**Форма № Н-9.02**

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально-науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проєкту (роботи)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_магістра\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(освітньо-кваліфікаційний рівень)

на тему

Створення експертної системи для автомобільних двигунів автомобілів лінійки Mitsubishi.

Виконав: студент \_6\_\_ курсу, групи 601 ТН

спеціальності

\_\_122 Комп’ютерні науки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Пєвнєв Д.П.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_Скакаліна О.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2022 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**Спеціальність 122 «Комп’ютерні науки»**

**на тему**

**«Створення експертної системи для автомобільних двигунів автомобілів лінійки Mitsubishi.»**

**Студента групи 601 - ТН Пєвнєва Дмитра Павловича**

Керівник роботи

доцент

Скакаліна О.В.

Завідувач кафедри

кандидат технічних наук,

доцент

Полтава – 2022

**РЕФЕРАТ**

Кваліфікаційна робота магістра: 84 с., 40 рисунків, 1 додаток, 23 джерела.

**Об’єкт дослідження:** експертні системи.

**Мета роботи:** Створення спеціалізованої експертної системи діагностики несправностей двигунів у лінійці автомобілів марки «Mitsubishi»

**Методи:** Об’єктно-орієнтоване проєктування, проєктування баз даних, методологія розробки MVC.

**Ключові слова:** веб-сайт, інформаційна система, MVC, експертна система.

**ABSTRACT**

Master`s qualification work: 84 pages, 40 drawings, 1 appendix, 23 sources.

**Object of research**: expert system.

**Purpose:** Creation of a specialized expert system for diagnosing engine malfunctions in the range of "Mitsubishi" cars.

**Methods**: Object-oriented design, database design, MVC development methodology.

**Keywords:** website, information system, MVC, expert system.

**ЗМІСТ**

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ 9](#_Toc121559841)

[1.1 Вступ до експертних систем 9](#_Toc121559842)

[1.2 Переваги та недоліки експертних систем 9](#_Toc121559843)

[1.3 Приклади рішень 10](#_Toc121559844)

[1.4 Постановка задачі дипломної роботи 14](#_Toc121559845)

[РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ 15](#_Toc121559846)

[2.1 Історія експертних систем 15](#_Toc121559847)

[2.2 Компоненти та принцип роботи експертної системи 20](#_Toc121559848)

[2.3 Класифікація експертних систем 22](#_Toc121559849)

[2.4 Життєвий цикл розроблення програмного забезпечення 27](#_Toc121559850)

[2.5 Принцип роботи веб-сайту 30](#_Toc121559851)

[РОЗДІЛ 3 ПРОЦЕС РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-КОНФЕРЕНЦІЙ 34](#_Toc121559852)

[3.1 Вибір та обґрунтування веб-технологій для розроблення програмного забезпечення 34](#_Toc121559853)

[3.1.1 Обґрунтування вибору використання мови PHP.. 34](#_Toc121559854)

[3.1.2 Обґрунтування вибору використання MySQL. 37](#_Toc121559855)

[3.1.3 Обґрунтування вибору використання XAMPP. 38](#_Toc121559856)

[3.1.4 Обґрунтування вибору використання Codeigniter. 39](#_Toc121559857)

[3.2 Проєктування інформаційної системи 40](#_Toc121559858)

[3.3 Розробка структури сайту та макетів 45](#_Toc121559859)

[3.4 Розробка бази даних 48](#_Toc121559860)

[3.5 Реалізація програмного продукту 52](#_Toc121559861)

[ВИСНОВКИ 62](#_Toc121559862)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 63](#_Toc121559863)

[ДОДАТОК А ВИХІДНИЙ КОД ОСНОВНИХ КОМПОНЕНТІВ 65](#_Toc121559864)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**SDLC –** Software development lifecycle.

**UML –** Unified Modeling Language.

**HTTP –** HyperText Transfer Protocol.

**ПК –** Персональний комп’ютер.

**URL –** Uniform Resource Locator.

**DNS –** Domain Name System.

**PHP –** Мова програмування.

**HTML –** HyperText Markup Language.

**CSS –** Cascading Style Sheets.

**MySQL –** Вільна система керування реляційними базами даних.

**XAMPP –** Багатоплатформова збірка веб-сервера.

**СУБД –** Система управління баз даних.

**БД –** База даних.

**MVC –** Model-view-controller.

**W3С –** World Wide Web Consortium.

**CRM** – Customer Relationship Management.

**CMS –** Content Management System.

**ВСТУП**

Обчислювальна техніка на даному етапі завоювала ключові позиції в багатьох сферах людської діяльності. Крім класичних застосувань, пов’язаних з виконанням інженерних і економічних розрахунків, розробкою автоматизованих систем керування, створенням інформаційно-пошукових систем і т.д., зараз успішно розвивається науковий напрямок, що має назву «штучний інтелект». Глобальною метою вказаного наукового напрямку є створення інтелектуальних автоматизованих систем, що виконують ті ж функції, що й творча особистість, в крайньому разі, їх найпростіші прояви.

За останні роки із загального напрямку «штучний інтелект» виділився новий науковий напрямок, пов’язаний зі створенням експертних систем, призначених для вирішення задач експертного оцінювання ситуацій у різних предметних областях. При побудові експертних систем основними є питання: „Які знання повинні бути в них подані та у якій формі?” Структура знань залежить від сфери їх використання і може мати досить складний характер. Така структура містить у собі різні факти із предметної області, взаємозв’язків між ними, правил дій і т.д. Вона також повинна містити в собі знання, що стосуються способу включення знань до експертної системи. Складність і різноманітність структур знань спонукали до розробки різноманітних способів подання знань, з яких варто виділити як найбільш поширені: логічні моделі, продукційні та фреймові системи, семантичні мережі, модель дошки оголошень. Кожний спосіб подання має свої переваги та недоліки й тяжіє до певної структури знань. В останні роки все більше почали використовуватися моделі подання знань, що поєднують зазначені способи.

Особливість систем подання знань полягає в тому, що вони моделюють діяльність людини, яка часто здійснюється у неформальному вигляді. Якщо методи вирішення обчислювальних задач основуються на чітких алгоритмах, обґрунтованість яких базується на понятті збіжності, то моделі подання знань мають справу з інформацією, що одержується від експертів, яка часто має якісний, і до того ж, суперечливий характер. Проте, у силу специфіки функціонування комп’ютерних систем, подібна інформація повинна бути приведена до однозначного нормалізованого вигляду. Це здійснюється на основі ідей багатозначної логіки, теорії нечітких множин і аналогічних математичних моделей. У той же час моделі подання знань є предметом досліджень і розробок у середовищі вузькоспеціалізованих фахівців, в основному програмістів і математиків, у той час як потреба в таких моделях відчувається практично у всіх предметних областях. Це пояснюється необхідністю побудови експертних систем, структура яких нерозривно пов’язана з формами подання знань, обумовлених у свою чергу особливостями предметної області.

# РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1. **Вступ до експертних систем**

Expert System - це комп'ютерна програма, яка використовує технології штучного інтелекту (AI) для імітації судження та поведінки людини чи організації, яка має досвід у певній галузі.

Експертні системи, як правило, призначені для доповнення, а не заміни, експертів людини.

Сучасні експертні системи знань використовують машинне навчання та штучний інтелект для імітації поведінки чи судження експертів домену. Ці системи можуть з часом покращити свою продуктивність, оскільки вони набирають більше досвіду, як і люди.

1. **Переваги та недоліки експертних систем**

Експертні системи найбільш поширені в області складних проблем і вважаються широко використовуваними альтернативами в пошуку рішень, які вимагають існування певних людських знань. Експертна система також може обґрунтовувати надані нею рішення на основі знань і даних попередніх користувачів. Зазвичай експертні системи використовуються для прийняття бізнес-маркетингових стратегічних рішень, аналізу продуктивності систем реального часу, конфігурації комп’ютерів і виконання багатьох інших функцій, які зазвичай потребують людського досвіду.

Переваги експертної системи:

* ЕС може надати послідовні відповіді на повторювані рішення, процеси та завдання. Поки база правил у системі залишається незмінною, незалежно від того, скільки разів подібні проблеми тестуються, зроблені висновки залишаться незмінними;
* якщо є сумніви щодо вирішення певної проблеми; він запропонує користувачам відповісти на кілька запитань, щоб зробити логічний висновок;
* ЕС не має людських обмежень і може працювати цілодобово безперервно. Користувачі зможуть часто використовувати його для пошуку рішень. Знання експертів є безцінним активом для компанії. Він може зберігати знання та використовувати їх до тих пір, поки це буде потрібно організації;
* на відміну від людей, які часто мають проблеми з адаптацією до нового середовища, експертна система має високу адаптивність і може відповідати новим вимогам за короткий період. Він також може отримувати нові знання від експерта та використовувати їх як правила висновку для вирішення нових проблем.

Недоліки експертних систем:

* ЕС не може дати творчі та інноваційні відповіді, як люди-експерти в незвичайних обставинах;
* впровадження експертної системи в бізнес буде фінансовим тягарем для невеликих організацій, оскільки вона потребує високих витрат на розробку, а також подальших періодичних витрат на оновлення системи для адаптації до нового середовища;
* експерти в галузі не завжди зможуть пояснити свою логіку та міркування, необхідні для процесу розробки знань. Отже, завдання кодифікації знань є дуже складним і може вимагати багато часу;
* він не безпомилковий. Під час обробки може виникнути помилка через деякі логічні помилки, допущені в базі знань, що надасть неправильні рішення.

1. **Приклади рішень**

Прикладом експертної системи для діагностики автомобілів слугує програмне забезпечення ODIS.

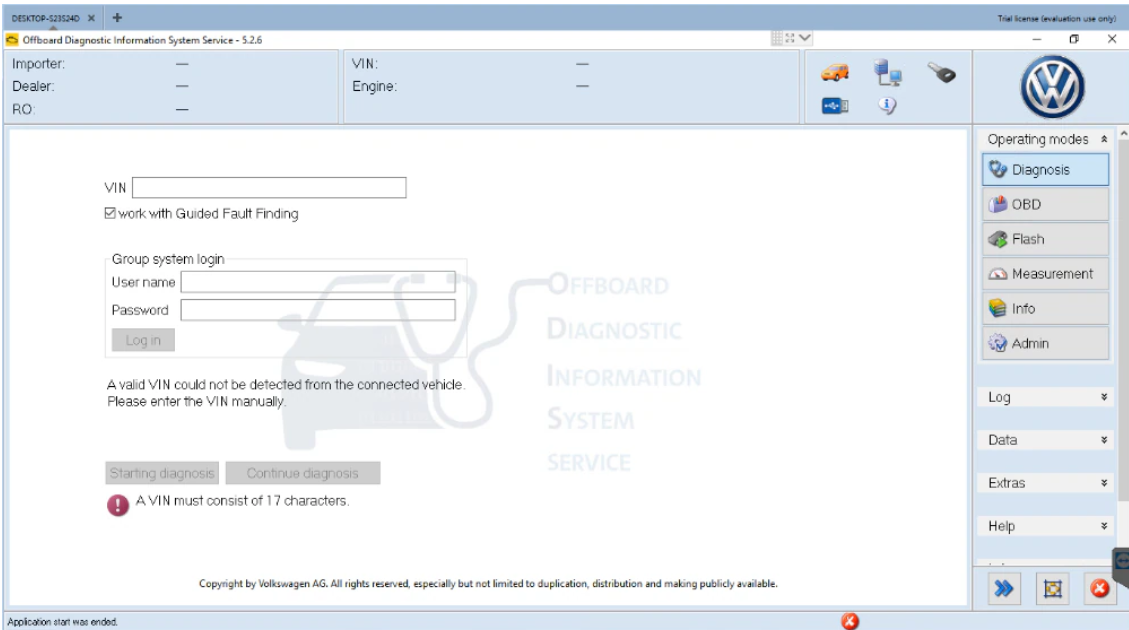


Рисунок 1.1 – Вікно програмного пакету ODIS

ODIS — це система заводської діагностики, яка забезпечує повне покриття для всіх автомобілів групи VW, що продаються в Північній Америці, включаючи нові електромобілі з технологічними пакетами ADAS.

