្ថែមរបស់ទំនិញ, មានជា Size និង Color ជាដើម។



រូបភាព 16 : Details Screen

១២. Cart Page

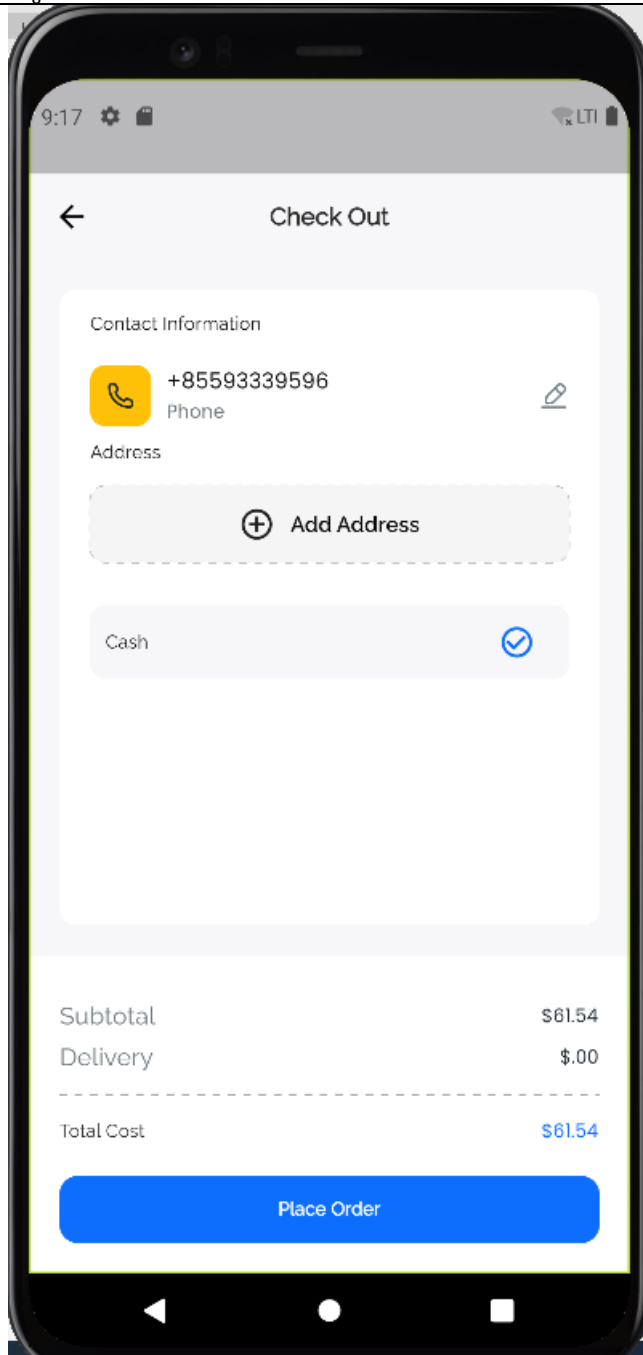
តាមការបង្ហាញនៃរូបភាពទី17 : Cart Screen ជា Screen ដែល user ធ្វើជ្រើសរើស product ណាមួយទុកក្នុង Screen cart នេះមុននឹងធ្វើការ Payment។



រូបភាព 17 : Cart Screen

**៥. Order Screen**

តាមការបង្ហាញនៃរូបភាពទី១៨ : Search Screen ជា screen ដែល route ចេញពី screen cart page នៅពេល user add cart ហើយចុច button check out ។ នៅក្នុង screen នេះតម្រូវអោយ user ត្រូវបំពេញព័ត៌មានផ្សេងទៀតដូចជា phone number និង address ជាដើម។

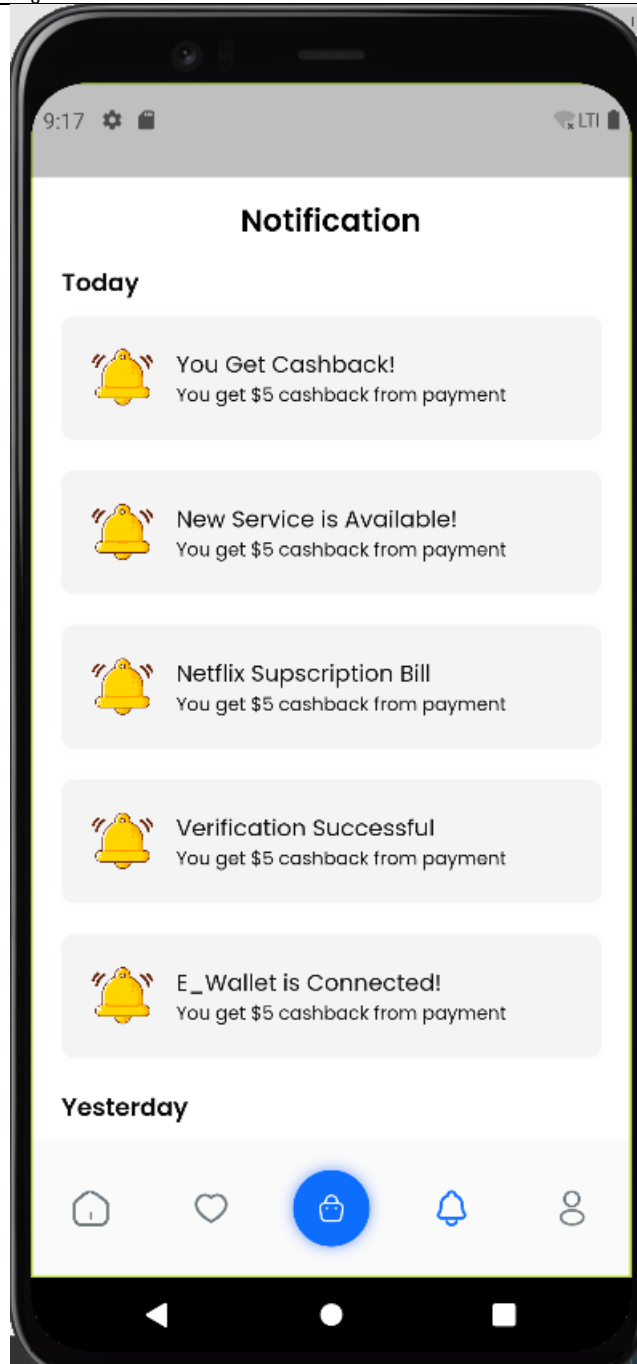


រូបភាព 18 : Order Screen

៥៥. Notification Screen

តាមការបង្ហាញនៃរូបភាពទី១៩ : Notification Screen ជា screen ដែលបង្ហាញនូវព័ត៌មានថ្មីរបស់ ទំនិញទៅកាន់ user ។ ដែលរាល់ព័ត៌មានដែល update ទាំងអស់នឹង alert នៅក្នុង screen នេះ។

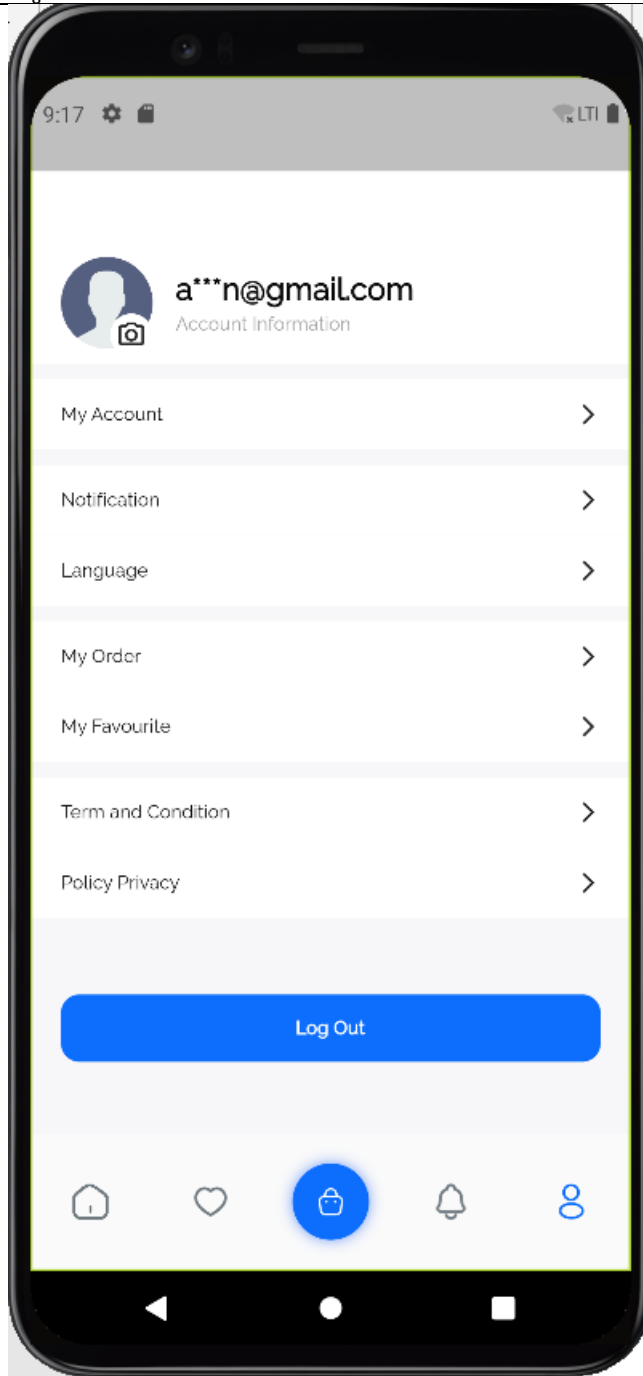




រូបភាព 19 : Notification Screen

៥. Profile Screen

តាមការបង្ហាញនៃរូបភាពទី២០ : Profile Screen ជា screen ដែលបង្ហាញពីព័ត៌មានផ្ទាល់ខ្លួនរបស់អ្នកប្រើប្រាស់ ដែលមានដូចជា ឈ្មោះអ្នកប្រើប្រាស់ លេខទូរស័ព្ទ និងរាល់ទិន្នន័យផ្សេងៗទៀតដែលអ្នកប្រើប្រាស់ធ្វើការបញ្ជាទិញទំនិញ។



