

**ព្រះរាជាណាចក្រកម្ពុជា**

**ជាតិ សាសនា ព្រះមហាក្សត្រ**

**សាកលវិទ្យាល័យនំរតុន**

**មហាវិទ្យាល័យ វិទ្យាសាស្ត្រ**

**ផ្នែក: កុំព្យូទ័រវិទ្យា**

**ជំនាញ: វិទ្យាសាស្ត្រកុំព្យូទ័រ / អភិវឌ្ឍន៍កម្មវិធី - COM3MS2**

**មុខវិជ្ជា: ប្រព័ន្ធឯកទេសជំនាញ**

# **ការរចនា និងការអភិវឌ្ឍប្រព័ន្ធជំនាញណែនាំការជ្រើស រើសស្មាតហ្វូន និងកុំព្យូទ័រតាមប្រព័ន្ធបណ្តាញ ដោយប្រើ ប្រាស់ Flask និង MySQL**

**Design and Development of a Web-Based Expert System for  
Smartphone and Laptop Recommendations using Flask and MySQL**

**រៀបរៀង និងចងក្រងដោយ:**

**១. សេង សុខអាង – អ្នកគ្រប់គ្រងគម្រោង – B20245713**

**២. លីម រតនា – អ្នកអភិវឌ្ឍន៍ផ្នែកខាងក្រោយ – Bo. 014350**

**៣. ជ័យ ច័ន្ទសុមុនី – អ្នកអភិវឌ្ឍន៍ខាងមុខ – B20236436**

**បង្រៀន និងដឹកនាំដោយសាស្ត្រាចារ្យ**

**សេក សុជាតិ**

**ថ្ងៃទី ២៤ ខែធ្នូ ឆ្នាំ២០២៥ (ថ្ងៃខែឆ្នាំជាក់សំណើកិច្ចការស្រាវជ្រាវ)**

- ឈ្មោះប្រទេសជាតិ: ព្រះរាជាណាចក្រកម្ពុជា
- បាវចនាប្រទេស: ជាតិ សាសនា ព្រះមហាក្សត្រ
- សាកលវិទ្យាល័យ: សាកលវិទ្យាល័យនំរតុន
- មហាវិទ្យាល័យ/ដេប៉ាតឺម៉ង់: ផ្នែកកុំព្យូទ័រវិទ្យា នៃមហាវិទ្យាល័យវិទ្យាសាស្ត្រ
- មុខវិជ្ជា/ឯកទេសជំនាញ: ការអភិវឌ្ឍប្រព័ន្ធឯកទេសជំនាញ (Expert System Development)
- ចំណងជើងគម្រោងស្រាវជ្រាវ: “ការរចនា និងការអភិវឌ្ឍប្រព័ន្ធឯកទេសជំនាញណែនាំការជ្រើសរើសស្ថាប័ន និងកុំព្យូទ័រតាមប្រព័ន្ធបណ្តាញ ដោយប្រើប្រាស់ Flask និង MySQL”
- ប្រភេទគម្រោង: Web Application / Expert System (AI)
- រៀបចំដោយ (សមាជិកក្រុម):
  - ១. សេង សុខរាង – អ្នកគ្រប់គ្រងគម្រោង – B20245713
  - ២. លីម រតនា – អ្នកអភិវឌ្ឍន៍ផ្នែកក្រោយ – B20236572
  - ៣. ជ័យ ច័ន្ទសុម្ពិ – អ្នកអភិវឌ្ឍន៍ផ្នែកខាងមុខ – B20236436
- គ្រូបង្រៀន/អ្នកណែនាំ: សេក សុជាតិ
- ឆ្នាំសិក្សា/ថ្នាក់/ក្រុម: ឆ្នាំទី៣ ឆមាសទី១ វេនព្រឹក (COM3MS2)
- កាលបរិច្ឆេទដាក់សំណើ: ថ្ងៃទី២៤ ខែធ្នូ ឆ្នាំ២០២៥

## Contents

មូលសង្ខេប (Abstract).....	5
១. បរិបទ និងមូលហេតុ (Background & Rationale).....	6
១.១ បរិបទទូទៅ.....	6
១.២ បញ្ហាដែលជួបប្រទះ.....	6
១.៣ មូលហេតុជ្រើសរើសគម្រោងនេះ.....	6
២. គោលបំណង (Objectives) .....	7
២.១ គោលបំណងទូទៅ (General Objective) .....	7
២.២ គោលបំណងជាក់លាក់ (Specific Objectives) .....	7
៣. វិសាលភាពគម្រោង (Project Scope).....	7
៣.១ ក្នុងវិសាលភាព (In-Scope) .....	7
៣.២ ក្រៅវិសាលភាព (Out of Scope).....	8
៤. អ្នកប្រើប្រាស់ និងតួនាទី (Users & Roles) .....	8
៥. តម្រូវការ (Requirements) .....	9
៥.១ តម្រូវការមុខងារ (Functional Requirements).....	9
៥.២ មិនមែនតម្រូវការមុខងារ (Non-Functional) .....	9
៦. វិធីសាស្ត្រអភិវឌ្ឍន៍ (METHODOLOGY / SDLC).....	10
៦.១ ការជ្រើសរើសគំរូ SDLC.....	10
៦.២ ហេតុផលសម្រាប់ការប្រើប្រាស់ Agile .....	11
៦.៣ ផែនការអនុវត្តតាមវដ្ត (Sprint Planning).....	11
៧. បច្ចេកវិទ្យា និងឧបករណ៍ (Technology Stack) .....	12
៨. រចនាសម្ព័ន្ធប្រព័ន្ធ (System Design) .....	13

៨.២ Architecture Overview .....	14
៨.២ Database Design (ERD) .....	15
៨.៣ UI/UX Mockup .....	16
៩. ផែនការអនុវត្ត និងកាលវិភាគ (Work Plan & Timeline).....	18
១០. លទ្ធផលរំពឹងទុក (Expected Outputs/Deliverables).....	21
១១. ការធានាគុណភាព និងការតេស្ត (QA & Testing).....	21
១២. ហានិភ័យ និងយុទ្ធសាស្ត្រកាត់បន្ថយ (Risks & Mitigation).....	22
១៣. តួនាទីក្រុម (Team Roles & Responsibilities) .....	22
១៤. ថវិកា (Budget) (ប្រសិនបើត្រូវការ).....	23
១៥. សេចក្តីសន្និដ្ឋាន (Conclusion) .....	23
ឯកសារយោង (References) .....	23
ឧបសម្ព័ន្ធ (Appendices).....	27

# ការរចនា និងការអភិវឌ្ឍប្រព័ន្ធជំនាញណែនាំការជ្រើសរើសស្ពាន់តូប និងកុំព្យូទ័រតាម ប្រព័ន្ធបណ្តាញ ដោយប្រើប្រាស់ Flask និង MySQL

និពន្ធនាយក និងស្ថាប័នពាក់ព័ន្ធ (Authors and Affiliations)

Seng Sok Ang <sup>1</sup>                      Lim Ratana <sup>2</sup>                      Chey Chansomony <sup>3</sup>                      សេក សុជាតិ <sup>4</sup>  
[sengsokang93@gmail.com](mailto:sengsokang93@gmail.com)    [lrathana888@gmail.com](mailto:lrathana888@gmail.com)    [chevchansomony314@gmail.com](mailto:chevchansomony314@gmail.com)    [socheat.sek@gmail.com](mailto:socheat.sek@gmail.com)

<sup>1, 2, 3, 4</sup> Students of Expert System, DCS of the Faculty of Science, Norton University (Year 3, Semester 1, 2025 - 2026)

<sup>5</sup> Lecturer of Expert System Development, DCS of the Faculty of Science, Norton University