Пакет професійної бортової діагностичної та інформаційної системи (ODIS) поставляється з попередньо завантаженим діагностичним програмуванням для автомобілів Volkswagen Audi з додатковими оновленнями ліцензії для автомобілів Lamborghini та Bently.

Прикладом реалізації конструктора експертної системи є «Exsys Corvid Expert System Development Tool»

Exsys Corvid — це потужний і широко перевірений інструмент для створення та розміщення інтерактивних програм експертної системи в Інтернеті.  
Corvid є результатом понад 28 років роботи з підприємствами та організаціями над створенням інструменту, який базується на тому, що потрібно розробникам для створення реальних систем. Він використовує дуже прагматичний підхід до розробки системи – полегшує розробникам створення та впровадження систем якомога швидше. Використовуючи онлайн-посібники, більшість користувачів можуть розпочати створення невеликих систем за кілька годин. Незважаючи на його легкість в обігу, Corvid використовувався для створення дуже складних систем.

Corvid вирішує 3 основні проблеми розробки експертної системи:

* повністю фіксуючи логіку та процес прийняття рішень експертом із домену;
* обертання системи в інтерфейс користувача з бажаним зовнішнім виглядом;
* інтеграція з іншими ІТ-ресурсами.

Corvid надає кілька способів опису логіки, тому можна використовувати відповідний підхід до проблеми. Corvid використовує "евристичні" правила If/Then на основі змінних. Є 7 типів змінних від досить стандартних числових і рядкових змінних до змінних колекції для динамічних звітів або змінних достовірності, які полегшують побудову імовірнісних систем. Змінні мають асоційовані методи та властивості, які дозволяють використовувати їх різними способами.

Правила в системі — це лише правила If/Then, які використовують англійську (або будь-яку іншу мову, яка є кращою) та алгебру. Правила легко читати, і вони, по суті, такі ж, як експерт використовував би, щоб пояснити, чому вони прийняли рішення. Зазвичай кожне правило означає невеликий крок у прийнятті рішення. Деякі правила можуть представляти логіку вищого рівня, інші можуть охоплювати проміжні кроки та використовуватися для отримання інформації, яка використовується правилами вищого рівня.

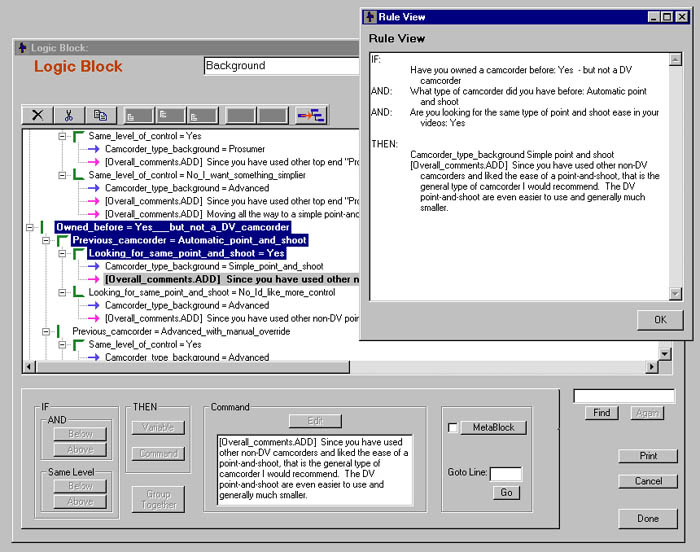


Рисунок 1.1 – Логічний блок генератора експертних систем Exsys Corvid

1. **Постановка задачі дипломної роботи**

У кваліфікаційній роботі магістра необхідно спроєктувати та програмно реалізувати експертну систему діагностики несправностей двигунів у лінійці автомобілів марки «Mitsubishi».

Експертна система складається із наступних завдань:

1. Розробка користувацького інтерфейсу для експертної системи.
2. Розробка адміністративного інтерфейсу.
3. Створення логічного блоку експертної системи.
4. Реалізація реєстрації та авторизації.
5. Створення бази даних яка повинна налічувати наступні компоненти:

* користувачі та їх ролі;
* блоки ремонту автомобілів;
* моделі двигунів;
* помилки.

# РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ

1. **Історія експертних систем**

Офіційно, історія експертних систем починається з 1965 року. Саме тоді ця технологія була офіційно представлена ​​Стенфордським прое=єктом евристичного програмування. Едвард Фейгенбаум – «батько експертних систем» – очолив перший проєкт [7].

Едвард Фейгенбаум брав участь як у MYCIN, так і в Dendral — двох окремих ранніх експертних системах.



Рисунок 2.1 – Едвард Фейгенбаум приймає участь у розробці MYCIN

Dendral була експертною системою, яка спеціалізувалася на аналізі та ідентифікації хімічних сполук. Її широко вважають першою експертною системою.

В DENDRAL використовувалися методи побудови і перевірки. Вихідні дані введені в систему з мас-спектрометром, який бомбардував дослідний зразок потоку електронів. При цьому структура зразка зазнавала зміни, переміщення атомів у структуру зразка оцінювалося системою як від’єднання вузла одного підграфа та приєднання його до іншого. Проблема складалася в тому, що для молекули може бути досить багато варіантів поділу на фрагменти, так як при бомбардуванні зразка можуть розриватися різні зв'язки в молекулі і переміщатися різні її фрагменти. У цьому зв'язку має сенс говорити тільки про ймовірність розриву певного зв'язку, але не можна припускати, як саме розділити молекулу на фрагменти[10].

В рамках проєкту DENDRAL була розроблена програма CONGEN, яка формувала опис повної хімічної структури, маніпулюючи символами, що представляють атоми і молекули. У якості вхідної інформації ця програма отримувала формулу молекул і набір обмежених, накладених на можливі взаємні зв'язки між атомами. Система використовує спеціальний алгоритм, розроблений Ледербергом, для систематичного перечислення всіх можливих молекулярних структур, а потім застосовує знання з хімії для скорочення цього списку до осяжного розміру. Знання в DENDRAL представлені у вигляді процедур для створення молекулярних структур і у вигляді правил для керованих даних і оцінюваної частини програми. Система реалізована на мові INTERLISP.

Результатом виконання програми є список усіх можливих комбінацій атомів у структурі молекули з урахуванням заданих обмежень.

MYCIN був отриманий від Dendral. Це була ще одна експертна система — та, яка зосереджена на ідентифікації бактерій, що спричиняють інфекції, та рекомендації антибіотиків.

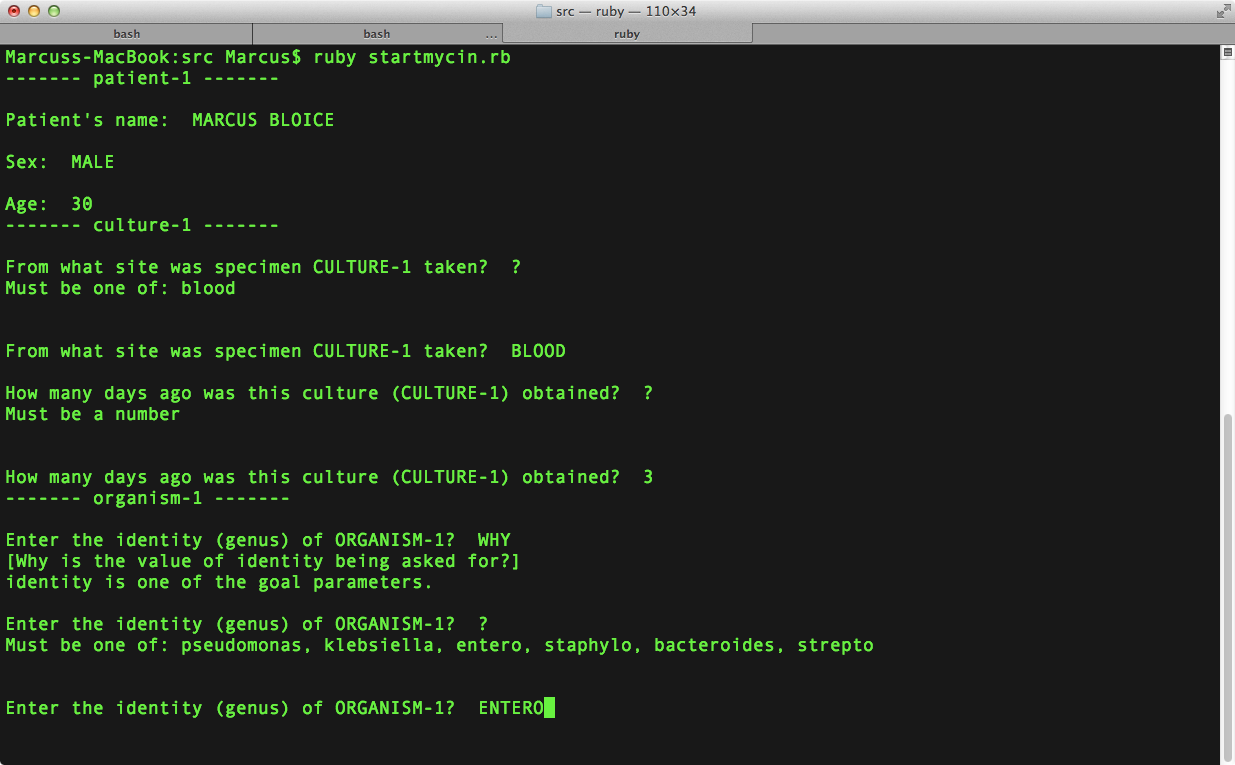


Рисунок 2.2 – Інтерфейс проограми MYCIN

Розробка даної експертної системи відбувалася в Стенфордський університет; особливо Е.Х. Важливу роль у його розвитку відіграв Shortliffe. Система MYCIN здатна допомогти терапевтам у діагностиці та лікуванні ряду захворювань інфекційних захворювань, зокрема менінгіту та бактеріального септицемії. Коли пацієнт має ознаки такого інфекційного захворювання, як правило, проводиться посів крові та сечі щоб визначити вид бактерії, що викликає інфекцію. Зазвичай це займає від 24 до 48 години до оприлюднення результатів лабораторних досліджень. У разі перерахованих вище інфекційних захворювання, проте лікар повинен буде почати лікування до того, як ці результати будуть доступні, оскільки інакше хвороба може прогресувати і фактично спричинити смерть пацієнта.