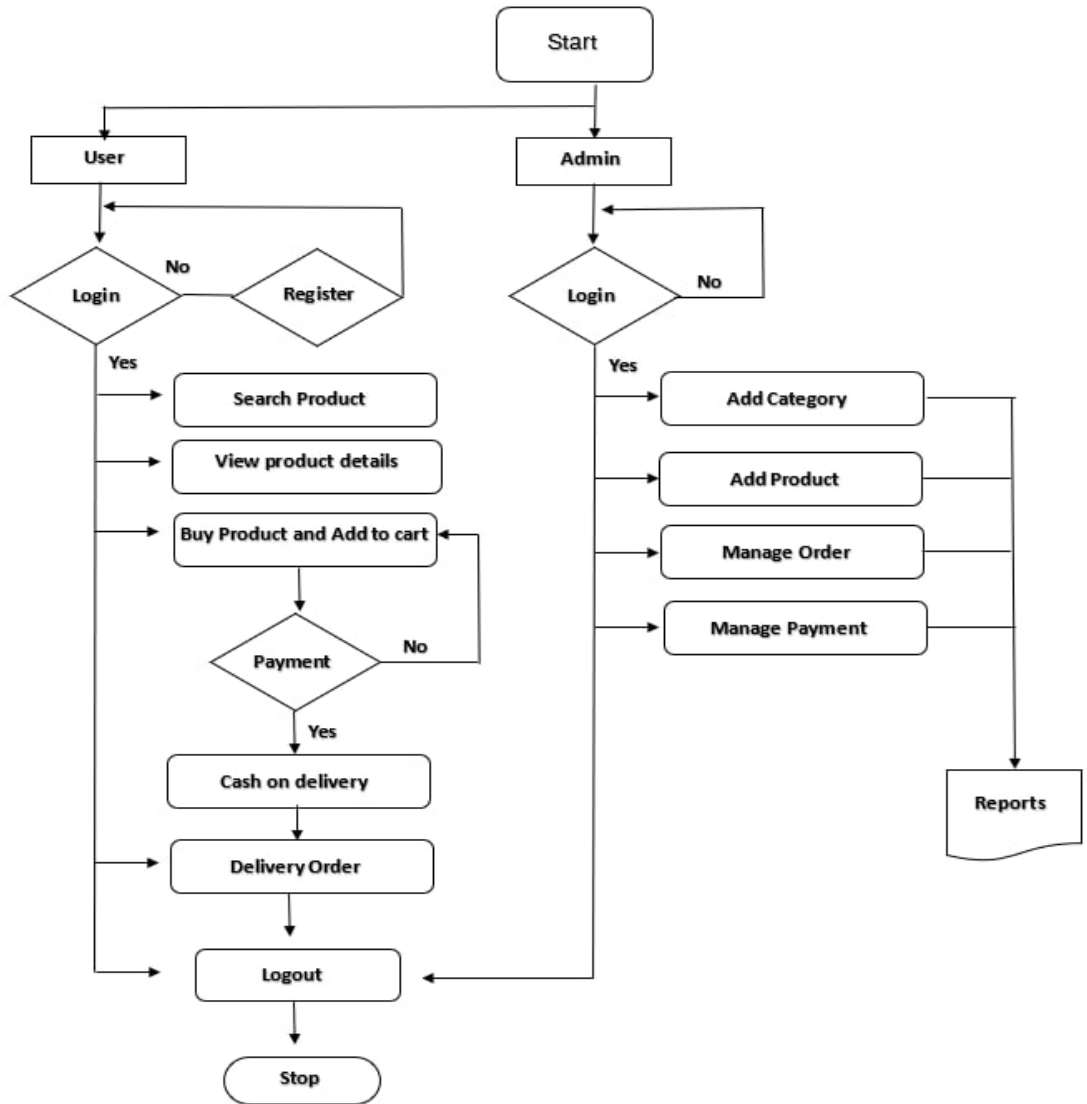
រូបភាព 20: Profile Screen

**៣.២.១.២. Process**

មុននឹងចូលទៅប្រើប្រាស់ប្រព័ន្ធនេះបានយើងត្រូវបែងចែកតួនាទីរបស់ User ជា២គឺ៖

- User: User ទូទៅត្រូវការ Username និង Password ដើម្បីអាចមានសិទ្ធិចូលទៅប្រើប្រាស់ប្រព័ន្ធបាន។

- Admin: Admin ត្រូវមាន Username និង Password ដែលបានកំណត់តួនាទីរួចដើម្បីអាចចូលទៅគ្រប់គ្រងប្រព័ន្ធសំណើរបាន។



រូបភាព 21 : Process របស់Appបញ្ជាទិញទំនិញ Nike Sneaker online store

### ៣.៣. ការអនុវត្តន៍

នៅក្នុងដំណាក់កាលនេះជាដំណាក់កាលដែលអ្នកអភិវឌ្ឍន៍កម្មវិធីបានធ្វើការអនុវត្តន៍ក៏ដូចជាបង្កើតប្រព័ន្ធឡើងដោយប្រើប្រាស់ភាសាសរសេរកូដ dart ឲ្យប្រើជាមួយ Framework Flutter សម្រាប់ផ្នែក frontend mobile និងផ្នែកBackend ប្រើប្រាស់ភាសាសរសេរកូដ Node.js ជាមួយនឹង PostgreSQL ជា database engine ។

- Installation: ជាការដែល developer ត្រូវទៅតម្លើងកម្មវិធីដែលពាក់ព័ន្ធមួយចំនួនដូចជា Vs code, Xcode, Android Studio ដែលសុទ្ធសឹងជាកម្មវិធីសម្រាប់សរសេរនិងតេស្ត code ផ្នែក Frontend ដើម្បីឲ្យប្រព័ន្ធមានដំណើរការល្អប្រសើរពេលដាក់អោយ User ប្រើប្រាស់។ សម្រាប់ផ្នែក backend គឺមានកម្មវិធីដែលពាក់ព័ន្ធមួយចំនួនដូចជា ExpressJS ជា Framework សម្រាប់សរសេរភាសា NodeJS ចំពោះ PgAdmin មានតួនាទីគ្រប់គ្រងលើ database ដែលវាជាអ្នកភ្ជាប់ការងារក៏ដូចជា ទំនាក់ទំនងពី NodeJS មកកាន់ database PostgreSQL ។
- Database Creation: ជាការទៅសរសេររបងើត query ទៅលើ tableនីមួយៗដើម្បីធ្វើការភ្ជាប់ relationship ជាមួយគ្នាទៅតាមគម្រោងដែលបានសិក្សា និងគ្រោងទុកនៅ PostgreSQL និងមាន ការ Configure User និង Password ដើម្បីភ្ជាប់ project ជាមួយនិង database។
- Project Building: ចំពោះ project នេះត្រូវបានសាងសង់ឡើងដោយផ្អែកទៅលើ flow មួយចំនួន ដែលបានគ្រោងទុកនិងជាការកំណត់របស់ developer ក្នុងការកំណត់សិទ្ធិទៅលើ Userដែលជាអ្នក ប្រើប្រាស់ ក្នុងនោះត្រូវបានបែងចែកជា២គឺ៖
  - public user: និយាយចំពោះអ្នកប្រើប្រាស់ទូទៅដែលមិនមែនជា admin ឬជា User មិនទាន់មាន account login ។ អ្នកទាំងនេះអាចចូលទៅប្រើប្រាស់បានជាមួយ Feature public មួយចំនួនដូចជា homepage និង details page ជាដើមដោយមិនមានការទាមទារអោយ មាន user login ឡើយ។
  - user login: សំដៅទៅដល់អ្នកប្រើប្រាស់ដែលបាន register account ដែលមាន username និង password ត្រឹមត្រូវដើម្បីមានសិទ្ធិក្នុងការ login ទៅប្រើប្រាស់ App ក៏ដូចជា Feature ទាំងអស់ដែល បានអនុញ្ញាតដោយ developer ទៅតាមតម្រូវការជាក់ស្តែង។
- Testing and Development: ជាដំណាក់កាលដែលអ្នកអភិវឌ្ឍន៍កម្មវិធី (developer) ត្រូវធ្វើការ តេស្តកូដរបស់ខ្លួនដែលបានសរសេររួចដើម្បីប្រាកដថា ប្រព័ន្ធនោះអាចដំណើរការបានរលូន និងបន្តធ្វើ ការងារទាំងនេះរហូតចប់ដំណើរការទាំងមូលរបស់ប្រព័ន្ធសំណើរប្រព័ន្ធកម្មវិធី។
- Deployment and User Manual: ជាដំណាក់កាលចុងក្រោយគេ ដែល developer បានសម្រេច ការបញ្ចប់ការសាងសង់នៃដំណើរការទាំងមូលរបស់ប្រព័ន្ធ និងត្រូវដាក់ប្រព័ន្ធឡើងទៅលើ store ដែលអាចអោយ user ធ្វើការទាញយកកម្មវិធីមកប្រើប្រាស់លើទូរស័ព្ទដែលបាន ។ បន្ទាប់មកនិងមាន ការរៀបចំសរសេរពីរបៀបប្រើប្រាស់ និងតម្រូវការថែទាំ ឬត្រូវធ្វើការ update features ថ្មីៗបន្ថែមទៀត ទៅលើប្រព័ន្ធនាពេលខាងមុខ។