## មូលសង្ខេប (Abstract)

ដើម្បីដោះស្រាយបញ្ហានេះ គម្រោងនេះមានគោលបំណងអភិវឌ្ឍន៍ "ប្រព័ន្ធជំនាញណែនាំការជ្រើសរើសស្ពាន់តូប និងកុំព្យូទ័រ" (Smartphone/Laptop Buying Advisor) សម្រាប់អ្នកប្រើប្រាស់ទូទៅ ដើម្បីជួយណែនាំផលិតផលដែលសមស្របបំផុតដោយប្រើប្រាស់បច្ចេកវិទ្យា Web-Based Expert System ។ ដំណោះស្រាយដែលបានស្នើឡើងគឺការប្រើប្រាស់ Inference Engine ប្រភេទ Forward Chaining ដើម្បីវិភាគតម្រូវការរបស់អ្នកប្រើ (ថវិកា, ការប្រើប្រាស់, ម៉ាក) និង ផ្តល់ជម្រើសជ្រើសរើស (Rules) ក្នុងមូលដ្ឋានចំណេះដឹង (Knowledge Base) (MDPI, 2020) ។

ប្រព័ន្ធនេះត្រូវបានអភិវឌ្ឍឡើងដោយប្រើប្រាស់បច្ចេកវិទ្យាទំនើប និងស្រាល (Lightweight) រួមមាន Flask Framework (Python) សម្រាប់គ្រប់គ្រងតក្កវិជ្ជា Backend (Flask Docs, 2024), MySQL សម្រាប់គ្រប់គ្រងប្រព័ន្ធទិន្នន័យទំនាក់ទំនង (RDBMS) (Oracle, 2023), និង CSS 3 សម្រាប់ការរចនាចំណុចប្រទាក់ដែលឆ្លើយតបគ្រប់ឧបករណ៍ (CSS 3, 2023) ។

លទ្ធផលរំពឹងទុកនៃគម្រោងនេះ គឺការបង្កើតបាននូវ Web Application មួយដែលអាចដើរតួជាជំនួយការបច្ចេកទេសឆ្លាតវៃ (Intelligent Assistant) ដែលផ្តល់អត្ថប្រយោជន៍សំខាន់ៗដូចជា៖ ការកាត់បន្ថយពេលវេលាស្រាវជ្រាវរបស់អ្នកប្រើប្រាស់, ការផ្តល់អនុសាសន៍ដែលមានតម្លាភាព និងផ្អែកលើតក្កវិជ្ជាសុទ្ធសាធ, និងការជួយសិស្សនិស្សិតក្នុងការជ្រើសរើសឧបករណ៍ដែលគាំទ្រដល់ការសិក្សារបស់ពួកគេបានយ៉ាងមានប្រសិទ្ធភាព (Cornell, 2023) ។

## ១. បរិបទ និងមូលហេតុ (Background & Rationale)

### ១.១ បរិបទទូទៅ

បច្ចុប្បន្ននេះ បច្ចេកវិទ្យាបានក្លាយជាផ្នែកមួយដែលមិនអាចខ្វះបានក្នុងជីវិតប្រចាំថ្ងៃ ជាពិសេសសម្រាប់វិស័យអប់រំ និងការងារ។ របាយការណ៍ទីផ្សារបានបង្ហាញថា ចំនួនម៉ូដែលស្ថាប័ន និងកុំព្យូទ័រដែលដាក់លក់នៅលើទីផ្សារមានការកើនឡើងយ៉ាងខ្លាំងជាប្រចាំឆ្នាំ (IDC, 2024)។ ភាពសម្បូរបែបនេះ បើទោះបីជាផ្តល់ជម្រើសច្រើនក៏ដោយ ក៏វាបានបង្កើតជាបញ្ហាស្មុគស្មាញសម្រាប់អ្នកប្រើប្រាស់ដែលមិនមានជំនាញបច្ចេកទេស (Non-technical users) ក្នុងការវាយតម្លៃ និងជ្រើសរើសផលិតផលដែលសាកសមនឹងតម្រូវការរបស់ពួកគេ។

### ១.២ បញ្ហាដែលជួបប្រទះ

- **បញ្ហាទី ១៖ ភាពស្មុគស្មាញនៃជម្រើស (Choice Overload):** អ្នកប្រើប្រាស់ជាច្រើនជួបប្រទះនូវស្ថានភាព "Decision Fatigue" ឬភាពនឿយហត់ក្នុងការសម្រេចចិត្ត ដោយសារមានជម្រើសច្រើនពេក និងមិនយល់ពីពាក្យបច្ចេកទេសដូចជា CPU Cores, RAM Speed, ឬ GPU Architecture (Arxiv, 2021)។
- **បញ្ហាទី ២៖ ការចំណាយពេលស្រាវជ្រាវយូរ:** សិស្សនិស្សិត ឬបុគ្គលិកការិយាល័យត្រូវចំណាយពេលវេលាម៉ោងដើម្បីអានការ Review ឬមើលវីដេអូប្រៀបធៀប មុនពេលសម្រេចចិត្តទិញ។
- **បញ្ហាទី ៣៖ ការណែនាំដែលមានភាពលំអៀង:** ការសាកសួរអ្នកលក់តាមហាង ជាញឹកញយទទួលបានចម្លើយដែលលំអៀងទៅរកផលិតផលដែលមានកម្រៃដើមសារខ្ពស់ ជាជាងផលិតផលដែលសមស្របនឹងតម្រូវការអ្នកទិញពិតប្រាកដ។

### ១.៣ មូលហេតុជ្រើសរើសគម្រោងនេះ

ក្រុមការងារបានជ្រើសរើសអភិវឌ្ឍ "ប្រព័ន្ធជំនាញ (Expert System)" នេះឡើង ព្រោះវាមានសមត្ថភាពក្នុងការរក្សាទុកចំណេះដឹងរបស់អ្នកជំនាញ IT និងប្រើប្រាស់តក្កវិជ្ជា Rule-Based Filtering ដើម្បីដោះស្រាយបញ្ហាខាងលើបានយ៉ាងមានប្រសិទ្ធភាព (MDPI, 2020)។ គម្រោងនេះមិនត្រឹមតែជួយអ្នកប្រើប្រាស់ប៉ុណ្ណោះទេ ថែមទាំងអនុញ្ញាតឱ្យក្រុមនិស្សិតអនុវត្តចំណេះដឹងជាក់ស្តែងលើ Flask, MySQL និង Artificial Intelligence (AI)។

## ២. គោលបំណង (Objectives)

### ២.១ គោលបំណងទូទៅ (General Objective)

ដើម្បីរចនា និងអភិវឌ្ឍប្រព័ន្ធជំនាញតាមប្រព័ន្ធបណ្តាញ (Web-Based Expert System) មួយ ដែលមានសមត្ថភាពវិភាគតម្រូវការអ្នកប្រើប្រាស់ និងផ្តល់ការណែនាំអំពីការជ្រើសរើសស្ថាតហ្វូន ឬកុំព្យូទ័របានយ៉ាងត្រឹមត្រូវ និងរហ័សទាន់ចិត្ត។

### ២.២ គោលបំណងជាក់លាក់ (Specific Objectives)

- គ.ប ១ (O1): បង្កើតមូលដ្ឋានចំណេះដឹង (Knowledge Base) ដែលមានទិន្នន័យផលិតផលយ៉ាងតិច ៥០ ម៉ូដែល និងច្បាប់វិនិច្ឆ័យ (Rules) សម្រាប់សេនារីយ៉ូប្រើប្រាស់ផ្សេងៗគ្នា (ដូចជា Gaming, Study, Office Work) (Cornell, 2023)។
- គ.ប ២ (O2): អភិវឌ្ឍន៍ម៉ាស៊ីនវិនិច្ឆ័យ (Inference Engine) ដោយប្រើប្រាស់ Python ដែលអាចផ្តល់លទ្ធផលណែនាំយ៉ាងតិច ៣ ជម្រើស ដោយប្រើសំណួរមិនលើសពី ៥ សំណួរ។
- គ.ប ៣ (O3): បង្កើនប្រសិទ្ធភាពក្នុងដំណើរការស្វែងរក និងសម្រេចចិត្តជ្រើសរើសស្ថាតហ្វូន ឬកុំព្យូទ័រ ដោយជួយកាត់បន្ថយពេលវេលាស្រាវជ្រាវ និងផ្តល់ការណែនាំដែលមានតម្លៃជ្រាលជ្រៅសម្រាប់ការស្វែងរកព័ត៌មានដោយខ្លួនឯង។