Враховуючи дані пацієнта, які доступні системі, але які можуть бути неповними та неточними, MYCIN дає проміжну інформацію про організми, які, найімовірніше, є причиною інфекція. Враховуючи це показання, MYCIN рекомендує введення ряду препаратів які повинні контролювати хворобу шляхом пригнічення зазначених організмів. Взаємодія між собою призначених ліків з ліками, які пацієнт вже приймає, можливість враховувати токсичні лікарські реакції тощо. Крім того, MYCIN може коментувати встановлений діагноз та призначення препаратів. Система MYCIN явно залишила свій відбиток на експертних системах, які були розроблені з тих пір. Навіть зараз , ця експертна система та її похідні є джерелами ідей щодо представлення та маніпулювання медичними знаннями. Система MYCIN також дала важливий імпульс до розробки подібних експертних систем у галузях, відмінних від медицини. Розробка системи INTERNIST-1 також почалася на початку 70-х років. Система все ще розробляється H.E. Попл і Дж. Д. Майерс в Пітсбурзькому університеті. Пізніше в своїх дослідженнях Попл і Майерс перейменували систему на CADUCEUS.

Метою проєкту INTERNIST/CADUCEUS є вивчення моделей для діагностики захворювання у внутрішній медицині. У внутрішній медицині налічується кілька сотень різних захворювань. Лікар-терапевт повинен не тільки мати на увазі всю клінічну картину цих захворювань під час діагностичного процесу, але також має враховувати можливі комбінації симптомів і ознак, які можуть бути спричинені взаємодією кількох захворювань, наявних у пацієнта одночасно. Кількість захворювань у внутрішній медицині та можливі поєднання клінічних ознак і симптомів, настільки велика, що їх неможливо розглянути один за одним. Тому INTERNIST/CADUCEUS зосереджується на тих захворюваннях, які є швидше за все, враховуючи симптоми, клінічні ознаки та отримані результати лабораторних досліджень від пацієнта[14].



Рисунок 2.3 – Інтерфейс INTERNIST

Ці системи не намагалися бути загальним інтелектом. Вони не вирішували проблеми. Швидше, вони зосереджувалися на обмеженому (але глибокому) фундаменті знань. І це зробило їх однією з перших успішних спроб програмного забезпечення ШІ. Тобто машини, які з’явилися для аналізу та «думання».

SUMEX, комп’ютер, розроблений для заохочення застосування штучного інтелекту в медицині. Громадський доступ через Національну медичну бібліотеку.

Розквіт експертних систем припав на 80-ті роки. За цей час дві третини компаній зі списку Fortune 500 використовували експертні системи.

Інтерес до експертних систем був міжнародним. Вони побачили збільшення фінансування досліджень у Європі та Проєкт комп’ютерних систем п’ятого покоління в Японії, у якому дослідники зосередилися (частково) на технології висновків і базах знань[2].

Машина Symbolics Lisp: рання платформа для експертних систем.

Але експертні системи не обійшлися без проблем. Були труднощі з керуванням і підтримкою бази знань. Були труднощі з написанням правил, які відображали знання експертів. І ажіотаж навколо експертних систем зростав швидше, ніж технологія могла встигати.

У 1990-х роках і далі історія експертних систем включає в себе падіння популярності та ажіотажу технології.

Очевидний занепад експертних систем у цей час має дві причини.

По-перше, просто експертні системи не виправдали ажіотажу. Вони не змогли виконати обіцяну надмірну функціональність. Вони не розширилися до більш загальної форми штучного інтелекту досить швидко, тому їх було відкинуто.

Інше пояснення полягає в тому, що вони були поглинені іншими технологічними інструментами. Оскільки експертні системи ставали все більш відомими, програмісти та розробники могли використовувати технологію, що стоїть за ними, як частину інших пропозицій.

1. **Компоненти та принцип роботи експертної системи**

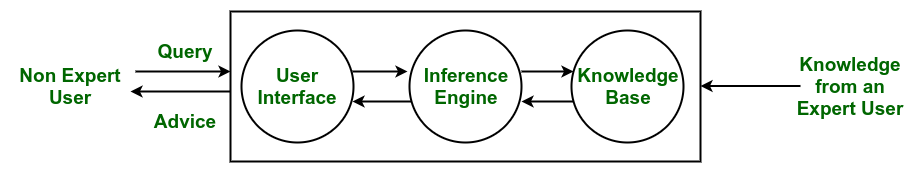


Рисунок 2.4 – Архітектура експертної системи

Архітектура експертної системи(рис. 1.1):

* база знань. База знань представляє факти та правила. Вона складається із знань у певній області, а також правил для вирішення проблеми, процедур і внутрішніх даних, що мають відношення до області;
* механізм Висновків. Функція механізму логічного висновку полягає в тому, щоб отримати відповідні знання з бази знань, інтерпретувати їх і знайти рішення, що стосується проблеми користувача. Механізм логічного висновку отримує правила зі своєї бази знань і застосовує їх до відомих фактів, щоб вивести нові факти. Механізми висновків також можуть містити пояснення та можливості налагодження;
* модуль отримання знань і навчання Функція цього компонента полягає в тому, щоб дозволити експертній системі отримувати все більше і більше знань з різних джерел і зберігати їх у базі знань;
* цей модуль дає можливість не експертному користувачу взаємодіяти з експертною системою та знаходити рішення проблеми.

Основні методи експертної системи:

* Forward Chaining. Це стратегія експертної системи, щоб відповісти на запитання: «Що може статися далі?». Тут механізм логічного висновку слідкує за ланцюжком умов і виводів і, нарешті, виводить результат. Він розглядає всі факти та правила та сортує їх, перш ніж прийти до рішення. Ця стратегія дотримується для роботи над висновком, результатом або ефектом. Наприклад, прогнозування статусу ринку акцій як ефект змін процентних ставок;

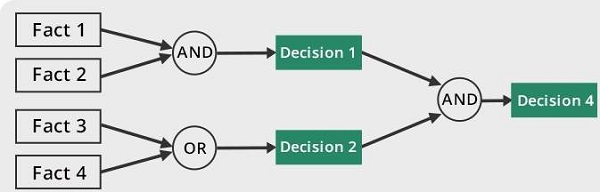


Рисунок 2.5 – Схема логіки методу Forward Chaining

* Backward Chaining. За допомогою цієї стратегії експертна система знаходить відповідь на запитання «Чому це сталося?» На основі того, що вже сталося, механізм логічного висновку намагається з’ясувати, які умови могли мати місце в минулому для отримання цього результату. Ця стратегія використовується для з’ясування причини або причини. Наприклад, діагностика раку крові у людини;

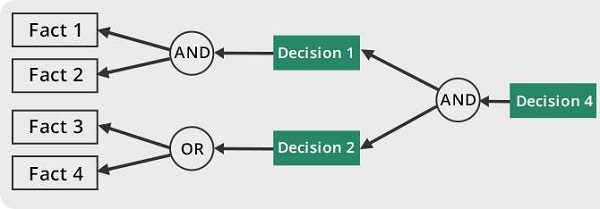


Рисунок 2.6 – Схема логіки методу Forward Chaining

1. **Класифікація експертних систем**

За принципом роботи експертні систему можна розділити на чотири класи:

* експертна система на основі правил;
* експертна система на основі фреймів;
* експертна система на основі нечіткої логіки;
* експертна система на основі експертної системи.
* **експертна система на основі правил.** Експертна система на основі правил є найпершим типом експертних систем і найпоширенішим способом побудови експертної системи для дослідників. Експертна система, доповнена правилами виробництва, була інтегрована в різні дослідницькі області, щоб допомогти дослідникам вирішувати всілякі проблеми з попередніми вхідними знаннями. Структура експертної системи зазвичай складається з трьох частин, включаючи правило виробництва, базу даних і стратегію контролю;

Виробниче правило — це структура оператора "умова + результат". Наприклад, умова: «якщо вид є динозавром», результат: «то вид є рептилією». Якщо умова в операторі виконується, результат правила може бути виконаний. База даних відповідає за збереження умов і результатів у формулюванні правил виробництва. Коли виробниче правило виконується, відповідна умова викликається з бази даних, а результат правила поміщається в базу даних як умова для інших правил. Роль стратегії контролю полягає в тому, щоб пояснити, як застосовувати правила, тобто в процесі розв'язання задачі вибрати відповідні правила. Процес від вибору правил до виконання операцій зазвичай полягає в тому, що проблема містить кілька умов: пошук відповідних умов у базі даних і пошук правил для відповідних умов. Якщо збігається більше ніж одна умова, вам потрібно відсортувати стратегію, щоб вирішити, яке правило використовувати першим. Після вибору правила виконується розділ операції (результату) правила. Робота стратегії управління зазвичай виконується в модулі ("інференційному механізмі"). Процес висновку механізму логічного висновку можна розділити на прямий і зворотній.

Експертна система на основі правил має багато переваг. Використання оператора «умова – результат» для вираження правил знань є дуже природним для людей. У той же час в експертній системі на основі правил знання та міркування зберігаються й обробляються окремо, відповідно до повсякденних звичок людей. Проте деякі недоліки також заважали подальшому розвитку такої експертної системи. Наприклад, коли правило відповідає умові, вираз оператора має бути написаний суворо відповідно до оператора в базі даних, навіть якщо відносні тонкі відмінності відхиляються через вимоги до точності відповідності. Крім того, швидкість експертної системи на основі правил не є домінуючою порівняно з іншими типами експертних систем, оскільки під час використання правила потрібно сканувати весь набір правил у базі даних[21].