**ជំពូកទី៤**

**សំយោគ សន្និដ្ឋាន និងការផ្តល់អនុសាសន៍**

**៥.១. សំយោគ**

តាមរយៈការសិក្សាទៅលើការសរសេរបង្កើតប្រព័ន្ធក្នុងការបញ្ជាទិញទំនិញរបស់ហាង “Nike Sneaker App Store” យើងសង្កេតឃើញថាប្រព័ន្ធបច្ចុប្បន្ននៅមិនទាន់អាចបំពេញតម្រូវការរបស់អតិថិជនបានគ្រប់គ្រាន់នៅឡើយទេ ហើយចំពោះការផ្សព្វផ្សាយទំនិញក៏នៅប្រើប្រាស់ក្រដាស ដើម្បីផ្សព្វផ្សាយដែលធ្វើឲ្យមានការចំណាយទាំងពេលវេលានិងកម្លាំងថែមទាំងមិនមានប្រសិទ្ធភាពគ្រប់គ្រាន់ក្នុងការផ្សព្វផ្សាយឡើយ។ ដោយឆ្លងកាត់ការប្រមូលទិន្នន័យនិងឯកសារដែលពាក់ព័ន្ធជាច្រើនរួចមកយើងក៏បានបង្កើតនូវប្រព័ន្ធថ្មីដែលអាចសម្រួលដល់ការទិញលក់នៅក្នុងហាងឲ្យទទួលបានប្រសិទ្ធភាពលក់និងបំពេញតម្រូវការរបស់អតិថិជនបានយ៉ាងល្អប្រសើរផងដែរ។

**៥.២. សេចក្តីសន្និដ្ឋាន**

ក្រោយពីបានសិក្សានិងស្វែងយល់ពីការប្រើប្រាស់របស់ប្រព័ន្ធសំណើដែលមានឈ្មោះថា “Nike Sneaker App Store” រួចមកយើងសង្កេតឃើញប្រព័ន្ធសំណើនេះអាចជួយស្វែងរកទំនិញអោយអតិថិជនបានរហ័សទាន់ចិត្តតាមរយៈការប្រើប្រាស់ Feature Search និងមាន Feature Notification ដែលអាចអោយអតិថិជនទទួលបានដំណឹងថ្មីៗពីខាងហាងបានគ្រប់ពេលវេលានិងគ្រប់ទីកន្លែង។ ទន្ទឹមនឹងសមត្ថភាពដោះស្រាយបញ្ហារបស់ប្រព័ន្ធសំណើខាងលើក៏មានលក្ខខណ្ឌមួយចំនួនដែលប្រព័ន្ធសំណើនេះត្រូវសិក្សា និងធ្វើការអភិវឌ្ឍន៍បន្ថែមនោះគឺប្រព័ន្ធសំណើមិនទាន់អាចធ្វើការ Payment ជាមួយនឹងធនាគារបានទេ។

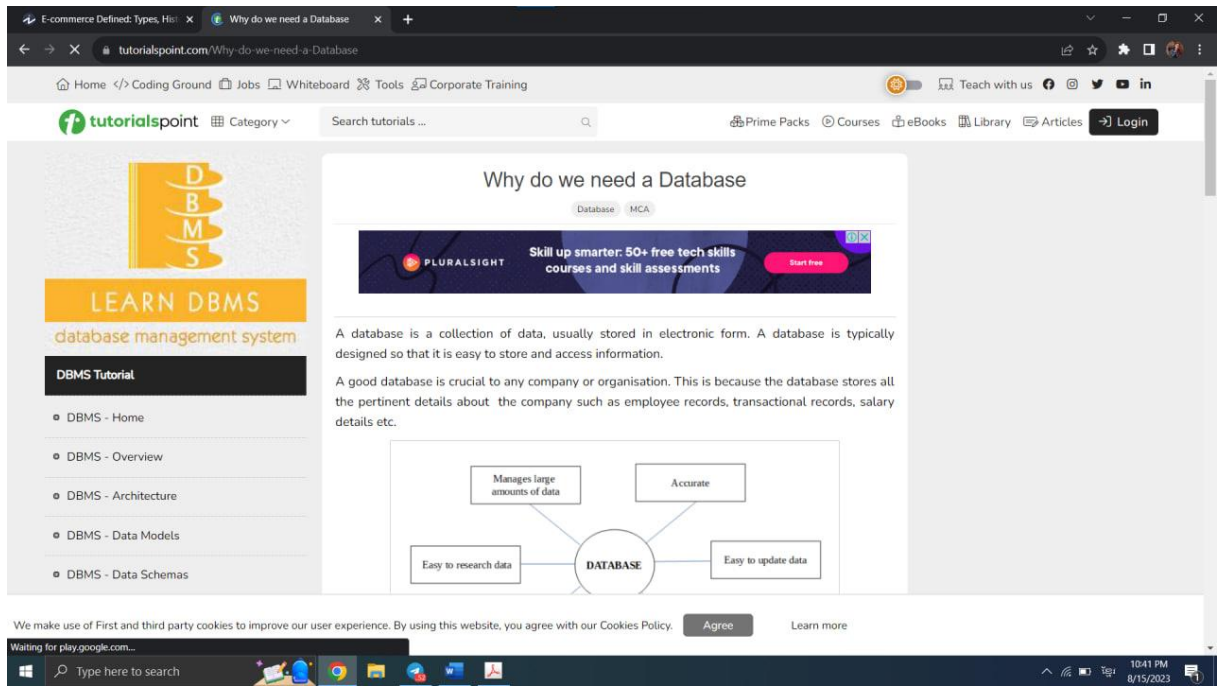
# ឯកសារយោង

## Books

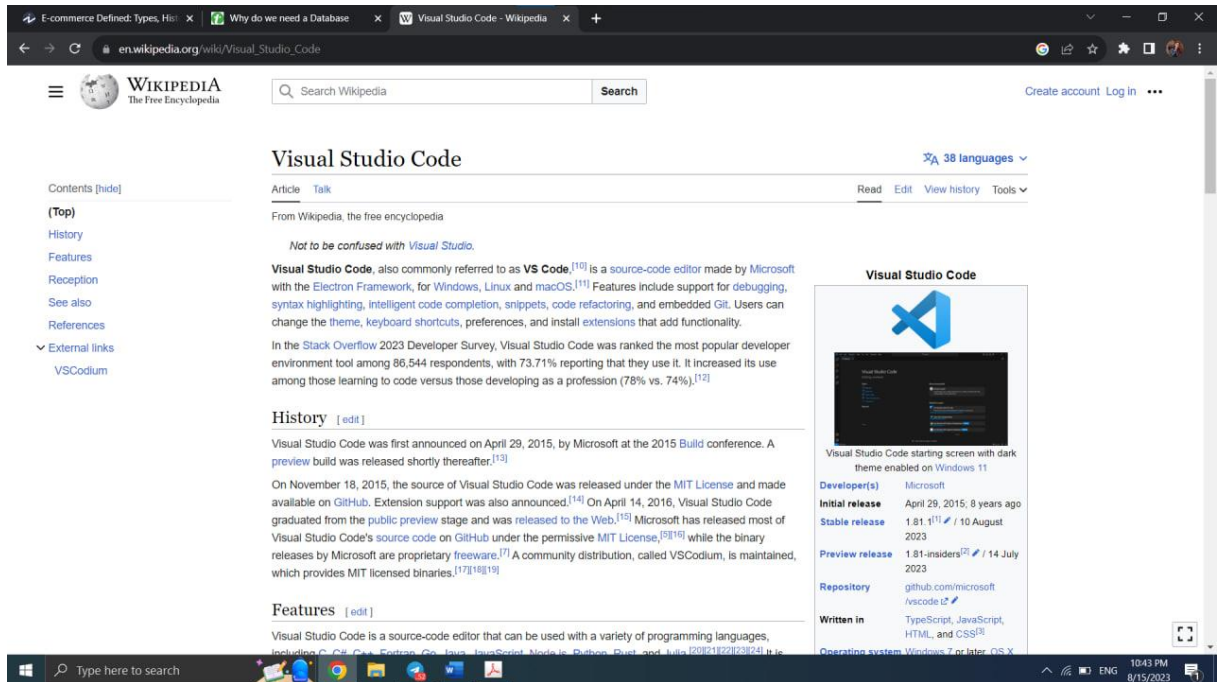
- ១ E-Commerce ទ្រឹស្តីរបស់លោក Carter McNamara ដកស្រង់ចេញពីសៀវភៅ Electronic Commerce បោះពុម្ពផ្សាយនៅឆ្នាំ២០០១។
- ២ PostgreSQL ទ្រឹស្តីរបស់លោក Neil Matthew and Richard Stones ដកស្រង់ចេញពីសៀវភៅ Beginning Databases with PostgreSQL 2nd edition បានបោះពុម្ពផ្សាយនៅឆ្នាំ២០០៥។
- ៣ Flutter ដកស្រង់ចេញពីសៀវភៅអេឡិចត្រូនិក(E-Book) របស់ TutorailsPoint Copyright ឆ្នាំ២០១៩។
- ៤ Dart programming ដកស្រង់ពីសៀវភៅ pdf ឈ្មោះ Dart language with null safety របស់លោកគ្រូ Oum Saokosal ឆ្នាំ២០១០ នៅទំព័រទី២។