## ៣. វិសាលភាពគម្រោង (Project Scope)

### ៣.១ ក្នុងវិសាលភាព (In-Scope)

- ប្រព័ន្ធនែនាំ (Recommendation Engine): មុខងារស្វ័យដែលស្វែងរកប្រើប្រាស់អំពី ថវិកា (Budget), គោលបំណង (Purpose), និង ម៉ាក (Brand Preference) ដើម្បីផ្តល់លទ្ធផល។
- ប្រព័ន្ធប្រៀបធៀប (Comparison Module): មុខងារអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ជ្រើសរើសផលិតផល ២ ដើម្បីមើលលក្ខណៈសម្បត្តិប្រៀបធៀបគ្នានៅលើតារាងតែមួយ (Side-by-side comparison)។
- ការគ្រប់គ្រងទិន្នន័យ (Admin Dashboard): មុខងារសម្រាប់ Admin ក្នុងការ បន្ថែម, លុប, ឬកែប្រែ (CRUD) ទិន្នន័យផលិតផល និងកែប្រែច្បាប់ (Rules) នៅក្នុង Database MySQL ។

## ៣.២ ក្រៅវិសាលភាព (Out of Scope)

- **មិនមានការដាក់ឱ្យប្រើប្រាស់ជាសាធារណៈ (No Public Hosting):** ប្រព័ន្ធនឹងដំណើរការត្រឹមតែនៅលើម៉ាស៊ីនក្នុងស្រុក (Localhost/Flask Dev Server) ប៉ុណ្ណោះ មិនមានការបង្ហាញលើ Cloud Server ទេ។
- **មិនមានមុខងារទូទាត់ប្រាក់ (No Payment Gateway):** ប្រព័ន្ធនេះគ្រាន់តែណែនាំប៉ុណ្ណោះ មិនមានការលក់ដូរផ្ទាល់ទេ។
- **មិនគាំទ្រឧបករណ៍មួយទឹក (No Second-hand Market):** ទិន្នន័យផ្តោតតែលើផលិតផលថ្មី និងតម្លៃទីផ្សារផ្លូវការ។
- **មិនមាន Mobile App (Native):** ប្រព័ន្ធនេះជា Web Application ដែលដំណើរការលើ Browser ប៉ុណ្ណោះ។

## ៤. អ្នកប្រើប្រាស់ និងតួនាទី (Users & Roles)

ឈ្មោះតួនាទី (Role)	សិទ្ធិ/ការងារ (Permissions)
Admin (អ្នកគ្រប់គ្រង)	មានសិទ្ធិពេញលេញលើប្រព័ន្ធ (Full Access) រួមទាំងការគ្រប់គ្រងគណនីអ្នកប្រើប្រាស់ (User Management), ការកំណត់រចនាសម្ព័ន្ធប្រព័ន្ធ, និងការមើលរបាយការណ៍សរុប។
Staff (អ្នកបញ្ចូលទិន្នន័យ)	មានសិទ្ធិអនុវត្តមុខងារ <b>CRUD</b> (Create, Read, Update, Delete) ទៅលើមូលដ្ឋានចំណេះដឹង (Knowledge Base) ដូចជាការបន្ថែមម៉ូដែលថ្មី ឬកែប្រែតម្លៃទីផ្សារ ប៉ុន្តែមិនអាចលុបគណនី Admin បានទេ។
End User (អ្នកប្រើទូទៅ)	មិនតម្រូវឱ្យ Login សម្រាប់ការប្រើប្រាស់មុខងារវិនិច្ឆ័យ (Diagnosis/Recommendation) ទេ។ ពួកគេអាចឆ្លើយសំណួរ, មើលលទ្ធផលណែនាំ, និងប្រើប្រាស់មុខងារប្រៀបធៀបផលិតផលបាន។



## ៥. តម្រូវការ (Requirements)

### ៥.១ តម្រូវការមុខងារ (Functional Requirements)

យោងតាមស្តង់ដារ IEEE 29148 សម្រាប់ការសរសេរតម្រូវការ (IEEE, 2018), ប្រព័ន្ធត្រូវមានមុខងារស្នូលដូចខាងក្រោម៖

- **ត.ក.ម ១ (FR1) - ការប្រមូលព័ត៌មាន (Input Processing):** ប្រព័ន្ធត្រូវផ្តល់នូវចំណុចប្រទាក់ (Form) ដើម្បីឱ្យអ្នកប្រើប្រាស់ជ្រើសរើសលក្ខខណ្ឌដូចជា៖ ដែនកំណត់ថវិកា (Budget Range), ប្រភេទការប្រើប្រាស់ (Gaming, Camera, Study), និងប្រព័ន្ធប្រតិបត្តិការដែលចូលចិត្ត (iOS/Android/Windows) ។
- **ត.ក.ម ២ (FR2) - ម៉ាស៊ីនវិនិច្ឆ័យ (Inference Engine):** ប្រព័ន្ធត្រូវប្រើប្រាស់ក្បួនដោះស្រាយ Forward Chaining ដើម្បីផ្តល់ធាតុចូល (Inputs) ជាមួយច្បាប់ (Rules) ក្នុង Database និងបង្កើតជាលទ្ធផលណែនាំយ៉ាងតិច ៣ ជម្រើស។
- **ត.ក.ម ៣ (FR3) - ការបង្ហាញលទ្ធផល (Result Visualization):** ប្រព័ន្ធត្រូវបង្ហាញព័ត៌មានលម្អិតនៃផលិតផលដែលបានណែនាំ (រូបភាព, តម្លៃ, Specs សំខាន់ៗ) និងមូលហេតុនៃការណែនាំ (Explanation Facility) ។
- **ត.ក.ម ៤ (FR4) - ការគ្រប់គ្រងចំណេះដឹង (Knowledge Management):** Admin ត្រូវតែអាចបន្ថែម ឬកែប្រែច្បាប់វិនិច្ឆ័យ (Rules) តាមរយៈចំណុចប្រទាក់វេបសាយ ដោយមិនចាំបាច់ចូលទៅសរសេរកូដផ្ទាល់ (Hard-coding) ។

### ៥.២ មិនមែនតម្រូវការមុខងារ (Non-Functional)

- **ម.ត.ក.ម ១ (NFR1) - សុវត្ថិភាព (Security):** ប្រព័ន្ធត្រូវការការពារប្រឆាំងនឹងការវាយប្រហារប្រភេទ SQL Injection និង XSS ដោយប្រើប្រាស់មុខងារការពារដែលមានស្រាប់នៅក្នុង Flask និង SQLAlchemy, ស្របតាមស្តង់ដារ OWASP Top 10 (OWASP, 2021) ។
- **ម.ត.ក.ម ២ (NFR2) - ប្រសិទ្ធភាព (Performance):** ប្រព័ន្ធត្រូវមានរយៈពេលឆ្លើយតប (Response Time) តិចជាង ២ វិនាទី សម្រាប់ការដំណើរការ Inference Engine លើទិន្នន័យចំនួន ៥០ ម៉ឺន (ISO 25010) ។