* **фреймворкова експертна система.**У 1975 році Мінскі запропонував структуру даних, яка кодує інформацію, пов’язану з концепцією, і називає цю структуру даних «фреймом» [4]. Фреймворк містить назву концепції, слот для мітки головного атрибута/функції та можливі значення кожного атрибута або процес збору процедурної інформації про концепцію;

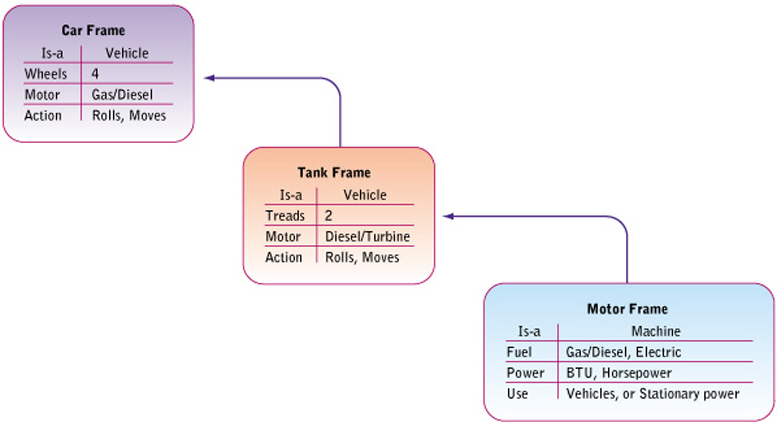


Рисунок 2.7 – Діаграма фреймової експертної системи

Коли зустрічається певний екземпляр концепції, відповідне власне значення екземпляра вводиться в структуру, яка називається екземпляром. Основу можна використовувати для міркувань. Структура містить багатоаспектну інформацію про концепцію, яку можна використовувати, навіть якщо сама інформація не спостерігається. Шукаючи структуру для опису поточного екземпляра, вона часто не відповідає ситуації. Конкретні приклади відповідатимуть деяким фрагментам кадрів, які раніше відповідали кадрам-кандидатам. Фреймворк зазвичай використовується тоді коли більша частина структури кадру може бути зіставлена, оскільки кожен кадр містить інформацію, яка може дозволити невідповідності атрибутів (функцій) для підвищення відмовостійкості кадру. Крім того, ми також можемо зберегти структуру процесу пошуку, щоб оптимізувати напрямок тесту, поки не знайдемо найбільш відповідну структуру. Експертна система на основі фреймворку використовує фреймворк у базі даних для вирішення конкретних проблем із введенням і виведенням нової інформації за допомогою механізму логічного висновку. Фреймворк представляє концепцію «класу», а фреймворк може бути «підкласом» іншого фреймворку, наприклад фреймворк «людина» є підмножиною фреймворку «людина». Деякі відношення підкласів і класів є семантичними типами, наприклад «я» + «Нью-Йорк», «Нью-Джерсі», «Лос-Анджелес». Фреймворк підкласу успадковує всі властивості каркасу свого батьківського класу, усуваючи необхідність повторно вводити властивості процесу. Однак ми повинні звернути увагу на деякі особливі обставини: деякі підкласи та їхні батьківські класи можуть мати спільні відмінності властивостей. Коли фрейм належить кільком фреймам одночасно, він також успадковує властивості всіх цих фреймів;

* **експертна система на основі нечіткої логіки.** У 1965 році американський математик Л. Заде вперше запропонував концепцію нечіткої множини, знаменуючи народження нечіткої математики [5]. Нечіткість відноситься до невибіркової природи об'єктивних речей або атрибутів, і існує серія перехідних станів між ними без очевидної лінії поділу. Нечітка теорія дозволяє людям використовувати математичні інструменти для роботи з неточними явищами в реальному світі. В експертній системі на основі нечіткої логіки нечітка логіка є основою міркування експертної системи. Цей вид методу міркування використовує нечітке правило як передумову та використовує правило нечіткої мови для виведення приблизного висновку нечіткого судження(рис. 2.8).

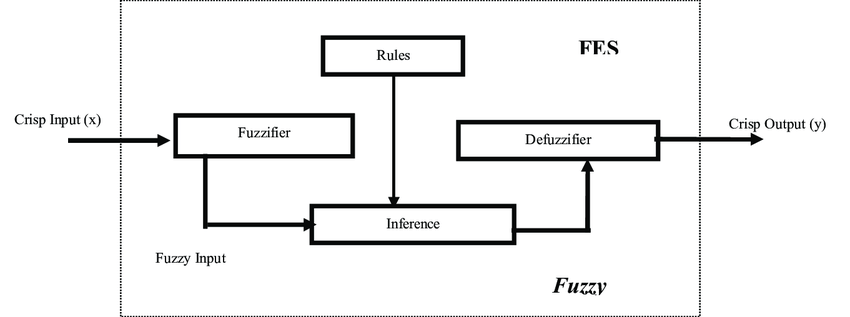


Рисунок 2.8 – Діаграма експертної системи на основі нечіткої логіки

Правила нечіткої мови включають Generalized Modus Ponens (GMP) і Generalized Modus Tollens (GMT).

Перевага експертної системи на основі нечіткої логіки полягає в тому, що вона може виражати високу майстерність експертних навичок і має достатню надійність, а також може виконувати евристичні та попередні міркування. Однак експертна система цього типу має труднощі в отриманні знань, а її аргументація залежить від нечіткої бази знань, здатність до навчання невисока, схильна до помилок.

* **експертна система на основі нейронної мережі.** Модель нейронної мережі принципово відрізняється від логічної системи, описаної вище. У нейронній мережі знання більше не перетворюються на явні правила ручною обробкою, а автоматично отримуються алгоритмом навчання та створюють власні неявні правила. Порівняно з традиційною експертною системою, нейронна мережа має більш потужну функцію: вона більш ефективна, ніж традиційна послідовна операція; має певний ступінь відмовостійкості; вага підключення нейронної мережі може бути змінена, тощо. Нейронна мережа отримує знання автоматично за допомогою екземплярів навчання. Експерти надають приклади та очікування рішення, алгоритм навчання нейронної мережі постійно змінює розподіл ваги мережі, досягаючи стабільного результату після навчання(рис. 2.9);

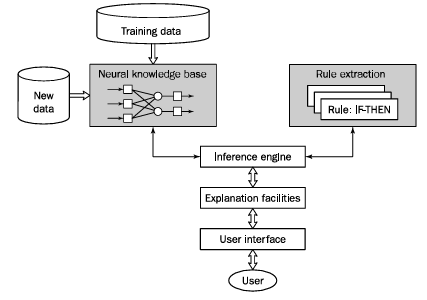


Рисунок 2.9 – Діаграма експертної системи на основі нейронної мережі

Оскільки вхід і вихід нейронної мережі є числовими, необхідно закодувати екземпляр під час використання нейронної мережі для отримання знань. Експертна система на основі нейронної мережі також має слабкі сторони: продуктивність системи обмежена набором навчальних зразків. У разі неправильного вибору набору вибірки або занадто малої вибірки здатність нейронної мережі до індукційного міркування дуже погана. Крім того, нейронна мережа не в змозі пояснити власний процес міркування та важливість зберігання знань, оскільки її модель базується на поверхневій нейронній активності людини. В даний час найбільш часто використовувана модель нейронної мережі включає модель BP, ART, CMAC, SOM і так далі. До конкретних вимог експертної системи можна застосовувати різні моделі[6].

1. **Життєвий цикл розроблення програмного забезпечення**

Життєвий цикл розробки програмного забезпечення просто описує кожне завдання, необхідне для створення програмного додатку. Це допомагає зменшити відходи та підвищити ефективність процесу розробки. Моніторинг також гарантує, що проєкт залишається за планом і продовжує залишатися реальною інвестицією для компанії.

Багато компаній поділяють ці кроки на менші підрозділи. Планування може бути розбито на технологічне дослідження, маркетингове дослідження та аналіз витрат і вигод. Інші ступені можуть зливатися один з одним. Фаза тестування може виконуватися одночасно з фазою розробки, оскільки розробникам потрібно виправити помилки, які виникають під час тестування.

Етапи життєвого циклу розробки програмного забезпечення моделі водоспаду:

* + 1. Планування.
    2. Визначення вимог.
    3. Дизайн.
    4. Розробка.
    5. Тестування.
    6. Впровадження та випуск.
    7. Технічне обслуговування.

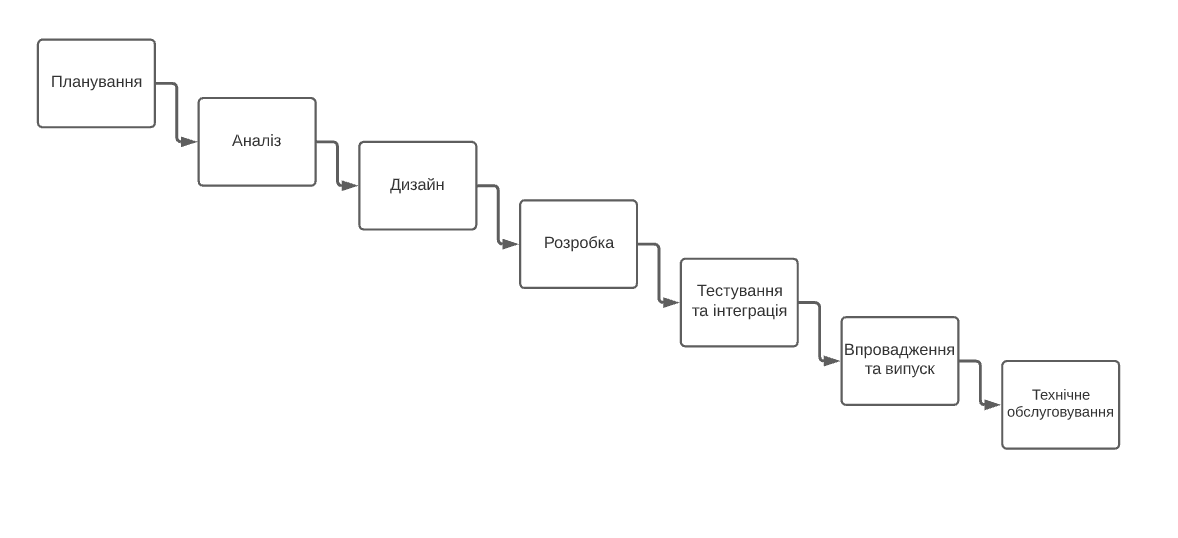


Рисунок 2.1 – Етапи життєвого циклу програмного забезпечення

* планування. На етапі планування лідери проєкту оцінюють умови проєкту, включаючи розрахунок трудових та матеріальних витрат, щоб створити графік з цілями. Планування також повинно включати сфери зворотного зв'язку від зацікавлених сторін або будь-кого, хто збирається отримати користь від заявок. Обсяг проєкту повинен бути чітко визначений, мета окресленої програми та межі, необхідні для того, щоб проєкт не розширювався поза межами чи зміщенням;
* визначення вимог. Після завершення етапу планування наступним кроком SDLC є визначення та документування потреб програмного забезпечення. Цей процес, проведений за допомогою документа "Специфікація вимог програмного забезпечення", також відомого як документ "SRS". Він включає все, що повинно бути розроблене під час життєвого циклу проєкту;
* дизайн. документи з проєктування системи та програмного забезпечення готуються відповідно до документу про специфікацію вимог. Це допомагає визначити загальну архітектуру системи. Після визначення дизайну може бути створений прототип ранньої версії програмного забезпечення, щоб продемонструвати основне уявлення про те, як буде виглядати програма, як вона буде реагувати та що вона здатна робити;
* розробка. Фаза проєктування системи закінчена, наступна фаза кодує. На цій фазі розробники починають будувати всю систему, записуючи код за допомогою обраної мови програмування. У фазі кодування завдання поділяються на одиниці або модулі та присвоюються різним розробникам. Це найдовша фаза процесу життєвого циклу розробки програмного забезпечення. На цій фазі розробник повинен дотримуватися певних заздалегідь визначених рекомендацій щодо кодування. Їм також потрібно використовувати інструменти програмування, такі як компілятор, перекладачі, дебагер для створення та реалізації коду;
* тестування. Після того, як програмне забезпечення буде завершено, і воно розгортається в умовах тестування. Команда тестування починає тестувати функціональність усієї системи. Це робиться для того, щоб перевірити, чи працює вся програма відповідно до вимоги клієнта. Під час цієї фази команда з питань якості та тестування може знайти деякі помилки/дефекти. Команда розробки виправляє помилку та відправляє назад до QA для повторного тесту. Цей процес триває, поки програмне забезпечення не стане стабільним та буде працювати відповідно до потреб бізнесу цієї системи;
* впровадження та випуск. Після того, як фаза тестування програмного забезпечення закінчиться і в системі не залишилося помилок, то починається остаточний процес розгортання. На основі відгуків, наданих менеджером проєкту, остаточне програмне забезпечення випускається та перевіряється на баги, якщо такі є;
* технічне обслуговування. Після того, як програма буде розгорнута і використовується, остаточна фаза виявляє помилки, які не були виявлені під час тестування та вирішують їх - це може розпочати власний ітеративний процес.