## Website

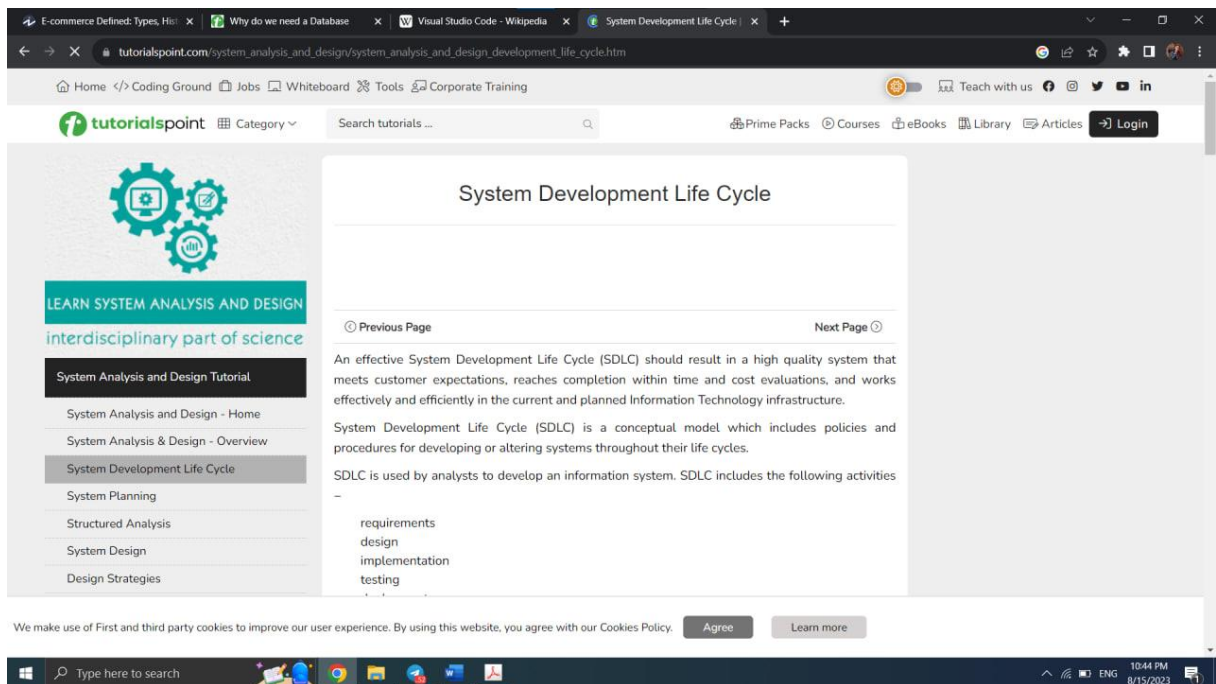
- 1. Database: <https://www.tutorialspoint.com/Why-do-we-need-a-Database> (8/15/2023)



- 2. Visual\_Studio\_Code: [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code) (8/15/2023)



3. Datababe\_Management\_System:[https://www.tutorialspoint.com/system\\_analysis\\_and\\_design/system\\_analysis\\_and\\_design\\_development\\_life\\_cycle.htm](https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_development_life_cycle.htm)(8/15/2023)



4. E-Commerce:<https://www.investopedia.com/terms/e/ecommerce.asp#toc-types-of-e-commerce>(8/15/2023)

E-commerce Defined: Types, His... x +

investopedia.com/terms/e/ecommerce.asp#toc-how-do-you-start-an-e-commerce-business

Investopedia INVESTING SIMULATOR BANKING PERSONAL FINANCE NEWS REVIEWS ACADEMY TRADE

Table of Contents

- What Is E-commerce?
- Understanding E-commerce
- Special Considerations
- History
- Pros and Cons
- Types
- Revenue Models
- Example
- FAQs
- The Bottom Line

Types of E-Commerce

Business to business Business to consumer Business to Government

Consumer to consumer Consumer to business Consumer to Government

investopedia / Sabrina Jlang

**Important:** The U.S. Census Bureau conducts estimates of retail e-commerce sales in the United States. In the first quarter of 2023, retail e-commerce accounted for 15.1% of total sales in the country, totaling roughly \$272.6 billion. These figures are adjusted for seasonal variation.<sup>[5]</sup>

Types of E-commerce Revenue Models

In addition to crafting what type of e-commerce company a business wants to be, the business must decide how it wants to make money. Due to the unique nature of e-commerce, the

heather mahalik e commerce book 42 days ago

Heather Mahalik E commerce book... google.com/search?q=heather-mahali...

Amazon.com amazon.com/Books-Heather-Mahalik/57...

Amazon.com - E commerce by heath... amazon.com/?crid=26HGKZ7J6G406i...

Type here to search 10:40 PM 8/15/2023

5. PostgreSQL:<https://dade2.net/what-is-postgresql-how-does-postgresql-work/>(8/15/2023)

E-commerce Defined: Type... x Why do we need a Datab... x Visual Studio Code - Wiki... x System Development Life... x DEV Node.js Express Login exe... x What is PostgreSQL? How... x +

dade2.net/what-is-postgresql-how-does-postgresql-work/

Software / June 17, 2020 / 0

## What is PostgreSQL? How Does PostgreSQL Work?

Welcome to Dade2!

Type here to search 10:47 PM 8/15/2023



6. Node.Js:<https://dev.to/tienbku/node-js-express-login-example-with-postgresql-5676>(8/15/2023)

The screenshot shows a web browser displaying an article on Dev.to. The article title is "Node.js Express Architecture with Authentication & Authorization". Below the title, there is a sub-header "More details at: <https://bezkode.com/jwt-refresh-token-node-js/>". The main text reads: "You can have an overview of our Node.js Express Login & Registration App with PostgreSQL in the diagram below:". The diagram illustrates the architecture of a Node.js Express application. It shows an "HTTP Request" entering from the top left, passing through a "Matched Route" in "Express.js". The request then goes through a "Security" layer containing "CORS Middleware", "JWT Authentication Middleware", and "Authorization Middleware". The request then reaches the "Controllers" layer, which interacts with "Sequelize" and a "DB Driver". The "DB Driver" is connected to a database. Finally, an "HTTP Response" is sent out from the bottom right. The diagram is enclosed in a dashed green box labeled "Node.js".

DEV Community

**Trending in Webdev**

The Webdev community is currently fascinated by the visualization of web frameworks as superheroes, with a spotlight on [React.js](#). Simultaneously, there's a growing interest in learning [Vue.js](#) due to its unique features and ease of use.

**We asked AI to imagine web frameworks as superheroes**  
Matja Sosc · Aug 14  
#webdev #javascript #react #gpt3

**Learn Vue: Introduction to Vue.js**  
KingstonCodes · Aug 14  
#javascript #vue #webdev #frontend

10:45 PM 8/15/2023

# ଉପସଂହୃତ

## A. Source Code

### 1 Configure Route

```
You, 32 seconds ago | 0 authors (Ankesh and others)
1  import 'dart:async';
2
3  import 'package:allpay/src/cores/wolk_though/page/onboarding_screen.dart';
4  import 'package:allpay/src/module/auth/local_storage/local_storage.dart';
5  import 'package:allpay/src/module/auth/sign_in/screen/forget_password.dart';
6  import 'package:allpay/src/module/auth/sign_in/screen/register_screen.dart';
7  import 'package:allpay/src/module/home/pages/detail_page.dart';
8  import 'package:allpay/src/module/home/pages/home_page.dart';
9  import 'package:allpay/src/module/home/pages/popular_page.dart';
10 import 'package:allpay/src/module/home/pages/search_result_screen.dart';
11 import 'package:allpay/src/module/my_card/screen/add_new_delivery_address.dart';
12 import 'package:allpay/src/module/my_card/screen/choose_location_address.dart';
13 import 'package:allpay/src/module/my_card/screen/my_card_page.dart';
14 // import 'package:allpay/src/module/my_card/screen/order_list_screen.dart';
15 import 'package:allpay/src/module/notification/screen/notification.dart';
16 import 'package:allpay/src/module/profile/page/edit_profile.dart';
17 import 'package:allpay/src/module/profile/page/profile_page.dart';
18 import 'package:allpay/src/module/profile/page/view_order_detail.dart';
19 import 'package:allpay/src/module/profile/page/view_order_page.dart';
20 import 'package:flutter/material.dart';
21 import 'package:get/get.dart';
22 import 'package:go_router/go_router.dart';
23 import '../cores/wolk_though/page/splash_screen.dart';
24 import '../module/auth/sign_in/screen/logins_screens.dart';
25 import '../module/auth/sign_in/screen/verify_otp_screen.dart';
26 import '../module/bottomnavigationbar/bottomnavigationbar.dart';
27 import '../module/favourite/favourite_page.dart';
28 import '../module/home/pages/search_screen.dart';
29 import '../module/home/pages/products_by_category.dart';
30
```

```
31 final GlobalKey<NavigatorState> _rootNavigatorKey = Get.key;
32 final GlobalKey<NavigatorState> _shellNavigatorKey =
33     GlobalKey<NavigatorState>();
34
35 const _initialLocation = '/sso';
36
37 final GoRouter router = GoRouter(
38     navigatorKey: _rootNavigatorKey,
39     debugLogDiagnostics: true,
40     initialLocation: _initialLocation,
41     // redirect: _redirect,
42     routes: [
43         ShellRoute(
44             navigatorKey: _shellNavigatorKey,
45             builder: (_, state, child) {
46                 return ButtonNavigationBar(
47                     child: child,
48                 ); // ButtonNavigationBar
49             },
50             routes: _shellRoutes,
51         ), // ShellRoute
52
53 > // GoRoute(...)
54 GoRoute(
55     path: '/sso',
56     builder: (_, state) => const SplashScreen(),
57     routes: const [],
58 ), // GoRoute
59 GoRoute(
60     path: '/boarding',
61     builder: (_, state) => const OnBoardingScreen(),
62     routes: const [],
```

```
67     ), // GoRoute
68     GoRoute(
69         path: '/login',
70         builder: (_, state) => const LoginScreens(),
71         routes: const [],
72     ), // GoRoute
73     GoRoute(
74         path: '/register',
75         builder: (_, state) => RegisterScreen(),
76         routes: const [],
77     ), // GoRoute
78     GoRoute(
79         path: '/fogetpass',
80         builder: (_, state) => const ForgetPassword(),
81         routes: const [],
82     ), // GoRoute
83     GoRoute(
84         path: '/otp',
85         builder: (_, state) => const VerifyOTPScreen(),
86         routes: const [],
87     ), // GoRoute
88     GoRoute(
89         path: '/view-order',
90         parentNavigatorKey: _rootNavigatorKey,
91         builder: (_, state) {
92             return const ViewOrderPage();
93         },
94         routes: <GoRoute>[
95             GoRoute(
96                 path: 'view-order-detail/:id',
97                 parentNavigatorKey: _rootNavigatorKey,
98                 builder: (_, state) {
```

```

99         int id = int.parse(state.pathParameters['id']!);
100         return ViewOrderDetailPage(id: id);
101     },
102     ), // GoRoute
103     ], // <GoRoute>[]
104     ), // GoRoute
105     ],
106 ); // GoRouter
107
108 final _shellRoutes = <GoRoute>[
109   GoRoute(
110     path: '/',
111     pageBuilder: (_, state) => const NoTransitionPage(child: HomePage()),
112     routes: <GoRoute>[
113       // GoRoute(...)
114       GoRoute(
115         parentNavigatorKey: _rootNavigatorKey,
116         path: 'cart',
117         pageBuilder: (_, state) => CustomTransitionPage(
118           child: const MyCardPage(),
119           transitionsBuilder: (_, animation, secondaryAnimation, child) =>
120             SlideTransition(
121               position: Tween<Offset>(
122                 begin: const Offset(0, 1),
123                 end: const Offset(0, 0),
124               ).animate( // Tween
125                 CurvedAnimation(parent: animation, curve: Curves.easeInOut),
126               ),
127               child: child,
128             ), // SlideTransition
129           ), // CustomTransitionPage
130         ), // GoRoute
131       ],
132     ), // CustomTransitionPage
133   ), // GoRoute