- ម.ត.ក.ម ៣ (NFR3) - ភាពងាយស្រួលប្រើ (Usability/Responsiveness): ចំណុចប្រទាក់អ្នកប្រើប្រាស់ (UI) ត្រូវតែសម្របតាមទំហំអេក្រង់ទូរស័ព្ទ និងកុំព្យូទ័រ (Responsive Design) ដោយស្វ័យប្រវត្តិ ដោយប្រើប្រាស់បច្ចេកវិទ្យា Tailwind ។

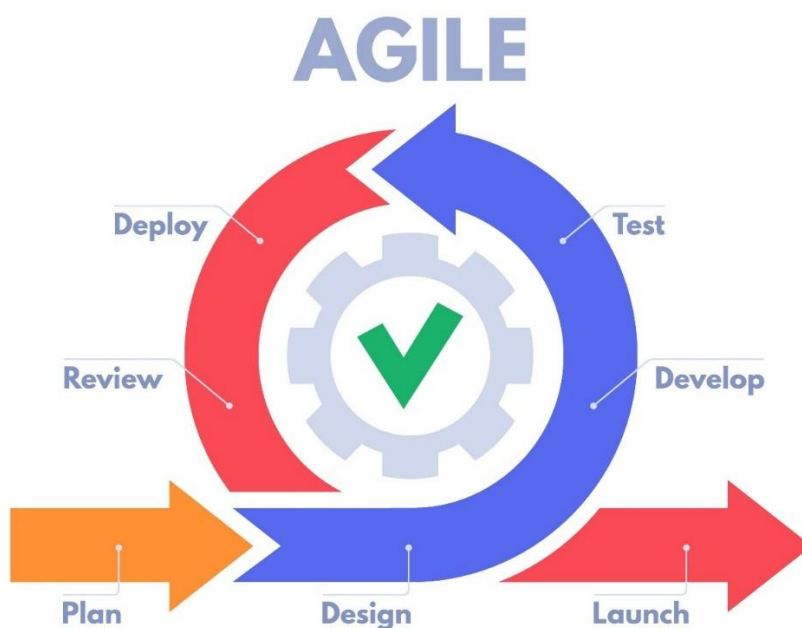
ម.ត.ក.ម ៤ (NFR4) - ភាពអាចទុកចិត្តបាន (Reliability): ប្រព័ន្ធត្រូវមានយន្តការ Validation ដើម្បីធានាថាទិន្នន័យដែល Admin បញ្ចូល (ដូចជាតម្លៃ ឬឈ្មោះម៉ូដែល) គឺត្រឹមត្រូវ និងមិនមានការចម្លងគ្នាមុននឹងរក្សាទុកក្នុង Database ។

## ៦. វិធីសាស្ត្រអភិវឌ្ឍន៍ (METHODOLOGY / SDLC)

### ៦.១ ការជ្រើសរើសគំរូ SDLC

ដើម្បីដោះស្រាយលក្ខណៈពិសេសនៃគម្រោងប្រព័ន្ធជំនាញដែលទាមទារ ការប្រមូលចំណេះដឹង (Knowledge Acquisition) ពីទីផ្សារបច្ចេកវិទ្យាដែលប្រែប្រួលរហ័ស និង ការកែតម្រូវច្បាប់ (Rule Refinement) ឥតឈប់ឈរ ក្រុមការងារបានជ្រើសរើសប្រើ វិធីសាស្ត្រអភិវឌ្ឍន៍អេឌីល (Agile - Iterative Model) ។

វិធីសាស្ត្រនេះអនុញ្ញាតឱ្យក្រុមការងារបែងចែកការងារជា **វដ្តអនុវត្ត (Sprints)** ចំនួន ៤ ដំណាក់កាលសំខាន់ៗ និងធ្វើការតេស្តរាល់ចុងបញ្ចប់នៃវដ្ត (End of each Sprint) ដើម្បីធានាបាននូវការកែលម្អ **មូលដ្ឋានចំណេះដឹង (Knowledge Base)** ឥតឈប់ឈរ និងកាត់បន្ថយហានិភ័យនៃការផ្លាស់ប្តូរតម្រូវការនៅពេលក្រោយ (Scrum Guide, 2020) ។



## ៦.២ ហេតុផលសម្រាប់ការប្រើប្រាស់ Agile

ភាពចាំបាច់របស់ គម្រោង	អត្ថប្រយោជន៍របស់ Agile
តម្រូវការបត់បែនខ្ពស់ (High Flexibility)	ទីផ្សារស្ថាប័ន និងកុំព្យូទ័រមានការផ្លាស់ប្តូរតម្លៃ និងម៉ូដែលថ្មីៗជាញឹកញយ។ Agile អនុញ្ញាតឱ្យយើងធ្វើបច្ចុប្បន្នភាព Rule Set និងទិន្នន័យផលិតផលបានភ្លាមៗ ទោះបីជាស្ថិតក្នុងដំណាក់កាលសរសេរកូដក៏ដោយ។
ការផ្ទៀងផ្ទាត់ចំណេះដឹង (Validation)	យើងអាចទទួលបានមតិកែលម្អពីអ្នកប្រើប្រាស់សាកល្បង (Test Users) នៅចុងបញ្ចប់នៃ Sprint នីមួយៗ ដើម្បីវាយតម្លៃថា តើការណែនាំរបស់ប្រព័ន្ធពិតជាសមហេតុផលនឹងតម្រូវការជាក់ស្តែងដែរឬទេ (Scrum Guide, 2020) ។
កាត់បន្ថយហានិភ័យ (Risk Reduction)	ការធ្វើតេស្តដំបូង (Early Testing) ជួយឱ្យយើងរកឃើញកំហុសនៅក្នុង Inference Engine Logic (ឧទាហរណ៍៖ ការណែនាំខុសម៉ាក ឬខុសកម្រិតថវិកា) តាំងពីដើមដំបូង មុនពេលដែលប្រព័ន្ធត្រូវបានបញ្ចប់ទាំងស្រុង។

## ៦.៣ ផែនការអនុវត្តតាមវដ្ត (Sprint Planning)

ដំណើរការអភិវឌ្ឍន៍នឹងត្រូវបានបែងចែកជា ៤ វដ្ត (Sprints) សំខាន់ៗ ដែលវដ្តនីមួយៗមានរយៈពេលពី ២ ទៅ ៣ សប្តាហ៍៖

វដ្ត (Sprint)	រយៈពេល	សកម្មភាពសំខាន់ៗ	លទ្ធផល (Deliverables)
Sprint 0: Planning & Design	សប្តាហ៍ទី ១	សិក្សាតម្រូវការ, រចនា ERD, និងរៀនពីបច្ចេកវិទ្យាថ្មី (Tailwind CSS, Flask 2.3) ។	SRS Doc, Database Schema
Sprint 1: Core Backend	សប្តាហ៍ទី ២-៤	ដំឡើង Flask Env, តភ្ជាប់ MySQL ជាមួយ SQLAlchemy, និងបង្កើត Login/Session ។	Functional Backend, User Auth
Sprint 2: Data & Rules	សប្តាហ៍ទី ៥-៧	បង្កើតទម្រង់បញ្ចូលទិន្នន័យ (WTForms) និងសរសេរកូដ Inference Engine (Python Logic) ។	Product CRUD, Basic Rec. Engine

<b>Sprint 3: UI &amp; Integration</b>	សប្តាហ៍ទី ៨-១០	រចនា UI ជាមួយ Tailwind CSS និង Jinja2, តភ្ជាប់ Frontend ទៅ Backend។	Complete UI, Responsive Pages
<b>Sprint 4: Testing &amp; Manual</b>	សប្តាហ៍ទី ១១-១២	សរសេរតេស្តដោយប្រើ Pytest, សរសេរឯកសារ Manual Docs, និងកែសម្រួលកំហុស។	Final System, User Manual