1. **Принцип роботи веб-сайту**

Веб-сторінка - це документ, доступний у World Wide Web. Веб-сторінки зберігаються на веб -сервері і їх можна переглянути за допомогою веб -браузера.

Веб-сторінка може отримати величезну інформацію, включаючи текст, графіку, аудіо, відео та гіпер -посилання. Гіпер -посилання - це посилання на інші веб-сторінки.

Статичні веб-сторінки також відомі як стаціонарна веб-сторінка. Вони завантажуються в браузер клієнта, оскільки саме вони зберігаються на веб-сервері. Такі веб-сторінки містять лише статичну інформацію. Користувач може лише прочитати інформацію, але не може робити жодної модифікації або взаємодіяти з інформацією.

Статичні веб-сторінки створюються лише за допомогою HTML. Статичні веб-сторінки використовуються лише тоді, коли інформацію більше не потрібно змінювати.

Динамічна веб-сторінка показує різну інформацію в інший момент часу. Можна змінити контент веб-сторінки, не завантажуючи всю веб-сторінку. Це стало можливим за допомогою технології AJAX.

Веб-браузери використовують інший мережевий протокол, який називається протоколом передачі гіпертексту (HTTP).

HTTPS — це абревіатура від Hypertext Transfer Protocol Secure. Це безпечне розширення або версія HTTP. Цей протокол в основному використовується для забезпечення безпеки даних, які передаються між веб-сайтом і веб-браузером. Він широко використовується в Інтернеті та використовується для безпечного зв’язку. Цей протокол використовує номер порту 443 для передачі даних.

HTTPS і HTTP є одним і тим же протоколом. Основна відмінність полягає в тому, що протокол HTTPS має додатковий рівень шифрування (SSL/TLS). Сайти HTTP змінюються на HTTPS, отримуючи сертифікат SSL (іноді його називають сертифікатом безпеки або цифровим сертифікатом). Сертифікат SSL – це невеликий файл даних, який захищає передачу конфіденційних даних між веб-браузером і веб-сервером.

Сертифікат SSL шифрує ці дані, роблячи їх нечитабельними під час процесу передачі. Він містить відкритий ключ, який дозволяє користувачам безпечно надсилати конфіденційну інформацію зі свого веб-браузера. Власник домену має закритий ключ, який розшифровує цю інформацію, коли вона потрапляє на сервер. Ця пара публічно-приватного ключа забезпечує безпечне з’єднання.

Щоб домен став підтримувати HTTPS, він має отримати сертифікат SSL від надійного центру сертифікації (CA). Коли веб-браузер намагається підключитися до сервера через HTTPS, він перевіряє, чи сертифікат SSL відповідає доменному імені, яке намагається ввести користувач, за допомогою процесу, який називається рукостисканням SSL/TLS.

Сертифікат містить цифровий підпис від центру сертифікації, щоб підтвердити, що сертифікат було видано для вказаного доменного імені. Коли веб-браузер перевіряє підпис сертифіката, щоб встановити довіру до сервера, з’єднання стає безпечним. Усі довірені ЦС автоматично розпізнаються браузерами.

Коли користувач вводить адресу, у свій браузер, він наказує йому відкрити TCP-канал до сервера, який відповідає на цю URL-адресу (Uniform Resource Locator). URL-адреса схожа на домашню адресу чи номер телефону, оскільки вона описує, як зв’язатися із доменом.

У цій ситуації комп’ютер користувача, який робить запит, називається клієнтом. URL-адреса, яку запитує користувач, є адресою, яка належить серверу.

Після встановлення TCP-з’єднання клієнт надсилає HTTP-запит GET на сервер для отримання веб-сторінки, яку він має відобразити. Після того, як сервер надіслав відповідь, він закриває TCP-з’єднання. Якщо ви знову відкриваєте веб-сайт у своєму браузері або якщо ваш браузер автоматично запитує щось із сервера, відкривається нове з’єднання, яке виконується за тим самим процесом, який описано вище. Запити GET є одним із видів HTTP-методів, які клієнт може викликати.

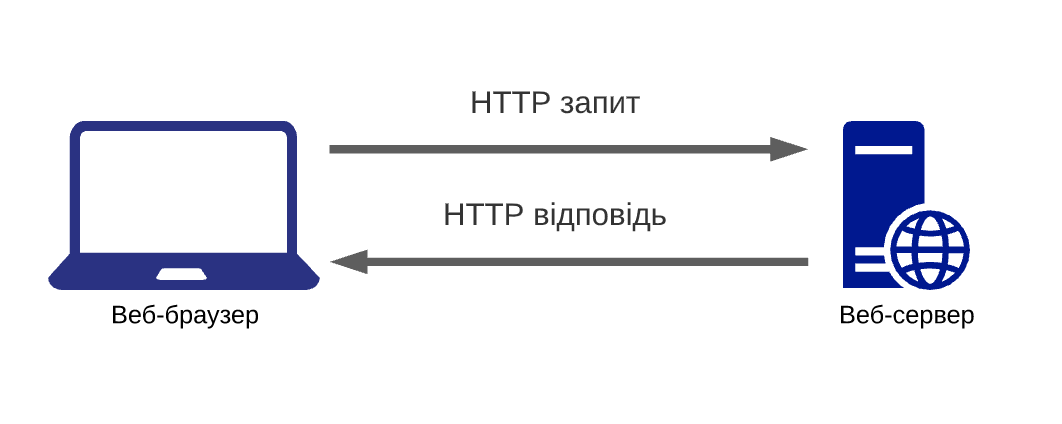
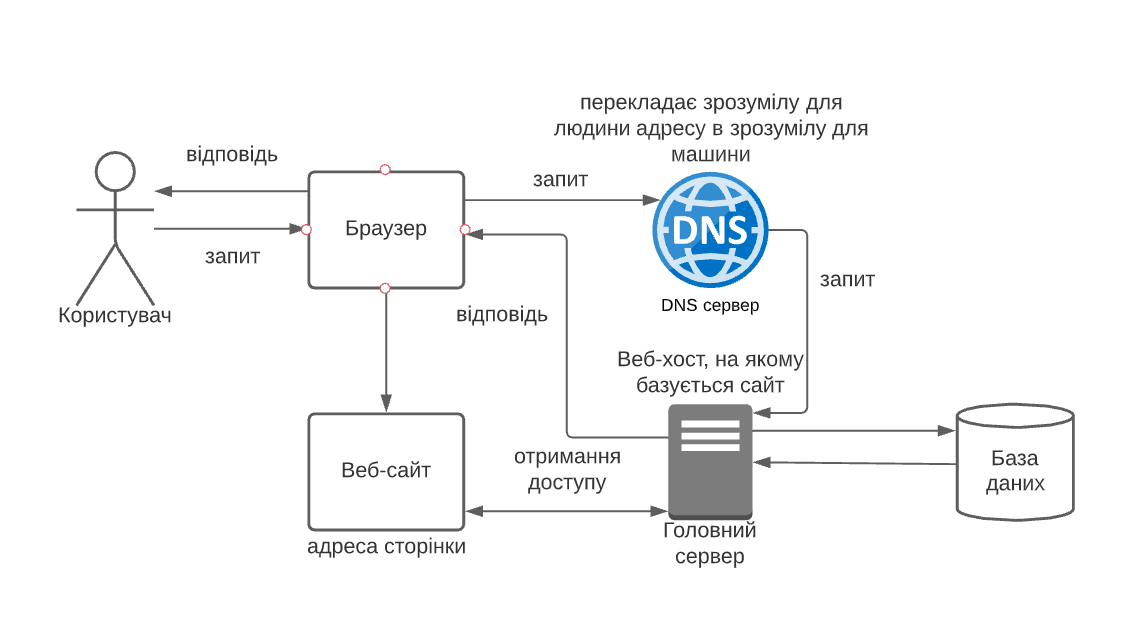


Рисунок 2.3 – Принцип роботи запитів HTTP

Кожен HTTP-запит, зроблений в Інтернеті, містить у собі ряд закодованих даних, що несуть різні типи інформації. Типовий HTTP-запит містить:

* тип версії HTTP;
* URL;
* метод HTTP;
* заголовки HTTP-запитів;
* необов’язковий корпус HTTP.

Нижче представлена схема, яка загально демонструє роботу веб-додатку.

  
Рисунок 2.5 – Принцип роботи веб-сайту

# РОЗДІЛ 3 ПРОЦЕС РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-КОНФЕРЕНЦІЙ

1. **Вибір та обґрунтування веб-технологій для розроблення  
    програмного забезпечення**
   * 1. **Обґрунтування вибору використання мови PHP.** PHP — це мова програмування та сценаріїв для створення динамічних інтерактивних веб-сайтів. WordPress написаний з використанням PHP як мови сценаріїв. Як і WordPress, PHP також має відкритий код.

PHP є серверною мовою, що означає, що він працює на вашому сервері веб-хостингу. Щоразу, коли хтось відвідує ваш веб-сайт, його браузер зв’язується з вашим сервером, щоб запитати сторінку. Код PHP виконується на сервері та створює сторінку HTML для надсилання відвідувачу. Потім відвідувач бачить HTML-сторінку у своєму браузері. Вони не бачать PHP-скрипт, оскільки він є лише на сервері хостингу[14].