```

```

138
139 GoRoute(
140     parentNavigatorKey: _rootNavigatorKey,
141     path: 'detail/:id',
142     builder: (_, state) => DetailPage(
143         id: int.tryParse(state.pathParameters['id'] ?? '')!,
144     ), // DetailPage
145     routes: const <GoRoute>[]), // GoRoute
146 GoRoute(
147     parentNavigatorKey: _rootNavigatorKey,
148     path: 'popular-router',
149     builder: (_, state) => PopularPage(),
150     routes: const [],
151 ), // GoRoute
152 > // GoRoute(...)
158 GoRoute(
159     parentNavigatorKey: _rootNavigatorKey,
160     path: 'go-search',
161     builder: (_, state) => const SearchScreen(),
162     routes: [
163         GoRoute(
164             parentNavigatorKey: _rootNavigatorKey,
165             path: 'search-result',
166             builder: (_, state) => SearchResultScreen(
167                 textSearch: state.queryParameters['textSearch'],
168                 // name: state.queryParameters['name'] ?? '',
169             ), // SearchResultScreen
170         ), // GoRoute
171     ],
172 ), // GoRoute
173 GoRoute(
174     parentNavigatorKey: _rootNavigatorKey,

```

```

175     path: 'category',
176     builder: (_, state) => ProductsByCategoryPage(
177         id: int.tryParse(state.queryParameters['id'] ?? ''),
178         name: state.queryParameters['name'],
179         // name: state.queryParameters['name'] ?? '',
180     ), // ProductsByCategoryPage
181 ), // GoRoute
182 GoRoute(
183     parentNavigatorKey: _rootNavigatorKey,
184     path: 'mycard-router',
185     builder: (_, state) => const MyCardPage(),
186     routes: const [],
187 ), // GoRoute
188 ], // <GoRoute>[]
189 ), // GoRoute
190 GoRoute(
191     name: 'AllFav',
192     path: '/fav',
193     builder: (_, state) => const FavouritePage(),
194 ), // GoRoute
195
196 GoRoute(
197     path: '/favorite-router',
198     pageBuilder: (_, state) => const NoTransitionPage(
199         child: FavouritePage(),
200     ), // NoTransitionPage
201     routes: [
202         GoRoute(
203             parentNavigatorKey: _rootNavigatorKey,
204             path: 'detail/:id',
205             builder: (_, state) => DetailPage(
206                 id: int.tryParse(state.pathParameters['id'] ?? '')!,

```

```

207     ), // DetailPage
208     ), // GoRoute
209   ],
210 ), // GoRoute
211 //GoRoute(path: '/boarding',builder: (_, state)=>const OnBoardingScreen()),
212
213 GoRoute(
214   path: '/notification',
215   pageBuilder: (_, state) => const NoTransitionPage(
216     child: NotificationScreen(),
217   ), // NoTransitionPage
218 ), // GoRoute
219 GoRoute(
220   path: '/profile/-router',
221   pageBuilder: (_, state) => const NoTransitionPage(
222     child: ProfilePage(),
223   ), // NoTransitionPage
224   routes: <GoRoute>[
225     GoRoute(
226       parentNavigatorKey: _rootNavigatorKey,
227       path: 'edit-profile',
228       builder: (_, state) => const EditProfilePage(),
229     ) // GoRoute
230   ], // <GoRoute>[]
231 ), // GoRoute
232
233 // Add new address
234
235 GoRoute(
236   path: '/add-new-delivery-address',
237   builder: (context, state) {
238     return const AddNewDeliveryAddress();

```



```

239     },
240     routes: [
241       GoRoute(
242         path: 'choose-location-address',
243         builder: (context, state) {
244           return const ChooseLocationAddress();
245         },
246       ) // GoRoute
247     ],
248   ), // GoRoute
249 ]; // <GoRoute>[]
250
251 const noAuthRoute = [
252   '/sso',
253   '/boarding',
254   '/login',
255   '/register',
256   '/fogetpass',
257   '/otp'
258 ];
259
260 FutureOr<String?> _redirect(_, state) async { The declaration '_redirect' isn't referenced. ⚠Try
261   final location = state.location;
262   final token = await LocalStorage.getStringValue(key: 'access_token');
263   debugPrint('TK : $token');
264   debugPrint('Location => $location');
265   if (!noAuthRoute.any((route) => location.contains(route)) && token.isEmpty) {
266     debugPrint('Error: Route Not allow');
267     return '/login';
268   }
269   return null;
270 }

```

## 2 Login Page

lib > src > module > auth > sign\_in > screen > logins\_screens.dart > ...

You, 30 seconds ago | 6 authors (Chhunvirak and others)

```
1 import 'package:flutter/material.dart';
2 // import 'package:flutter_dotenv/flutter_dotenv.dart';
3 import 'package:get/get.dart';
4 import 'package:go_router/go_router.dart';
```

```
5
6 import '../../../../../config/routers/router.dart';
7 import '../../../../../widget/custom_text.dart';      Angkeasel, 5 months ago • upload
8 import '../../../../../widget/home/custom_buttons.dart';
9
```

```
10 import '../controller/contoller.dart';
```

```
11
12 // import 'package:http/http.dart' as http;
13
```

Chhunvirak, 3 months ago | 4 authors (Chhunvirak and others)

```
14 class LoginScreens extends StatelessWidget {
15   const LoginScreens({super.key});
16
17   @override
18   Widget build(BuildContext context) {
19     final logController = Get.put(ControllerSignin());
20     // var phone = LocalStorage().getStringData(key: "ph");
21     // var password = LocalStorage().getStringData(key: "pass");
22     return Scaffold(
23       appBar: AppBar(),
24       //resizeToAvoidBottomInset: false,
25       body: Obx(
26         () => SafeArea(
27           minimum: const EdgeInsets.only(
28             left: 20,
29             right: 20,
30             top: 20,
```

```

31     ), // EdgeInsets.only
32     child: Column(
33       crossAxisAlignment: CrossAxisAlignment.start,
34       mainAxisAlignment: MainAxisAlignment.start,
35       children: [
36         Expanded(
37           child: SingleChildScrollView(
38             child: Column(
39               mainAxisAlignment: MainAxisAlignment.start,
40               children: [
41                 Text(
42                   "Hello Again!",
43                   style: Theme.of(context).textTheme.titleSmall!.copyWith(
44                     fontWeight: FontWeight.w700, fontSize: 32),
45                 ), // Text
46                 const SizedBox(
47                   height: 10,
48                 ), // SizedBox
49                 Center(
50                   child: Text(
51                     "Please fill your phone number and password to login",
52                     textAlign: TextAlign.center,
53                     style: Theme.of(context)
54                       .textTheme
55                       .titleMedium!
56                       .copyWith(
57                         fontWeight: FontWeight.w400,
58                         color: const Color(0xff707B81)),
59                   ), // Text
60                 ), // Center
61                 const SizedBox(height: 30),
62                 CustomTextFiled(

```

```

lib > src > module > auth > sign_in > screen > logins_screens.dart > ...
62     CustomTextFiled(
63       title: 'Email',
64       hintText: 'Email',
65       height: 60,
66       controller: logController.emailTxtController,
67       isValidate: false,
68       textInputAction: TextInputAction.next,
69       onChanges: (email) {},
70     ), // CustomTextFiled
71     const SizedBox(height: 30),
72     CustomTextFiled(
73       title: 'Password',
74       hintText: 'Password',
75       height: 60,
76       controller: logController.passwordTxtController,
77       maxlenght: 15,
78       isObscureText: !logController.isGreyEye.value,
79       suffixIcon: IconButton(
80         onPressed: () {
81           logController.isGreyEye.value =
82             !logController.isGreyEye.value;
83         },
84         icon: Icon(
85           logController.isGreyEye.value
86             ? Icons.visibility_outlined
87             : Icons.visibility_off_outlined,
88           size: 18,
89         ), // Icon
90       ), // IconButton
91
92     //initialValues: controllerPassword.text,
93