## ៧. បច្ចេកវិទ្យា និងឧបករណ៍ (Technology Stack)

ខាងក្រោមនេះជាតារាងលម្អិតនៃបច្ចេកវិទ្យា និងឧបករណ៍ដែលបានជ្រើសរើសសម្រាប់ការអភិវឌ្ឍប្រព័ន្ធជំនាញ ណែនាំស្ថាប័ន និងកុំព្យូទ័រ ដោយបញ្ជាក់ពីមូលហេតុបច្ចេកទេសច្បាស់លាស់ ៖

ផ្នែក (Component)	ជម្រើស/ Technology	មូលហេតុជ្រើសរើស (Rationale)
Frontend	HTML5 / Jinja2 Templates	ប្រើប្រាស់ Jinja2 ដើម្បីបង្ហាញទិន្នន័យ (Server-side Rendering) ពី Flask មកកាន់ HTML ដោយផ្ទាល់ និងមានសុវត្ថិភាព។
	Tailwind CSS 3	យើងប្រើប្រាស់ Tailwind CSS ជា Utility-First Framework ដែលអនុញ្ញាតឱ្យរចនា UI បានលឿន និងមានភាពបត់បែនខ្ពស់ (Custom Design)។
	Vanilla JS	ប្រើប្រាស់ JavaScript ធម្មតា (គ្មាន Framework ធ្ងន់ៗ) សម្រាប់ការធ្វើ Interactivity លើទំព័រវេបសាយ ដើម្បីរក្សាល្បឿនប្រព័ន្ធឱ្យលឿន។
Backend / Logic	Python 3.12	ភាសាស្ត្រូលសម្រាប់ការសរសេរ Inference Engine និងតក្កវិជ្ជា។
	Flask 2.3	Framework ស្ត្រូលសម្រាប់បង្កើត Web Application ដែលមានលក្ខណៈស្រាល និងងាយស្រួលគ្រប់គ្រង Routes។
	WTForms 3.1.2	បណ្តាញសម្រាប់បង្កើត និងគ្រប់គ្រង Forms ព្រមទាំងធ្វើ Validation ទិន្នន័យពីអ្នកប្រើប្រាស់បានយ៉ាងមានប្រសិទ្ធភាព។

	SQLAlchemy 2.0.45	(Version as requested) ORM សម្រាប់គ្រប់គ្រងការតភ្ជាប់ និងប្រតិបត្តិការជាមួយ MySQL Database ដោយមិនបាច់សរសេរ Raw SQL ច្រើន។
	Sessions	ប្រើប្រាស់ Flask Sessions (Server-side) ដើម្បីរក្សាទុកស្ថានភាពរបស់អ្នកប្រើប្រាស់ (State Management) អំឡុងពេលឆ្លើយសំណួរ។
Database	MySQL 8.0	ប្រព័ន្ធគ្រប់គ្រងទិន្នន័យទំនាក់ទំនង (RDBMS) សម្រាប់រក្សាទុកព័ត៌មានផលិតផល និងច្បាប់ (Rules)។
Server / Tools	Flask Dev Server	ប្រើប្រាស់ Server របស់ Flask ផ្ទាល់សម្រាប់ការអភិវឌ្ឍ និងបង្ហាញលទ្ធផល (Local Environment) ដោយមិនចាំបាច់ប្រើ Unicorn។
	Pytest	ឧបករណ៍សម្រាប់ធ្វើតេស្តកូដ (Unit Testing) ដើម្បីធានាថា Logic នៃការណែនាំដំណើរការត្រឹមត្រូវ។
	Microsoft Word	ការរៀបចំឯកសារបច្ចេកទេស និងសៀវភៅណែនាំដោយផ្ទាល់ដៃ (Manual)។
Database	MySQL 8.0	ជាប្រព័ន្ធគ្រប់គ្រងទិន្នន័យទំនាក់ទំនង (RDBMS) ដែលអនុញ្ញាតឱ្យប្រើប្រាស់ Foreign Keys ដើម្បីភ្ជាប់ទំនាក់ទំនងរវាងតារាង Products និង Brands បានយ៉ាងរឹងមាំ (MySQL Manual, 2023)។
Tools & Standards	Git & GitHub	ឧបករណ៍សម្រាប់គ្រប់គ្រងកំណែកូដ (Version Control) ដែលអនុញ្ញាតឱ្យសមាជិកក្រុមធ្វើការរួមគ្នាដោយគ្មានការបាត់បង់ទិន្នន័យ (Pro Git, 2020)។
	VS Code	កម្មវិធីសរសេរកូដ (IDE) ដែលមាន Extensions ជំនួយជាច្រើនសម្រាប់ Python និង Web Development។
	Zotero	សម្រាប់គ្រប់គ្រងឯកសារយោង និងបង្កើតបញ្ជី Reference តាមទម្រង់ IEEE យ៉ាងត្រឹមត្រូវ។

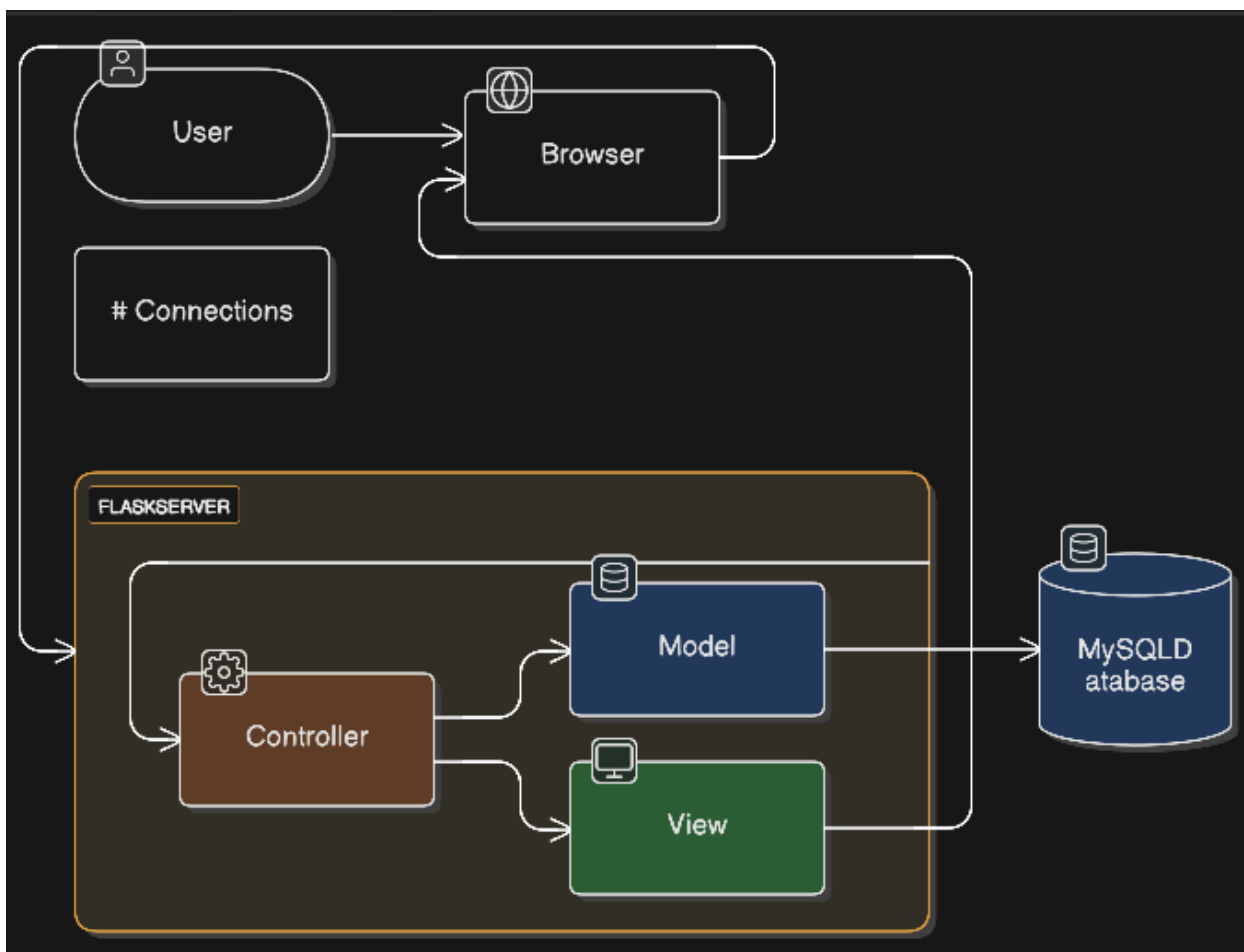
## ៨. បេសសម្ព័ន្ធប្រព័ន្ធ (System Design)