До причин вибору мови програмування PHP належать:

* доступність. Однією з головних переваг PHP є відкритий вихідний код і безкоштовність, що допомагає розробникам швидко інсталювати його та бути готовим для використання. Таким чином, PHP доступний безкоштовно для кожного користувача, а спільнота розробників PHP надає відмінну технічну підтримку. Тому всі його компоненти можна вільно використовувати та розповсюджувати;
* ­захищеність. PHP є одним із найбезпечніших способів розробки веб-сайтів і динамічних веб-додатків. PHP має кілька рівнів захисту для запобігання загрозам і зловмисним атакам. Використання PHP для розробки гібридних веб-додатків є надзвичайно важливим;
* швидіксть. PHP використовує власний простір пам’яті, тому навантаження на сервер і час завантаження автоматично зменшуються, що призводить до вищої швидкості обробки. Швидкість обробки висока, і веб-додатки, такі як електронна комерція, CRM, CMS і форуми, також розробляються швидше. Це головне значення використання мови PHP у веб-розробці;
* незалежність від платформи. PHP працює майже на кожній доступній платформі, що дозволяє працювати в різних операційних системах. Будь то Linux, UNIX, Mac OS і Windows; він також підтримує всі основні операційні системи. Це одна з життєво важливих переваг вивчення PHP для розвитку ваших навичок веб-розробки;
* підтримка всіх великих веб-сервісів. Окрім операційних систем, він також підтримує всі основні веб-сервери, такі як Apache, Microsoft IIS, Netscape, персональний веб-сервер, сервер iPlanet тощо. Оскільки мова програмування працює з багатьма операційними системами, її можна розгортати на різних платформах;
* підтримка всіх лідуючих баз даних. PHP підтримує всі провідні бази даних, включаючи MySQL, ODBC, SQLite тощо. Це головна перевага використання PHP для веб-розробки;
* PHP є однією з найкращих зручних для користувача мов програмування в галузі. Він також забезпечує більшу гнучкість, ніж C, C++ і ASP, і загалом допомагає покращити відвідуваність веб-сайту. Щоб розробити складні, динамічні та зручні веб-додатки, PHP є лише першою перевагою для всіх веб-розробників. Отже, ця функція є однією з головних переваг вивчення PHP.

Обґрунтування вибору використання мови JavaScript. JavaScript - це динамічна мова програмування, яка використовується для веб -розробки, у веб -додатках, для розробки ігор та багато іншого. Це дозволяє впроваджувати динамічні функції на веб -сторінках, які неможливо виконати лише за допомогою HTML та CSS.

* швидкість. Оскільки JavaScript є «інтерпретованою» мовою, це зменшує час, необхідний іншим мовам програмування, таким як Java, для компіляції. JavaScript також є сценарієм на стороні клієнта, що прискорює виконання програми, оскільки економить час, необхідний для підключення до сервера;
* оскільки JavaScript працює на стороні клієнта, перевірка даних можлива в самому браузері, а не надсилання їх на сервер. У разі будь-якої невідповідності весь веб-сайт не потрібно перезавантажувати. Браузер оновлює лише вибраний сегмент сторінки;
* JavaScript покращує продуктивність веб-сайтів і веб-додатків, зменшуючи довжину коду. Коди містять менше накладних витрат завдяки використанню різноманітних вбудованих функцій для циклів, доступу до DOM тощо;
* популярність. Оскільки всі сучасні браузери підтримують JavaScript, його можна побачити майже всюди. Усі відомі компанії використовують JavaScript як інструмент, включаючи Google, Amazon, PayPal тощо;
* JavaScript ідеально працює з іншими мовами програмування, тому багато розробників віддають перевагу йому при розробці багатьох програм. Ми можемо вставити його на будь-яку веб-сторінку або в сценарій іншої мови програмування.

Також із використанням мови програмування Javascript, з’являється можливість використовувати Ajax. Ajax є скороченням від Asynchronous JavaScript and XML, що відноситься до набору методів веб-розробки, а не до фактичної мови програмування. Однак Ajax широко використовується в програмуванні на стороні клієнта, щоб дозволити надсилання та отримання даних до та з бази даних/сервера. Особливістю використання програмування Ajax є те, що ви можете обмінюватися даними у фоновому режимі, фактично не порушуючи роботу користувача.

Цей метод надзвичайно корисний як для продуктивності веб-сайту, так і для зручності використання. З точки зору зручності використання, відвідувачі можуть отримати вигоду від перегляду певної інформації, згенерованої без необхідності перезавантажувати сторінку.

* + 1. **Обґрунтування вибору використання MySQL.** MySQL – це система керування базами даних, яка дозволяє керувати реляційними базами даних. Це програмне забезпечення з відкритим кодом, яке підтримується Oracle.

До причин вибору СУБД MySQL можна віднести:

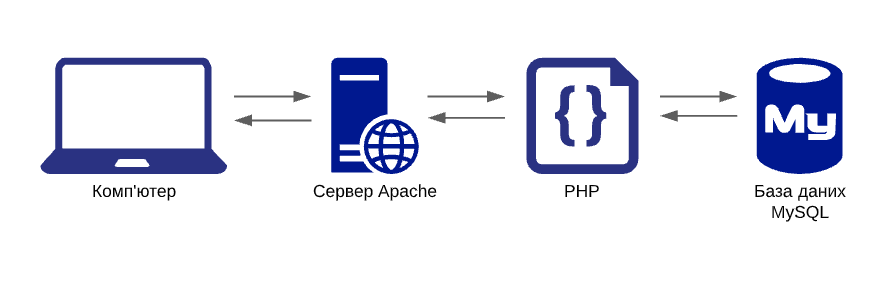
* простота у використанні. MySQL простий у використанні. Ми повинні отримати лише базові знання SQL. Ми можемо створювати та взаємодіяти з MySQL, використовуючи лише кілька простих операторів SQL;
* безпечність. MySQL складається з надійного рівня безпеки даних, який захищає конфіденційні дані від зловмисників. Крім того, паролі зашифровані в MySQL;
* архітектура клієнт/сервер. MySQL слідує роботі архітектури клієнт/сервер. Існує сервер бази даних (MySQL) і довільна кількість клієнтів (прикладних програм), які спілкуються з сервером; тобто вони можуть запитувати дані, зберігати зміни тощо;
* безкоштовність. MySQL є безкоштовним для використання, тому ми можемо безкоштовно завантажити його з офіційного веб-сайту MySQL;
* масштабованість. MySQL підтримує багатопотоковість, що робить його легко масштабованим. Він може обробляти майже будь-який обсяг даних, до 50 мільйонів рядків або більше. Стандартний розмір файлу становить приблизно 4 ГБ. Однак ми можемо збільшити це число до теоретичного обмеження в 8 ТБ даних;
* швидкість. MySQL вважається однією з дуже швидких мов баз даних, що підтримується великою кількістю порівняльних тестів;
* гнучкість. MySQL підтримує велику кількість вбудованих програм, що робить MySQL дуже гнучким;
* сумістність. MySQL сумісний із багатьма операційними системами, такими як Novell NetWare, Windows Linux, багатьма різновидами UNIX (такими як Sun Solaris, AIX і DEC UNIX), OS/2, FreeBSD та іншими. MySQL також надає засоби, які клієнти можуть запускати на тому ж комп’ютері, що й сервер, або на іншому комп’ютері (з’єднання через локальну мережу або Інтернет);
* можливість відкату. MySQL дозволяє повертати транзакції, фіксувати їх і відновлювати після збою;
* ефективність пам'яті. Його ефективність висока, оскільки він має дуже низьку проблему витоку пам'яті;
* висока продуктивність. MySQL швидший, надійніший і дешевший завдяки своїй унікальній архітектурі системи зберігання. Він забезпечує дуже високу продуктивність порівняно з іншими базами даних без втрати основної функціональності програмного забезпечення. Має утиліти швидкого завантаження через іншу кеш-пам'ять;
* підтримка GUI. MySQL надає уніфікований візуальний інструмент графічного інтерфейсу користувача бази даних під назвою "MySQL Workbench" для роботи з архітекторами баз даних, розробниками та адміністраторами баз даних. MySQL Workbench забезпечує розробку SQL, моделювання даних, міграцію даних і комплексні засоби адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого. MySQL має повну підтримку GUI від MySQL Server версії 5.6 і вище;
* незалежність від платформи. MySQL може завантажувати, інсталювати та запускати на більшості доступних операційних систем.
  + 1. **Обґрунтування вибору використання XAMPP.** XAMPP — це безкоштовний міжплатформний веб-сервер із відкритим кодом. XAMPP це скорочена форма для кросплатформних, Apache, MySQL, PHP і Perl. XAMPP є популярний міжплатформний веб-сервер, який дозволяє програмістам писати та тестувати свій код на локальному веб-сервері. Він був створений Apache Friends, і можуть переглядати або змінювати його вихідний код. Він включає MariaDB, Apache HTTP Server та інтерпретатори для PHP і Perl. Завдяки простоті розгортання XAMPP розробник може швидко та легко інсталювати стек WAMP або LAMP в операційній системі з додатковою перевагою, що також можна завантажити звичайні додаткові програми, такі як WordPress і Joomla. 

Рисунок 3.1 – Схема роботи сайту на локальному комп’ютері

Основними причинами вибору збірки XAMPP можна вважати:

* безкоштовне та зручне програмне забезпечення доступне для платформ Windows, Mac і Linux;
* це легке програмне забезпечення з відкритим кодом, яке дуже легко налаштувати, розробити, протестувати та розгорнути;
* виконує багато адміністративних завдань і економить час, керуючи змінами конфігурації тощо;
  + 1. **Обґрунтування вибору використання фреймворку Codeigniter**. CodeIgniter — веб-фреймворк для швидкої розробки програмного забезпечення з відкритим вихідним кодом, призначений для створення динамічних веб-сайтів. CodeIgniter частково базується на методології розробки модель-подання-контролер (MVC). Це дозволяє швидше розгортати проєкти, ніж писати код з нуля. Є багатий набір бібліотек, зручний інтерфейс і логічна структура доступу до бібліотек.

До переваг цього фреймворку можна віднести наступні пункти:

* обробка помилок. CodeIgniter забезпечує зручні та керовані інтерфейси, які дуже допомагають у розпізнаванні та обробці помилок у всьому додатку. Ви отримаєте вказівки щодо усунення помилок і інформацію про налагодження. Він покаже всі помилки PHP у вашій програмі;
* потужний метод перевірки. Функція перевірки форми допомагає написати код в один рядок. Ви можете використовувати ефективну систему перевірки, щоб писати код просто, без помилок;
* компактність. Codeigniter розроблено як легкий PHP-фреймворк, як і інші надійні PHP-фреймворки, він займає дуже малу площу. Невеликий розмір робить CodeIgniter швидшим за інші широко використовувані веб-фреймворки;
* простий у налаштуванні. Початківці та досвідчені PHP-програмісти можуть використовувати Codeigniter без додаткових витрат часу та зусиль. Чітка документація та надійна підтримка спільноти ще більше полегшують програмістам PHP використання Codeigniter для розробки якісних веб-додатків. Прості у використанні параметри налаштування, надані фреймворком PHP;
* особливості імміграції. Досить легко керувати оновленнями схеми бази даних у різних областях за допомогою аспекту міграції, імміграція з сервера на сервер проста в Codeigniter;
* розробка з вбудованими функціями безпеки. Codeigniter спрощує для програмістів переписування безпечних програм PHP, надаючи ряд функцій. Функції, надані програмістами Codeigniter, допомагають програмістам захистити веб-додаток від загальних попереджень про безпеку, таких як впровадження SQL, атаки міжсайтових сценаріїв і віддалене виконання коду. Програмісти можуть більше перевіряти попередження безпеки, просто викликаючи певні функції.