```

```

lib > src > module > auth > sign_in > screen > logins_screens.dart > ...
94         isValidate: false,
95       ), // CustomTextFiled
96       const SizedBox(height: 12),
97       Row(
98         mainAxisAlignment: MainAxisAlignment.end,
99         children: [
100           GestureDetector(
101             onTap: () {
102               debugPrint('=====>Forget password');
103               context.push("/fogetpass");
104             },
105             child: Text(
106               "Forget password",
107               style: Theme.of(context)
108                 .textTheme
109                 .titleMedium!
110                 .copyWith(
111                   fontWeight: FontWeight.w400,
112                   fontSize: 12,
113                   color: const Color(0xff707B81)),
114             ), // Text
115           ), // GestureDetector
116         ],
117       ), // Row
118       const SizedBox(height: 30),
119       logController.loginLoading.value
120         ? const CircularProgressIndicator.adaptive()
121         : CustomButtons(
122           title: 'Sign In',
123           onTap: () async {
124             await logController.onLogin();
125           },

```

```

lib > src > module > auth > sign_in > screen > logins_screens.dart > ...
125         ],
126         ), // CustomButtons
127     ],
128     ), // Column
129     ), // SingleChildScrollView
130     ), // Expanded
131 ],
132 ), // Column
133 ), // SafeArea
134 ), // Obx
135 bottomNavigationBar: SafeArea(
136   minimum: const EdgeInsets.only(
137     left: 20,
138     right: 20,
139     top: 20,
140     bottom: 30,
141   ), // EdgeInsets.only
142   child: GestureDetector(
143     onTap: () {
144       router.go('/register');
145     },
146     child: Row(
147       mainAxisAlignment: MainAxisAlignment.center,
148       children: [
149         Text(
150           "New User? ",
151           style: Theme.of(context)
152             .textTheme
153             .titleMedium!
154             .copyWith(color: const Color(0xff6A6A6A)),
155         ), // Text
156         Text("Create Account ",

```

```

lib > src > module > auth > sign_in > screen > logins_screens.dart > ...
156         Text("Create Account ",
157           style: Theme.of(context).textTheme.titleMedium!.copyWith()) // Text
158       ],
159     ), // Row
160   ), // GestureDetector
161 ), // SafeArea
162 ); // Scaffold
163 }
164 }
165

```

### 3 Register Page

```
lib > src > module > auth > sign_in > screen > register_screen.dart > RegisterScreen > RegisterScreen
Chhoeng Chhun Virak, 3 months ago | 3 authors (Chhoeng Chhun Virak and others)
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:go_router/go_router.dart';
4
5 import '../widget/custom_text.dart';
6 import '../widget/home/custom_buttons.dart';
7 import '../controller/contoller.dart';
8
9 Chhoeng Chhun Virak, 3 months ago | 3 authors (Chhoeng Chhun Virak and others)
10 class RegisterScreen extends StatelessWidget {
11   RegisterScreen({super.key});
12   final logController = Get.put(ControllerSignin());
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       appBar: AppBar(
17         elevation: 0,
18       ), // AppBar
19       body: Padding(
20         padding: const EdgeInsets.only(left: 20, right: 20, bottom: 30),
21         child: Column(
22           children: [
23             Expanded(
24               child: SingleChildScrollView(
25                 child: Column(
26                   crossAxisAlignment: CrossAxisAlignment.center,
27                   mainAxisAlignment: MainAxisAlignment.start,
28                   children: [
29                     Text(
30                       "Register Account",
31                       style: Theme.of(context)
```

```
lib > src > module > auth > sign_in > screen > register_screen.dart > RegisterScreen > build
31 .textTheme
32 .titleSmall!
33 .copyWith(fontWeight: FontWeight.w700, fontSize: 32),
34 ), // Text
35 const SizedBox(
36   height: 10,
37 ), // SizedBox
38 Text(
39   "Please fill your details to create an account",
40   textAlign: TextAlign.center,
41   style: Theme.of(context).textTheme.titleMedium!.copyWith(
42     fontWeight: FontWeight.w400,
43     color: const Color(0xff707B81)),
44 ), // Text
45 const SizedBox(
46   height: 40,
47 ), // SizedBox
48 Obx(
49   () => Column(
50     children: [
51       CustomTextFiled(
52         title: 'FirstName',
53         controller:
54           logController.registerFirstNameTxtController,
55         hintText: 'FirstName',
56         isValidate: false,
57       ), // CustomTextFiled
58       const SizedBox(height: 15),
59       CustomTextFiled(
60         title: 'LastName',
61         controller:
62           logController.registerLastNameTxtController,
```

```

lib > src > module > auth > sign_in > screen > register_screen.dart > RegisterScreen > build
63     hintText: 'LastName',
64     isValidate: false,
65   ), // CustomTextFiled
66   const SizedBox(height: 15),
67   CustomTextFiled(
68     title: 'Email',
69     controller:
70       logController.registerEmailTxtController,
71     hintText: 'Email',
72     isValidate: false,
73   ), // CustomTextFiled
74   Padding(
75     padding:
76       const EdgeInsets.only(top: 15.0, bottom: 30.0),
77     child: CustomTextFiled(
78       title: 'Password',
79       controller:
80         logController.registerPasswordTxtController,
81       hintText: 'Password',
82       maxlenght: 15,
83       isObscureText: !logController.isResPass.value,
84       suffixIcon: IconButton(
85         onPressed: () {
86           debugPrint(
87             'isVisibility Pass: ${logController.isResPass.value}');
88           logController.isResPass.value =
89             !logController.isResPass.value;
90         },
91         icon: Icon(logController.isResPass.value
92           ? Icons.visibility
93           : Icons.visibility_off)), // Icon // IconButton
94     isValidate: false,

```

```

lib > src > module > auth > sign_in > screen > register_screen.dart > RegisterScreen > build
95     ), // CustomTextFiled
96   ), // Padding
97 ],
98 ), // Column
99 ), // Obx
100 ],
101 ), // Column
102 ), // SingleChildScrollView
103 ), // Expanded
104 Column(
105   children: [
106     Obx(
107       () => logController.signUpLoading.value
108       ? const CircularProgressIndicator()
109       : CustomButtons(
110         title: 'Sign Up',
111         onTap: () {
112           logController.onRegister();
113         },
114       ), // CustomButtons
115     ), // Obx
116     const SizedBox(
117       height: 20,
118     ), // SizedBox
119     Row(
120       mainAxisAlignment: MainAxisAlignment.center,
121       children: [
122         Text(
123           "Already Have Account? ",
124           style: Theme.of(context).textTheme.titleMedium!.copyWith(
125             fontWeight: FontWeight.w500,
126             color: const Color(0xff6A6A6A)),

```

```
lib > src > module > auth > sign_in > screen > register_screen.dart > RegisterScreen > build
126         color: const Color(0xff6A6A6A),
127     ), // Text
128     GestureDetector(
129         onTap: () {
130             context.go("/login");
131         },
132         child: Text("Log In",
133             style: Theme.of(context).textTheme.titleMedium!), // Text
134     ) // GestureDetector
135 ],
136 ), // Row
137 ],
138 ) // Column
139 ],
140 ), // Column
141 ), // Padding
142 ); // Scaffold
143 }
144 }
145 }
```

## 4 HomePage

```
1 import 'dart:async';
2 import 'package:allpay/src/constant/app_setting.dart';
3 import 'package:allpay/src/module/home/widgets/custom_button_category.dart';
4 import 'package:allpay/src/module/profile/controller/profile_controller.dart';
5 import 'package:cached_network_image/cached_network_image.dart';
6 import 'package:flutter/material.dart';
7 import 'package:flutter_svg/flutter_svg.dart';
8 import 'package:get/get.dart';
9 import 'package:go_router/go_router.dart';
10 import 'package:smooth_page_indicator/smooth_page_indicator.dart';
11 import '../controllers/home_controller.dart';
12 import '../widgets/custom_product_cart.dart';
13 class HomePage extends StatefulWidget {
14     const HomePage({
15         Key? key,
16     }) : super(key: key);
17
18     @override
19     State<HomePage> createState() => _HomePageState();
20 }
21
22 class _HomePageState extends State<HomePage> {
23     @override
24     void initState() {
25         homeController.currentPage.value = 0;
26         homeController.getCategory();
27         homeController.getProduct(
28             page: homeController.currentPage.value,
29             query: '',
30         );
31     }
32 }
```



```

31     homeController.getSlide();
32
33     super.initState();
34 }
35
36 Future<void> _refresh() async {
37     homeController.currentPage.value = 0;
38     await homeController.getCategory();
39     await homeController.getProduct(
40         page: homeController.currentPage.value,
41         query: '',
42     );
43     await homeController.getSlide();
44 }
45
46 final profileCon = Get.put(ProfileController());
47 final homeController = Get.put(HomeController());
48 PageController pageControllers = PageController(initialPage: 0);
49 int index = 0;
50 Timer? timer;
51
52 @override
53 Widget build(BuildContext context) {
54     return Scaffold(
55         backgroundColor: AppColor.backgroundColor,
56
57         // floatingActionButton: FloatingActionButton(onPressed: () {...
61         appBar: AppBar(
62             centerTitle: true,
63             title: SvgPicture.asset('assets/svg/homepage-logo.svg'),
64             elevation: 0,
65         ), // AppBar