## ៨.១ Architecture Overview

ប្រព័ន្ធនេះត្រូវបានរចនាឡើងដោយប្រើប្រាស់ស្ថាបត្យកម្ម Model-View-Controller (MVC) ជាទម្រង់គំនិត (MVC-inspired Architecture) ដែលត្រូវបានអនុវត្តយ៉ាងទូលំទូលាយនៅក្នុងការអភិវឌ្ឍ Web Application ដោយប្រើ Flask Framework (Flask Docs, 2024)។ ទោះបីជា Flask មិនបានបង្ខំឱ្យប្រើ MVC ជាផ្លូវការឡើយ ក៏ដោយ ការបែងចែកតួនាទី Model, View និង Controller ជួយឱ្យប្រព័ន្ធមានរចនាសម្ព័ន្ធច្បាស់លាស់ ងាយស្រួលថែទាំ និងអភិវឌ្ឍបន្ថែម។

**Model (Database Layer):** ប្រើប្រាស់ SQLAlchemy ដើម្បីកំណត់រចនាសម្ព័ន្ធទិន្នន័យ (Table Schema) និងគ្រប់គ្រងការតភ្ជាប់ទៅកាន់ MySQL Database។

- **View (Presentation Layer):** ប្រើប្រាស់ Jinja2 Templates រួមជាមួយ Tailwind CSS 3 ដើម្បីបង្កើតចំណុចប្រទាក់អ្នកប្រើប្រាស់ (User Interface) ដែលបង្ហាញ Form សំណួរ និងលទ្ធផលណែនាំ។
- **Controller (Business Logic Layer):** ប្រើប្រាស់ Python Functions នៅក្នុង Flask Routes ដើម្បីដំណើរការ **Inference Engine** (Forward Chaining Logic) ក្នុងការផ្តល់ធាតុចូល (Input) ទៅនឹងផលិតផល។



## ៨. ២ Database Design (ERD)

យោងតាមគោលការណ៍ Relational Database (MySQL Manual, 2023), មូលដ្ឋានទិន្នន័យត្រូវបានរចនាឡើងដោយមានតារាង (Entities) និងទំនាក់ទំនង (Relations) សំខាន់ៗដូចខាងក្រោម៖



### ៨.៣ UI/UX Mockup

ចំណុចប្រទាក់ត្រូវបានរចនាឡើងដោយផ្ដោតលើភាពងាយស្រួល (Usability) និងការបង្ហាញលទ្ធផលប្រៀបធៀបដែល ច្បាស់លាស់ (NNGroup, 2021)៖





Tell us your needs. We'll find your perfect teach.

 \$300-\$2000

#### Usage Purpose



#### Brand Preference (Optional)


Select a brand






Find My Device

- **Home/Diagnosis Screen:** ជាទម្រង់ Form ដែលមានជម្រើស (Dropdown/Slider) សម្រាប់ជ្រើសរើស ថ្វីកា និង ការប្រើប្រាស់។

### Recommendations




**AlienWare**  
**\$1.799**

 8500  16GB  18 hrs




Brand & Specs

View Details

☒ Compare




**MSI KATANA 17**  
**\$1.999**

 11200  16GB  14 hrs




Explanation  
Why this fits: This device delivery the best CPU performance for your gaming needs, \$1500 budget

View Details

☒ Compare



**MSI THIN 15**  
**\$999**

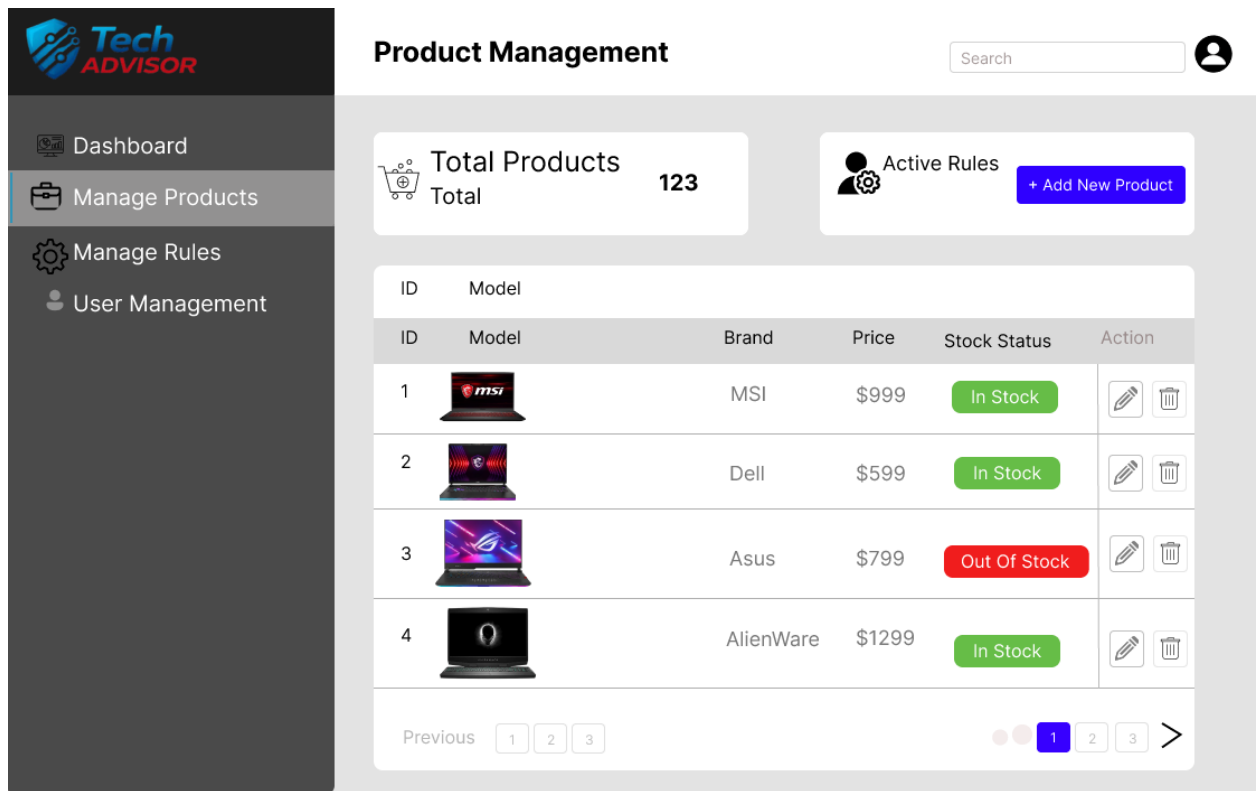
 8500  16GB  18 hrs

Brand & Specs

View Details

☒ Compare

- **Result Screen:** បង្ហាញកាតផលិតផល (Product Cards) ចំនួន ៣ ដែលល្អបំផុត ដោយមានប៊ូតុង "Compare" ។



- **Admin Dashboard:** តារាង CRUD សម្រាប់គ្រប់គ្រងផលិតផល និងមើលស្ថិតិអ្នកចូលទស្សនា។