1. **Проєктування інформаційної системи**

Під час проєктування експертної системи необхідно розробити документ, що описує аспекти майбутнього веб-додатку.

Для цього треба створити контекстну діаграму типу IDEF0, яка відобразить зв’язки процесу розробки експертної системи з зовнішнім середовищем.

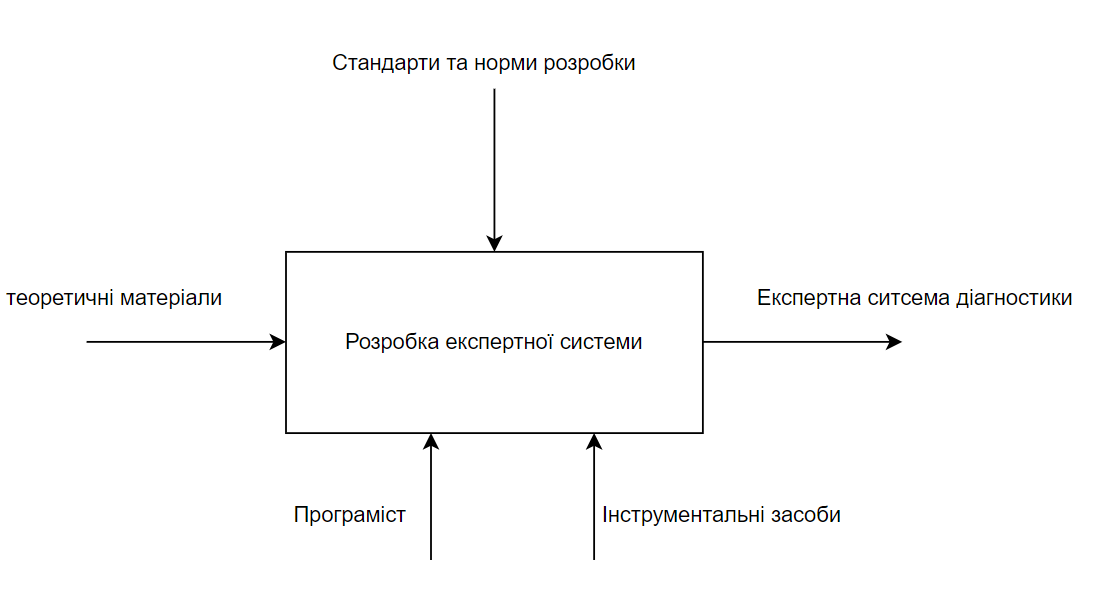


Рисунок 3.2 – Контекстна діаграма процесу розробки сайту

На вході бачимо теоретичні матеріали, які містять правила принципу роботи системи та стандарти життєвого циклу програмного забезпечення. На стороні керування знаходяться стандарти та норми розробки. Механізмом виступає програміст, який буде писати код та інструментальні засоби, наприклад мови програмування, сервери, бази знань та інше. На виході ми отримуємо експертну систему діагностики.

На основі контекстної діаграми будуємо діаграму декомпозиції.

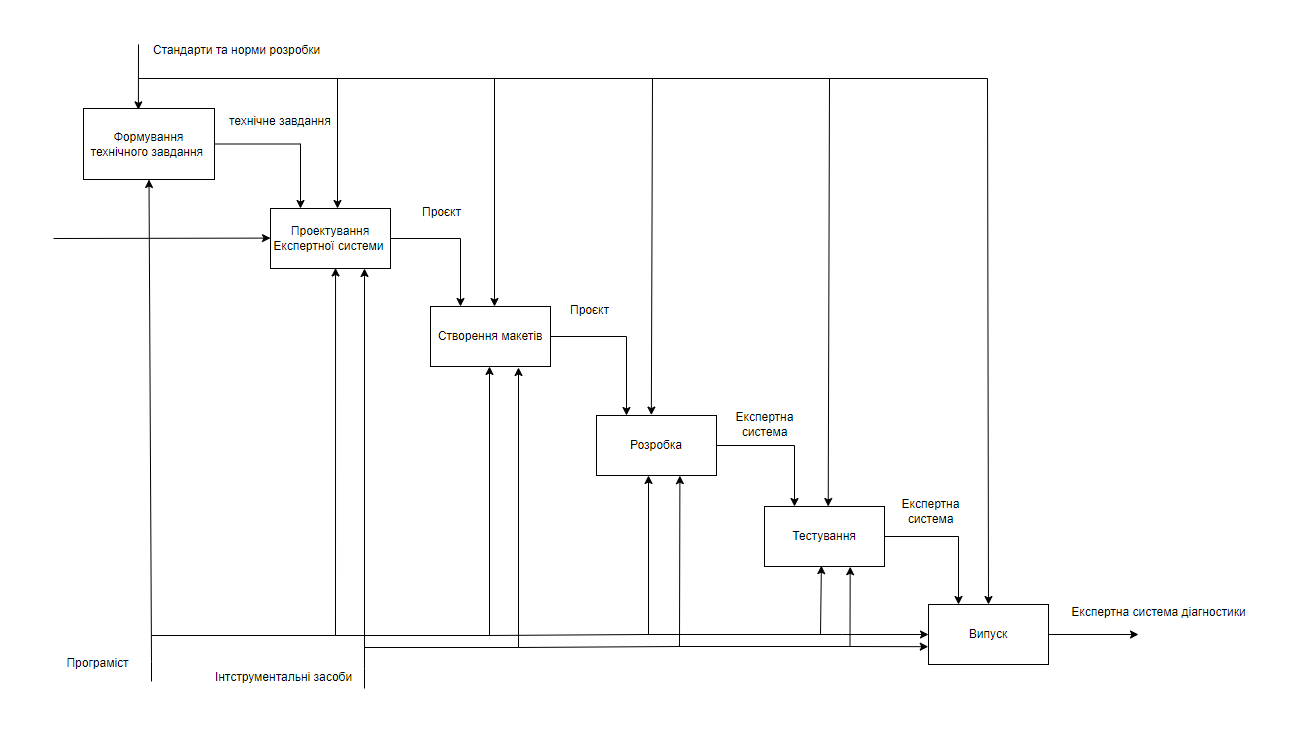


Рисунок 3.3 – Діаграма декомпозиції процесу розробки сайту

Виділимо основні етапи створення експертної системи:

1. Формування чіткого технічного завдання, який описує необхідні функції сайту. Основними технічними завданнями будуть: реалізація блоку реєстрації та авторизації, реалізація логіки експертної системи, реалізація можливості завантаження мануалів та реалізація додавання коментарів користувачів.
2. Проєктування ЕС. Виділення усіх етапів розробки, планування логіки роботи веб-додатку.
3. Створення макетів. Створення візуальних шаблонів за допомогою яких далі буде створена верстка веб-додатку.
4. Розробка. Повна реалізація усіх функцій сайту, під’єднання бази даних, верстка макету та поєднання верстки із логікою.
5. Тестування. Перевірка роботи усіх функцій сайту, перевірка верстки.
6. Випуск. Завершення роботи.

Для формування логіки роботи модулів веб-додатку, створимо діаграму варіантів використання, яка буде зображувати ролі акторів та їх можливості роботи в ІС.

Далі, необхідно створити діаграму варіантів використання, на якій буде відображено акторів, їх ролі та можливості під час роботи із експертною системою.

Діаграма варіантів використання використовується для представлення динамічної поведінки системи. Вона інкапсулює функціональні можливості системи, об’єднуючи варіанти використання, акторів та їхні зв’язки, моделює завдання, служби та функції, необхідні системі/підсистемі програми. Він описує високорівневу функціональність системи, а також розповідає, як користувач працює з системою.

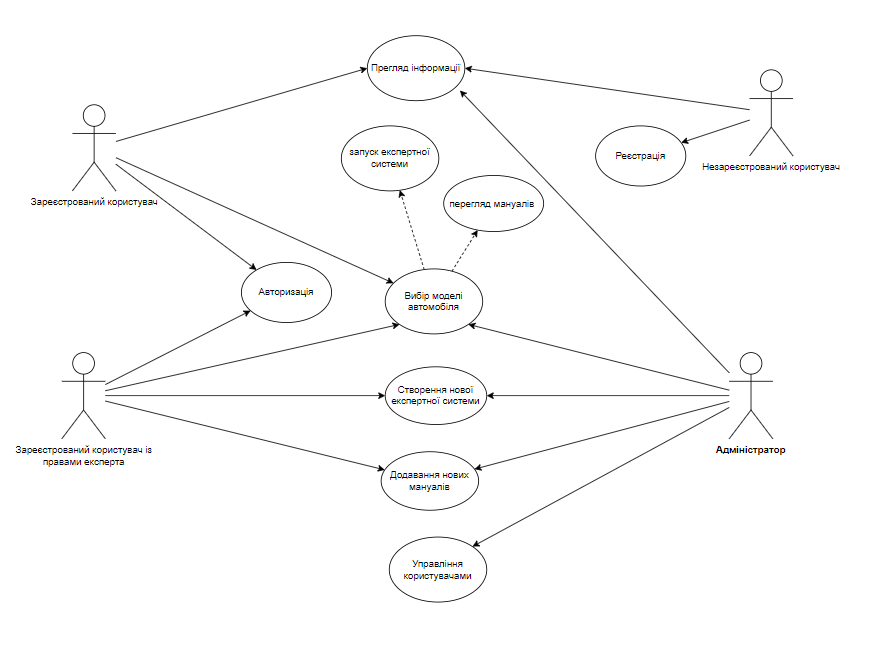


Рисунок 3.4 – Діаграма варіантів використання для ІС інтернет-конференції

Таблиця 3.1 – Виявлення акторів

|  |  |
| --- | --- |
| Актор | Короткий опис |
| Зареєстрований користувач | Має можливість перегляду посібників та запуску експертної системи. |
| Зареєстрований користувач із підвищеними правами | Має усі можливості звичайного користувача, але також може завантажувати посібники до додатку. |
| Адміністратор ресурсу | Має усі права користувачів, а також може управляти користувачами |

Розглянемо діаграму діяльності для надання користувачу прав на завантаження файлу.