```

```

66 //=====> Body <=====
67 body: Obx(
68   () => homeController.loadingFetchAllProduct.value &&
69     homeController.currentPage.value == 0
70   ? const Center(
71     child: CircularProgressIndicator(
72       color: Colors.red,
73     )) // CircularProgressIndicator // Center
74   : RefreshIndicator(
75     onRefresh: _refresh,
76     child: SingleChildScrollView(
77       physics: const AlwaysScrollableScrollPhysics(),
78       padding: const EdgeInsets.only(top: 20),
79       child: Column(
80         crossAxisAlignment: CrossAxisAlignment.start,
81         children: [
82           //Search
83           Padding(
84             padding: const EdgeInsets.only(
85               left: 20, right: 10, bottom: 0), // EdgeInsets.only
86             child: Row(
87               children: [
88                 Expanded(
89                   child: GestureDetector(
90                     onTap: () {

```

```

90                     child: Row(
91                       children: [
92                         Expanded(
93                           child: GestureDetector(
94                             onTap: () {
95                               context.push('/go-search');
96                             },
97                           child: Container(
98                             padding: const EdgeInsets.symmetric(
99                               horizontal: 15,
100                              vertical: 15,
101                            ),
102                             decoration: BoxDecoration(
103                               color: Colors.white,
104                               borderRadius: BorderRadius.circular(14),
105                               boxShadow: [
106                                 BoxShadow(
107                                   color: Colors.black.withOpacity(.04),
108                                   blurRadius: 4,
109                                   offset: const Offset(0, 4),
110                                 )
111                               ],
112                             ),
113                           child: Row(
114                             children: [
115                               SvgPicture.asset(
116                                 'assets/svg/search_icon.svg'),
117                               const SizedBox(width: 30),
118                               const Text(
119                                 'Looking for shoes',
120                                 style: TextStyle(
121                                   fontSize: 14,

```

```

122         color: Colors.appBarColor,
123         fontWeight: FontWeight.w500,
124     ),
125 ),
126 ],
127 ),
128 ),
129 ),
130 ),
131 IconButton(
132     onPressed: () {},
133     icon: Icon(Icons.tune_rounded))
134 ],
135 ),
136 ),
137
138 //=====>Select Category<=====
139 const SizedBox(height: 20),
140 if (homeController.categoryList.isNotEmpty)
141     Padding(
142         padding: const EdgeInsets.only(left: 20),
143         child: customTitle(context, title: 'Select Category'),
144     ),
145 const SizedBox(height: 20),
146
147 SingleChildScrollView(
148     scrollDirection: Axis.horizontal,
149     padding: const EdgeInsets.symmetric(horizontal: 20),
150     child: Row(
151         children: homeController.categoryList
152             .asMap()
153             .entries
154             .map(
155                 (e) => Padding(
156                     padding: const EdgeInsets.only(right: 16),
157                     child: CustomCategoryCard(
158                         title: e.value.name,
159                         onTap: () {
160                             context.push(
161                                 '/category?id=${e.value.id}&name=${e.value.name}',
162                             );
163                         },
164                     ),
165                 ),
166             )
167             .toList(),
168     ),
169 ),
170
171 //=====>Popular Shoes<=====
172 const SizedBox(
173     height: 20,
174 ),
175 if (homeController.productList.isNotEmpty)
176     Padding(
177         padding: const EdgeInsets.only(left: 20),
178         child: customTitle(context,
179             title: 'Popular Shoes',
180             isSeeAll: true, onTap: () {
181                 homeController.currentPage.value = 0;
182                 homeController.getProduct(
183                     page: homeController.currentPage.value,
184                     query: '',

```

```

185     );
186     context.push('/popular-router');
187   }},
188   ),
189
190   SingleChildScrollView(
191     scrollDirection: Axis.horizontal,
192     padding: const EdgeInsets.only(right: 20),
193     child: Row(
194       children: homeController.productList
195         .asMap()
196         .entries
197         .map((e) {
198           return Padding(
199             padding: const EdgeInsets.only(
200               left: 20,
201               right: 5,
202               top: 15,
203             ),
204             child: CustomProductCart(
205               title: e.value.name,
206               image: e.value.thumbnailUrl,
207               price: e.value.price,
208               onAdd: () {
209                 context.push('/detail/${e.value.id}');
210               },
211               // isFav: e.value.,
212               onTap: () {
213                 context.push('/detail/${e.value.id}');
214               },
215               // onFav: () {
216               // // homeController.detailsModelList[index].isFav =

```

```

216               // // homeController.detailsModelList[index].isFav =
217               // // !homeController.detailsModelList[index].isFav!;
218               // // if (homeController.detailsModelList[index].isFav!) {
219               // //   homeController.favCartList
220               // //     .add(homeController.detailsModelList[index]);
221               // // } else {
222               // //   homeController.favCartList.removeAt(index);
223               // //   homeController.update();
224               // // }
225               // // homeController.update();
226               // // },
227               // onAdd: () {
228               // // Navigator.push(context,
229               // //   MaterialPageRoute(builder: (context) {
230               // //     return Container();
231               // //   }));
232               // // },
233             ),
234           );
235         }).toList(),
236     ),
237   ),
238
239   //=====> New Arrivals Slide<=====
240   if (homeController.slidelist.isNotEmpty)
241     Padding(
242       padding: const EdgeInsets.only(left: 20, top: 20),
243       child: customTitle(
244         context,
245         title: 'New Arrivals',
246         isSeeAll: true,
247         onTap: () {

```



```
279         fit: BoxFit.cover,
280         image: CachedNetworkImageProvider(
281             image),
282     ),
283     : null,
284 ),
285 ),
286 );
287 },
288 ),
289 ),
290 if (homeController.slideList.length > 1)
291   Positioned(
292     bottom: 30,
293     child: SmoothPageIndicator(
294       controller: pageControllers,
295       count: homeController.slideList.length,
296       effect: const ExpandingDotsEffect(
297         dotWidth: 8,
298         dotHeight: 8,
299         dotColor: Colors.amber,
300         activeDotColor: Colors.white),
301     ),
302   ),
303 ],
304 ),
305 ],
306 ),
307 ),
308 ),
309 ),
310 );
```

```
309 ),
310 );
311 }
312 }
313 }
```

## B. Database Table

តារាងខាងក្រោមនេះគឺជាប្រព័ន្ធប្រើប្រាស់ដែលមានតួនាទីជា Admin។ Admin មានសិទ្ធិគ្រប់គ្រង ទិន្នន័យទាំងអស់តាមរយៈតារាងរូបភាពដូចខាងក្រោម៖

### - តារាង User Roles

	id [PK] integer	name character varying (255)	createdAt timestamp with time zone	updatedAt timestamp with time zone
1	1	USER	2023-08-05 06:10:11.874+00	2023-08-05 06:10:11.874+00
2	2	ADMIN	2023-08-05 06:10:11.874+00	2023-08-05 06:10:11.874+00

### - តារាង Products

id [PK] integer	name character varying (255)	description text	price double precision	discount double precision	thumbnail_url text	created_at timestamp with time zone	updated_at timestamp with time zone	product_section_id integer	user_id integer	category_id integer
1	7	Nike Blazer Mid '77 Vintage	30.77	0	https://static.nike.com/a/images/t/default/38967...	2023-08-05 12:03:12.695+00	2023-08-05 12:12:10.789+00	[null]	1	2
2	8	Nike Blazer Mid '77 Vintage	30.77	0	https://static.nike.com/a/images/t/default/101af8...	2023-08-05 12:18:21.475+00	2023-08-05 12:18:21.475+00	[null]	1	2
3	9	Nike Dunk Low By You	44.5	0	https://static.nike.com/a/images/t_PDP_864_v1/...	2023-08-08 12:57:37.878+00	2023-08-08 12:57:37.878+00	[null]	1	2
4	10	Nike Dunk Low Retro	44.5	0	https://static.nike.com/a/images/t_PDP_1728_v1/...	2023-08-08 13:03:09.573+00	2023-08-08 13:03:09.573+00	[null]	1	2
5	11	Nike Blazer Mid '77 Vintage	56.5	0	https://static.nike.com/a/images/t_PDP_1728_v1/...	2023-08-21 01:48:56.396+00	2023-08-21 01:48:56.396+00	[null]	1	2
6	12	Freak 5	46.5	0	https://static.nike.com/a/images/t_PDP_1728_v1/...	2023-08-21 01:57:20.758+00	2023-08-21 01:57:20.758+00	[null]	1	4
7	13	Freak 6	46.5	0	https://static.nike.com/a/images/t_PDP_1728_v1/...	2023-08-21 02:01:46.26+00	2023-08-21 02:01:46.26+00	[null]	1	4
8	14	Air Jordan 1 Low OG	39.5	0	https://static.nike.com/a/images/t_PDP_1728_v1/...	2023-08-21 02:06:02.041+00	2023-08-21 02:06:02.041+00	[null]	1	3

### - តារាង Cart

id [PK] integer	price double precision	quantity integer	created_at timestamp with time zone	updated_at timestamp with time zone	variant_id integer	product_id integer	size_id integer	user_id integer
1	2	44.5	2	2023-08-08 12:58:36.247+00	2023-08-14 15:12:16.36+00	3	9	3
2	5	30.77	2	2023-08-09 10:38:49.737+00	2023-08-14 15:55:29.897+00	2	8	2
3	7	44.5	1	2023-08-10 05:55:05.897+00	2023-08-10 05:55:05.897+00	3	9	5
4	8	30.77	1	2023-08-14 15:55:49.303+00	2023-08-14 15:55:49.303+00	1	7	2
5	9	30.77	1	2023-08-14 16:13:16.901+00	2023-08-14 16:13:16.901+00	1	7	2
6	17	44.5	1	2023-08-17 17:01:13.472+00	2023-08-17 17:01:13.472+00	3	9	2
7	28	30.77	1	2023-08-21 11:26:18.723+00	2023-08-21 11:26:18.723+00	2	8	7