## ៩. ផែនការអនុវត្ត និងកាលវិភាគ (Work Plan & Timeline)

ខាងក្រោមនេះជាតារាងពេលវេលាអនុវត្តគម្រោងរយៈពេល ១២ សប្តាហ៍ ដោយផ្អែកលើវិធីសាស្ត្រ Agile Sprints៖

សប្តាហ៍ (Week)	ដំណាក់កាល (Phase)	សកម្មភាពលម្អិត (Detailed Activities)	លទ្ធផល (Deliverables)
Week 1	Planning & Setup	<ul style="list-style-type: none"> <li>- សិក្សាពីតម្រូវការប្រព័ន្ធ (Requirements)</li> <li>- តម្លើង Python, Flask, MySQL, និង Git</li> <li>- បង្កើតគម្រោង Hello World ក្នុង Flask ដើម្បីតែស្ត Environment ។</li> </ul>	Project Environment Ready

Week 2	Database Design	<ul style="list-style-type: none"> <li>- រចនា ERD ចុងក្រោយសម្រាប់ Users, Brands, Products, Rules</li> <li>- សរសេរ SQL Script (DDL) ដើម្បីបង្កើតតារាងក្នុង MySQL 8.0</li> <li>- តេស្តការតភ្ជាប់ SQLAlchemy ទៅកាន់ Database ។</li> </ul>	Final ERD & Database Created
Week 3	Backend Foundation	<ul style="list-style-type: none"> <li>- បង្កើត Flask Routes សម្រាប់ទំព័រដើម និងទំព័រ Admin</li> <li>- បង្កើត WTForms សម្រាប់ Login និងការបញ្ចូលផលិតផល</li> <li>- អនុវត្តមុខងារ flask_login សម្រាប់សុវត្ថិភាព Admin ។</li> </ul>	Admin Login Functional
Week 4	CRUD Implementation	<ul style="list-style-type: none"> <li>- សរសេរមុខងារបញ្ចូលទិន្នន័យ (Create Product)</li> <li>- សរសេរមុខងារបង្ហាញបញ្ជីផលិតផល (Read Product List)</li> <li>- បញ្ចូលទិន្នន័យផលិតផលពិតចំនួន ១០ សម្រាប់តេស្ត។</li> </ul>	CRUD System Working
Week 5	Frontend Layout	<ul style="list-style-type: none"> <li>- តម្លើង Tailwind CSS 3 តាមរយៈ CDN ឬ NPM</li> <li>- បង្កើត Layout មេ (Base Template) ដោយប្រើ Jinja2</li> </ul>	Basic UI with Tailwind

		- រចនាទម្រង់ Form ឱ្យស្អាតសម្រាប់ទំព័រស្វែងរក (Search Page) ។	
Week 6	Inference Logic (1)	- ចាប់ផ្តើមសរសេរ Python Function សម្រាប់ <b>Expert System Logic</b>  - កំណត់ពិន្ទុ (Scoring System) សម្រាប់ CPU/RAM ឱ្យត្រូវនឹងការប្រើប្រាស់ (Gaming vs Office) ។	Scoring Algorithm Draft
Week 7	Inference Logic (2)	- ភ្ជាប់ Form ពី Frontend មកកាន់ Logic នៅ Backend  - សរសេរ Query ដើម្បីទាញយកផលិតផលដែលត្រូវនឹងពិន្ទុ (Match Making)  - ប្រើប្រាស់ <b>Flask Sessions</b> ដើម្បីរក្សាទុកចម្លើយអ្នកប្រើ។	Recommendation Engine Beta
Week 8	UI & Integration	- បង្ហាញលទ្ធផលលំដាប់ (Result Page) ជាកាតផលិតផលស្អាតៗ  - សរសេរ <b>Vanilla JS</b> តិចតួចសម្រាប់ការធ្វើអន្តរកម្ម (Interactions) លើទំព័រ។	Integrated UI Flow
Week 9	System Integration	ភ្ជាប់គ្រប់ Module (Auth, CRUD, Logic, UI), Optimize Query និង Code Structure	Full System (Alpha)
Week 10	Testing	Unit Testing ដោយ Pytest, Manual Testing តាម User Scenarios, Bug Fixing	Stable System

Week 11	Documentation	សរសេរ User Manual, Technical Documentation, Installation Guide	Complete Documentation
Week 12	Finalization	Code Cleanup, Final Review, រៀបចំ Final Report និង Presentation Slides	Final System & Presentation

(សម្គាល់៖ សូមភ្ជាប់ Gantt Chart នៅក្នុង Appendix D)

## ១០. លទ្ធផលរំពឹងទុក (Expected Outputs/Deliverables)

- ល.រ.១.១.១៖ ឯកសារបញ្ជាក់តម្រូវការប្រព័ន្ធ (SRS Document) ដែលមានលម្អិតពី Use Case និងតក្កវិជ្ជា។
- ល.រ.១.១.២៖ មូលដ្ឋានទិន្នន័យចំណេះដឹង (Knowledge Base) ពេញលេញក្នុង MySQL (Products & Rules Data) ។
- ល.រ.១.១.៣៖ Source Code នៃប្រព័ន្ធទាំងមូលដែលបានបង្ហាញលើ Git Repository ។
- ល.រ.១.១.៤៖ របាយការណ៍តេស្ត (Test Report) ដែលបង្ហាញពីលទ្ធផលនៃការធ្វើតេស្តភាពត្រឹមត្រូវនៃការណែនាំ។
- ល.រ.១.១.៥៖ សៀវភៅណែនាំអ្នកប្រើប្រាស់ (User Manual) សម្រាប់ Admin (របៀបបញ្ចូលទិន្នន័យ) និង User ។

## ១១. ការធានាគុណភាព និងការតេស្ត (QA & Testing)

ដើម្បីធានាថាប្រព័ន្ធដែលបានត្រឹមត្រូវ និងមានសុវត្ថិភាព ការតេស្តនឹងត្រូវធ្វើឡើងតាមស្តង់ដារ ISO 25010 (ISO, 2011)៖

- **Unit Testing:** ប្រើប្រាស់ unittest library របស់ Python ដើម្បីតេស្តមុខងារនីមួយៗដាច់ដោយឡែក (ឧទាហរណ៍៖ តេស្តមុខងារ calculate\_score()) ។
- **Integration Testing:** តេស្តការតភ្ជាប់រវាង Flask Backend និង MySQL Database ថាទិន្នន័យត្រូវបានទាញយកត្រឹមត្រូវ។
- **UAT (User Acceptance Test):** ឱ្យអ្នកប្រើប្រាស់សាកល្បងចំនួន ៥ នាក់ ធ្វើការស្វែងរកទូរស័ព្ទ និងវាយតម្លៃថា តើលទ្ធផលដែលចេញមកពិតជាត្រូវចិត្តពួកគេដែរឬទេ។

• Acceptance Criteria (AC):

- AC1: ប្រព័ន្ធត្រូវផ្តល់ការណែនាំយ៉ាងហោចណាស់ ១ ផលិតផល បើទោះបីជាលក្ខខណ្ឌតិចតួចក៏ដោយ (Fallback Logic) ។
- AC2: ពេលវេលាឆ្លើយតបត្រូវតិចជាង ២ វិនាទី។

១២. ហានិភ័យ និងយុទ្ធសាស្ត្រកាត់បន្ថយ (Risks & Mitigation)

ហានិភ័យ (Risk)	ផលប៉ះពាល់	កម្រិត	ការកាត់បន្ថយ (Mitigation Strategy)
Outdated Data (តម្លៃទូរស័ព្ទប្រែប្រួល)	មធ្យម (Medium)	Medium	បង្កើតមុខងារ Admin Panel ដែលងាយស្រួលកែប្រែតម្លៃបានរហ័ស (Real-time Update) ។
Logic Errors (ការណែនាំខុស)	ខ្ពស់ (High)	High	ធ្វើ Manual Validation លើ Rules ជាមួយអ្នកជំនាញ IT មុនពេលបញ្ចូលក្នុងកូដ។
Data Loss	ខ្ពស់ (High)	High	ធ្វើការ Backup Database ជាប្រចាំសប្តាហ៍ (Weekly SQL Dump) និងប្រើ Git ។

១៣. តួនាទីក្រុម (Team Roles & Responsibilities)

សមាជិក	តួនាទី	ភារកិច្ចសំខាន់
សេង សុខរាង	Project Manager (PM)/ QA/Tester	ដឹកនាំគម្រោង, សរសេររបាយការណ៍, និងសម្របសម្រួលក្រុម។ បង្កើត Test Cases, រកកំហុស (Bugs), និងសរសេរ User Manual ។
លីម រតនា	Backend Developer	អភិវឌ្ឍន៍ Inference Engine (Python/Flask) និង Database (MySQL) ។
ជ័យ ថ័នីសុផ្ញ៌	Frontend Developer	រចនា UI (Tailwind/Jinja2) និងធានាថាវា Responsive ។

## ១៤. ថវិកា (Budget) (ប្រសិនបើត្រូវការ)

មិនតម្រូវឱ្យមាន (Not Applicable): ដោយសារគម្រោងនេះដំណើរការលើម៉ាស៊ីនផ្ទាល់ (Localhost) និងប្រើប្រាស់ឧបករណ៍ Open Source ឥតគិតថ្លៃ ដូចជា Flask, MySQL, និង Tailwind CSS ហេតុនេះមិនមានតម្រូវការចំណាយថវិកាលើ Hosting ឬ Domain Name ទេ។

## ១៥. សេចក្តីសន្និដ្ឋាន (Conclusion)

សរុបសេចក្តីមក គម្រោង "ប្រព័ន្ធជំនាញណែនាំការជ្រើសរើសស្ថាពរ និងកុំព្យូទ័រ" នេះនឹងដើរតួនាទីយ៉ាងសំខាន់ក្នុងការដោះស្រាយបញ្ហា "ភាពស្មុគស្មាញនៃជម្រើស" សម្រាប់អ្នកប្រើប្រាស់ទូទៅ។ ដោយប្រើប្រាស់បច្ចេកវិទ្យា Flask និង Expert System Logic, យើងរំពឹងថានឹងបង្កើតបាននូវដំណោះស្រាយដែលមានប្រសិទ្ធភាព ចំណាយតិច និងងាយស្រួលប្រើ។ ជោគជ័យនៃគម្រោងនេះមិនត្រឹមតែជួយសម្រួលដល់ការសម្រេចចិត្តទិញប៉ុណ្ណោះទេ ប៉ុន្តែថែមទាំងជាជំហានដំបូងដ៏សំខាន់សម្រាប់ក្រុមនិស្សិតក្នុងការឈានទៅចាប់យកជំនាញ Artificial Intelligence និង Web Development ក្នុងកម្រិតខ្ពស់បន្តទៀត។

## ឯកសារយោង (References)

No.	Site / Link (Download Source)	Translate (KH)	Content (Micro-Detail / Main Idea)	Page No	Note (For Proposal Section)
1	[PDF] Rule-Based Expert System for Electronic Product Recommendation(2021)	ប្រព័ន្ធអ្នកជំនាញផ្អែកលើច្បាប់សម្រាប់ការណែនាំផលិតផលអេឡិចត្រូនិក	Expert System / Recommendation: Methodology for using rule-based logic to suggest electronic products.	Page 1	Expert System & Recommendation Engine
2	[PDF] Optimizing Smartphone	ការបង្កើនប្រសិទ្ធភាព	Smartphone Recommendation:	Page 1	Problem Domain & Solution

	Recommendation System through Adaptation of Genetic Algorithm (2021)	ប្រព័ន្ធនែនាំទូរស័ព្ទឆ្លាតវៃ	Applying genetic algorithms to Progressive Web Applications.		
3	[PDF] Expert System Design for Laptop Recommendation Using Forward Chaining (2021)	ការរចនាប្រព័ន្ធអ្នកជំនាញសម្រាប់នែនាំកុំព្យូទ័រយួរដៃដោយប្រើ Forward Chaining	Forward Chaining: Design and implementation of an inference engine for laptop selection.	Page 1	Inference Engine
4	[PDF] Rule-Based Recommendation Systems: A Systematic Review (2021)	ប្រព័ន្ធនែនាំផ្អែកលើច្បាប់៖ ការពិនិត្យឡើងវិញជាប្រព័ន្ធ	Rule-Based Filtering: Comprehensive review of current filtering techniques and architectures.	Page 1	Literature Review
5	[PDF] Knowledge-Based Recommendation Systems: State of the Art (2022)	ប្រព័ន្ធនែនាំផ្អែកលើចំណេះដឹង	Expert Systems: Overview of modern knowledge-based recommendation technologies.	Page 26 Page 30 Page 54	Related Works



6	[Book] Mastering Flask Web and API Development (2024)	ការគ្រប់គ្រងការអភិវឌ្ឍ Flask Web និង API	Flask Backend: Detailed technical guide on building web backends and APIs using Flask.	Page 3 Page 26 Page 35	Backend Implementation
7	[PDF] MySQL 8 Reference Manual (2023)	សៀវភៅណែនាំ MySQL 8	Database Server: Official technical specifications and management for MySQL 8 databases.	Page 10	Database Design
8	[PDF] Foundations of Computational Agents (Python AI) (2021)	មូលដ្ឋានគ្រឹះនៃភ្នាក់ងារគណនា	Expert System Theory: Theoretical framework for AI agents and expert systems using Python.	Page 109 Page 381	Expert System Theory
9	[PDF] The Scrum Guide (2020)	មគ្គុទ្ទេសក៍ Scrum	Agile / Scrum: The official standard for project management and team collaboration.	Page 3 Page 5-6	Methodology

10	[Book] Learning Path Learn Web Development with Python (2020)	មាតិកាសៀវភៅ អភិវឌ្ឍន៍វិប ជាមួយ Python	Python Web Development: Educational path for building web applications with Python.	Page 10  Page 16  Page 17	Backend
11	[Book] #1 Book for Python Programming to AI and Automation (2025)	សៀវភៅ Python សម្រាប់ការ សរសេរកម្ម វិធី AI និង ស្វ័យប្រវត្តិ កម្ម	Python & AI: Comprehensive guide from basic programming to advanced AI automation.	Page 2  Page 3	Programming & AI
12	[PDF] AI Rule-Based Expert System: Diagnosis of Bean Diseases (2025)	ប្រព័ន្ធអ្នក ជំនាញ AI៖ ការវិនិច្ឆ័យ ជំងឺសណ្តែក	AI Rule-Based Expert System: Practical application of rules for diagnosis and treatment.	Page 1  Page 2	AI  Implementation
13	[PDF] Architecting an AI-Driven Decision Support System (2025)	ការរចនា ប្រព័ន្ធគាំទ្រ ការសម្រេច ចិត្តដែល ដឹកនាំដោយ AI	Decision Support System: Technical review of online learning and assessment architectures.	Page 1  Page 3	System  Architecture

14	[PDF] Explainability-Driven Quality Assessment for Rule-Based Systems (2025)	ការវាយតម្លៃគុណភាពសម្រាប់ប្រព័ន្ធផ្អែកលើច្បាប់	Quality Assessment: Framework for assessing the reliability and explainability of rules.	Page 1	QA / Testing
----	--	---	--	--------	--------------

### ឧបសម្ព័ន្ធ (Appendices)

- **Appendix A:** Use Case Diagram (Admin vs User)
- **Appendix B:** ERD (Entity Relationship Diagram)
- **Appendix C:** UI Mockups (Login, Search, Result Pages)
- **Appendix D:** Gantt Chart (Project Timeline)
- **Appendix E:** Sample Test Cases