### - តារាង Category

id [PK] integer	name text	icon text	created_at timestamp with time zone	updated_at timestamp with time zone
1	All Shoes	https://storage.googleapis.com/e-commerce-app-storage/download.jpg	2023-08-05 07:27:36.205+00	2023-08-16 02:46:47.709+00
2	Man	https://storage.googleapis.com/e-commerce-app-storage/download.jpg	2023-08-05 11:57:01.449+00	2023-08-16 02:47:47.195+00
3	Women	https://storage.googleapis.com/e-commerce-app-storage/download.jpg	2023-08-16 02:43:45.03+00	2023-08-16 02:48:44.352+00
4	Basketball	https://storage.googleapis.com/e-commerce-app-storage/download.jpg	2023-08-16 02:49:00.194+00	2023-08-16 02:49:00.194+00
5	Baby/Toddler	https://storage.googleapis.com/e-commerce-app-storage/download.jpg	2023-08-16 02:50:02.38+00	2023-08-16 02:50:02.38+00
6	Big Kids	https://storage.googleapis.com/e-commerce-app-storage/download.jpg	2023-08-16 02:50:53.651+00	2023-08-16 02:50:53.651+00
7	Shoe	https://storage.googleapis.com/e-commerce-app-storage/IMAGE_2023-04-27_14:32:59.j...	2023-08-17 16:27:37.405+00	2023-08-17 16:27:37.405+00

### - តារាង Variant

id [PK] integer	name character varying (255)	product_name character varying (255)	value character varying (255)	image_url text	created_at timestamp with time zone	updated_at timestamp with time zone	product_id integer
1	White/Black shoes	Nike Blazer Mid '77 Vintage - White/Black shoes	#C92724	https://st...	2023-08-05 12:03:12.726+00	2023-08-05 12:03:12.726+00	7
2	White/Black shoes	Nike Blazer Mid '77 Vintage - White/Black shoes	#C92724	https://st...	2023-08-05 12:18:21.503+00	2023-08-05 12:18:21.503+00	8
3	Red Shoe	Nike Dunk Low By You - Red Shoe	#C92724	https://w...	2023-08-08 12:57:37.935+00	2023-08-08 12:57:37.935+00	9
4	White/White/Clear Jade	Nike Dunk Low Retro - White/White/Clear Jade	#C92724	https://st...	2023-08-08 13:03:09.896+00	2023-08-08 13:03:09.896+00	10
5	White/Wolf Grey/Black/Safety Orange	Nike Blazer Mid '77 Vintage - White/Wolf Grey/Black/Safety Oran...	#0000366	https://st...	2023-08-16 03:38:37.983+00	2023-08-16 03:46:33.967+00	7
6	Red Shoe	Nike Blazer Mid '77 Vintage - Red Shoe	#C92724	https://st...	2023-08-21 01:48:56.458+00	2023-08-21 01:48:56.458+00	11
7	Red Shoe	Freak 5 - Red Shoe	#C92724	https://st...	2023-08-21 01:57:20.771+00	2023-08-21 01:57:20.771+00	12

### - តារាង Order

	id [PK] integer	total_amount double precision	created_at timestamp with time zone	updated_at timestamp with time zone	orderstatus_id integer	deliveryaddress_id integer	user_id integer
1	1	32.2	2023-08-14 16:06:41.141+00	2023-08-14 16:06:41.141+00	1	1	2
2	2	32.2	2023-08-21 02:16:10.313+00	2023-08-21 02:16:10.313+00	1	1	1
3	3	32.2	2023-08-21 06:44:40.529+00	2023-08-21 06:44:40.529+00	1	1	1
4	4	32.2	2023-08-21 07:06:25.271+00	2023-08-21 07:06:25.271+00	1	1	1
5	5	32.2	2023-08-21 07:06:36.478+00	2023-08-21 07:06:36.478+00	1	1	1
6	6	32.2	2023-08-21 07:07:11.905+00	2023-08-21 07:07:11.905+00	1	1	1
7	7	56.2	2023-08-21 07:19:34.746+00	2023-08-21 07:19:34.746+00	1	1	1
8	8	32.2	2023-08-21 07:57:44.17+00	2023-08-21 07:57:44.17+00	1	1	7
9	9	137.5	2023-08-21 08:02:49.912+00	2023-08-21 08:02:49.912+00	1	1	7
10	10	40.5	2023-08-21 08:03:11.034+00	2023-08-21 08:03:11.034+00	1	1	7
11	11	56.5	2023-08-21 08:03:18.915+00	2023-08-21 08:03:18.915+00	1	1	7

- តារាង Order Status

	id [PK] integer	status character varying (255)	createdAt timestamp with time zone	updatedAt timestamp with time zone
1	1	New	2023-08-05 06:15:00.428+00	2023-08-05 06:15:00.428+00
2	2	Accepted	2023-08-05 06:15:07.061+00	2023-08-05 06:15:07.061+00
3	3	Prepared	2023-08-05 06:15:16.585+00	2023-08-05 06:15:16.585+00
4	4	Delivering	2023-08-05 06:15:23.551+00	2023-08-05 06:15:23.551+00
5	5	Delivered	2023-08-05 06:15:30.391+00	2023-08-05 06:15:30.391+00
6	6	Paid	2023-08-05 06:15:35.145+00	2023-08-05 06:15:35.145+00

- តារាង Delivery address

	id integer	first_name text	last_name text	phone text	street_no text	home_no text	full_address text	latitude double precision	longitude double precision	label text	is_default boolean	created_at timestamp with time zone	updated_at timestamp with time zone	user_id integer	
1	1	[null]	[null]	[null]	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	Work	false	2023-08-06 01:15:21.671+00	2023-08-21 14:51:...	1
2	2	Tinut	Chan	+855167636783	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	Work	false	2023-08-06 01:17:02.073+00	2023-08-21 14:51:...	1
3	3	Tinut	Chan	+855167636783	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	Work	false	2023-08-06 01:49:36.237+00	2023-08-21 14:51:...	1
4	4	Tinut	Chan	+855167636783	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	Home	false	2023-08-06 02:03:56.272+00	2023-08-21 14:51:...	1
5	5	Tinut	Chan	+XXXXXXXXXX	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	School	false	2023-08-06 05:58:48.627+00	2023-08-21 14:51:...	1
6	6	Tinut	Chan	+XXXXXXXXXX	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	School	false	2023-08-06 06:26:49.786+00	2023-08-21 14:51:...	4
7	7	Tinut	Chan11	+855 16873783	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	School	false	2023-08-10 11:52:41.258+00	2023-08-21 14:51:...	4
8	8	Sorn	Nantib	+855 96 8805230	110	10A	st 7A, yeay sar, tonlebet, th...	122.444		11	School	false	2023-08-12 01:53:13.019+00	2023-08-21 14:51:...	4

- តារាង Payment

	id [PK] integer	amount double precision	payment_method text	created_at timestamp with time zone	updated_at timestamp with time zone	order_id integer	user_id integer
1	1	32.2	Cash on delivery	2023-08-21 02:19:26.963+00	2023-08-21 02:19:26.963+00	2	[null]
2	2	32.2	Cash on delivery	2023-08-21 02:22:57.517+00	2023-08-21 02:22:57.517+00	2	[null]
3	3	100	Cash on delivery	2023-08-21 06:43:42.82+00	2023-08-21 06:43:42.82+00	2	[null]

- តារាង Profile

	id [PK] integer	first_name text	last_name text	phone_number text	email text	date_of_birth timestamp with time zone	password text	is_instructor boolean	bio text	skills text[]	created_at timestamp with time zone	updated_at timestamp with time zone	role_id integer	gender_id integer
1	1	Srun	Tola	[null]	admin@gmail.co...	[null]	\$2a\$08\$XVSIk34fL...	false	[null]	[]	2023-08-05 06:10:12.453+00	2023-08-05 06:10:12.453+00	2	[null]
2	2	virak	chhun	[null]	virak@gmail.com	[null]	\$2a\$08\$QyEEVBFV...	false	[null]	[]	2023-08-05 06:54:53.98+00	2023-08-05 06:54:53.98+00	1	[null]
3	3	kea	kea	[null]	kea@gmail.com	2002-02-0...	\$2a\$08\$YbSKLTjV...	false	[null]	[]	2023-08-05 13:45:17.751+00	2023-08-05 14:10:42.449+00	1	[null]
4	4	Tinut	Chan	[null]	tinutchan@gmai...	2034-02-0...	\$2a\$08\$N4fnr.csv/...	false	[null]	[]	2023-08-06 03:32:59.821+00	2023-08-17 06:36:20.001+00	1	[null]
5	5	Kea	Kea	[null]	Kea@gmail.com	[null]	\$2a\$08\$HUYLkFM...	false	[null]	[]	2023-08-10 05:39:47.098+00	2023-08-10 05:39:47.098+00	1	[null]
6	6	Sorn	Nantib	[null]	sornnantib@gm...	1988-04-0...	\$2a\$08\$GWXkkyS...	false	[null]	[]	2023-08-17 16:24:15.514+00	2023-08-21 14:52:51.803+00	1	[null]
7	7	Sophea	Sophanith	[null]	sopheasophanit...	[null]	\$2a\$08\$ve0R/Q2A...	false	[null]	[]	2023-08-21 07:33:15.797+00	2023-08-21 07:33:15.797+00	1	[null]