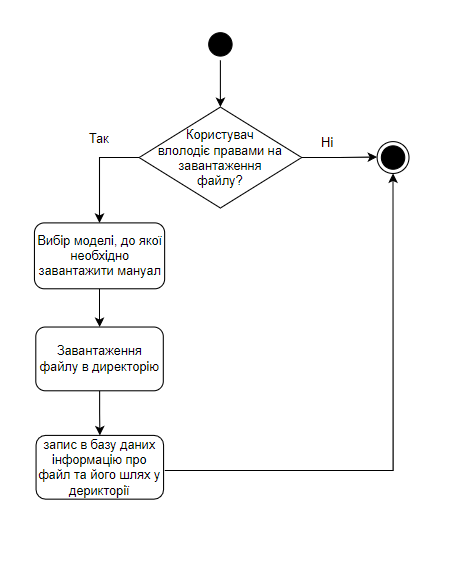


Рисунок 3.5 –Діаграма діяльності для завантаження документу на сайт

Першою чергою перевіряється, чи має користувач права на завантаження посібника, якщо права підтверджуються, користувач повинен обрати тип двигуна за яким буде закріплено посібник. Далі файл завантажується у директорію, додатку, а інформація про його назву та шлях зберігається у базі даних

1. **Розробка структури сайту та макетів**

Розглянемо структуру сайту для ІС, який містить усі необхідні модулі (мал. 3.8).

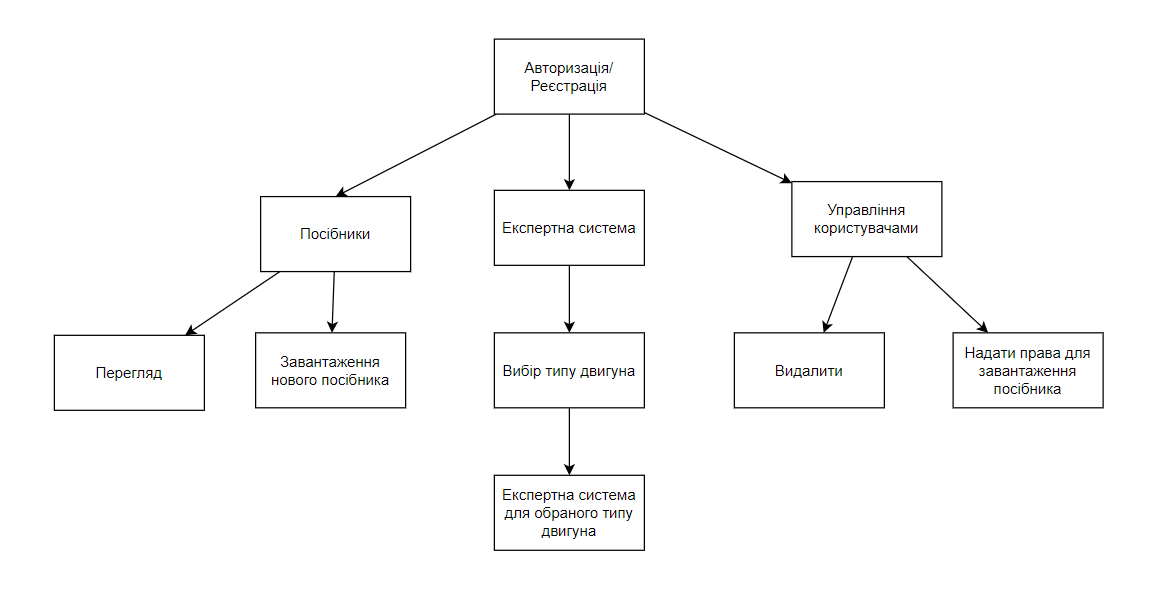


Рисунок 3.6 – Структура сайту

Для того щоб почати користуватися веб-додатком, спочатку необхідно пройти авторизацію, у випадку відсутності аккаунту, необхідно пройти реєстрацію, після цього, адміністратор ресурсу надає права для користування веб-додатком.

Після того як авторизація пройшла успішно, користувач отримує доступ для перегляду посібників для двигунів а також може перейти на сторінку із експертною системою для обраного двигуна.

У випадку, якщо користувач має права типу «up-user», або «admin», він також має змогу видаляти або завантажувати нові посібники для будь-якого типу двигуна.

У випадку, якщо користувач має права типу «admin», в нього буде можливість перейти на сторінку із списком користувачів, де він має змогу надати права доступу до додатку для нових користувачів, а також надавати більш розширені права користувачам.



Рисунок 3.7 – Макет експертної системи

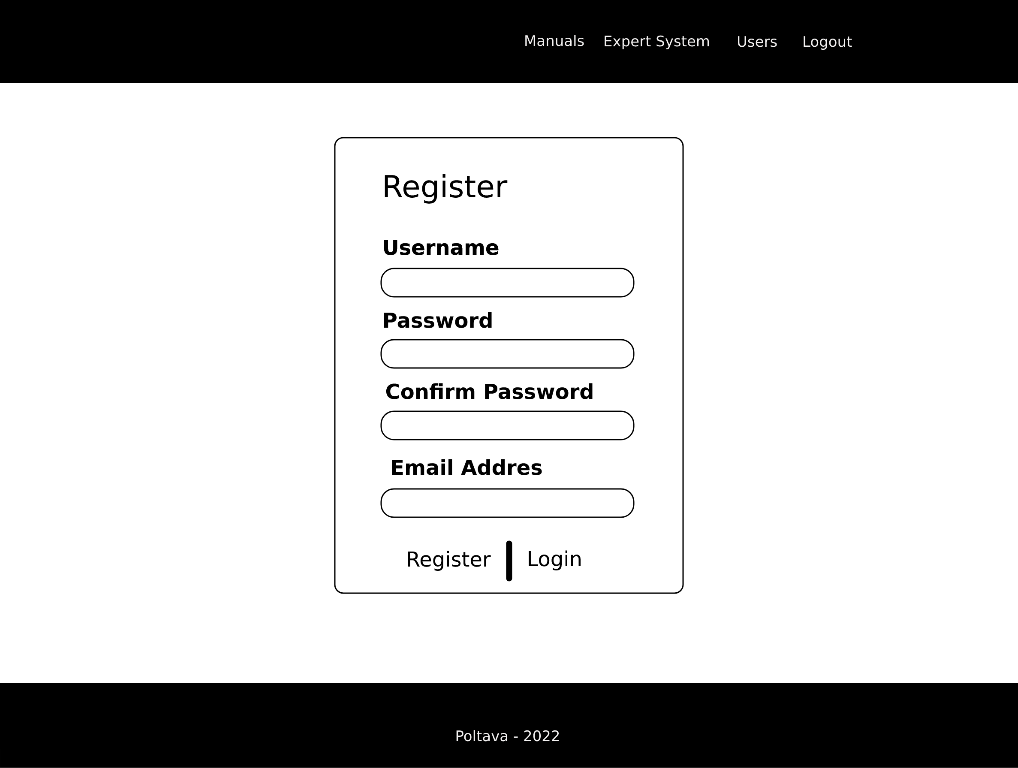


Рисунок 3.8 – Макет експертної системи

1. **Розробка бази даних**

Під час Розробки веб-додатку для експертної системи необхідно створити базу даних, яка буде містити наступні таблиці: (мал. 3.10):

* користувачі;
* ролі користувачів;
* посібники;
* моделі двигунів;
* зв’язна таблиця експертних систем із двигунами(таблиця помилок);
* таблиці із базою знань для кожного типу двигуна.

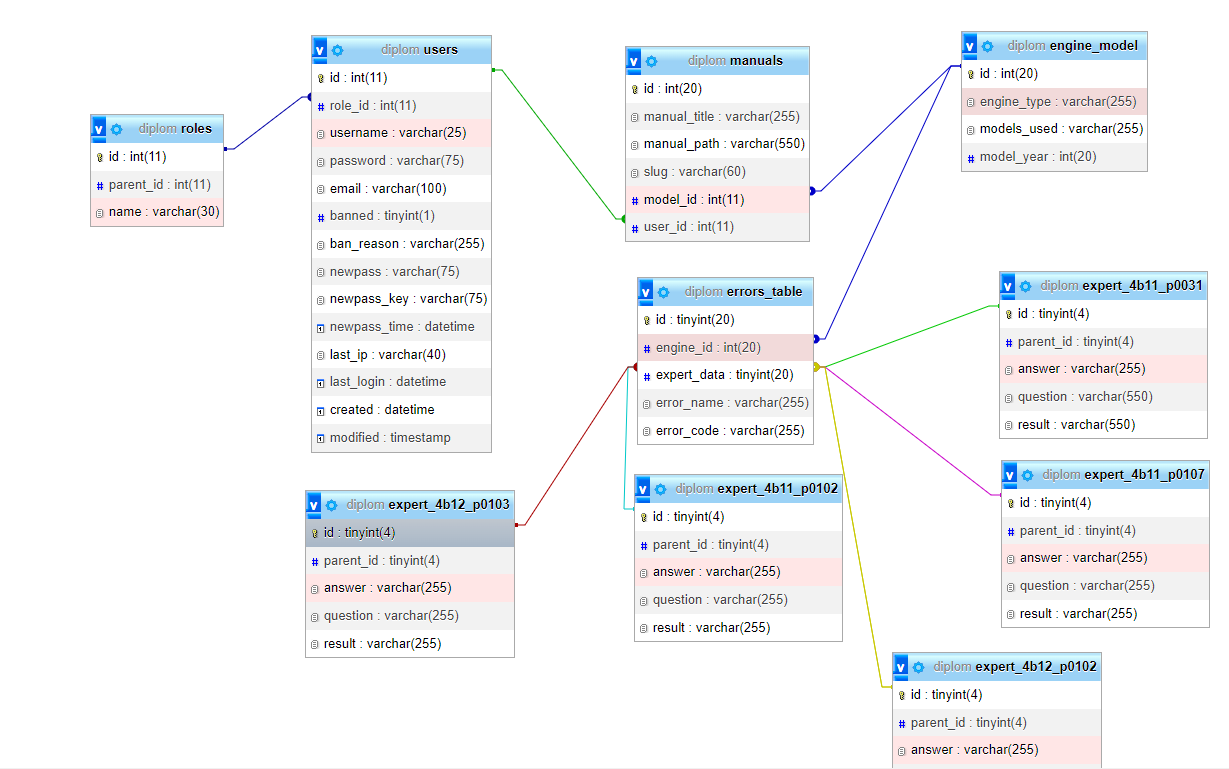


Рисунок 3.9 – Схема бази даних

Користувачі – таблиця у якій міститься інформація про користувачів.

Таблиця 3.2 – Таблиця «users»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 11 | Первинний ключ таблиці |
| 2 | role\_id | integer | 11 | Зовнішній ключ для таблиці «roles» |
| 3 | username | varchar | 25 | Текстове поле для логіну користувача |
| 4 | password | varchar | 75 | Текстове поле для паролю користувача |
| 5 | email | varchar | 100 | Текстове поля для пошти користувача |

Ролі – таблиця яка містить усі ролі користувачів, за допомогою яких реалізується розмежування доступу у веб-додатку.

Таблиця 3.3 – Таблиця «roles»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 11 | Первинний ключ таблиці |
| 2 | name | varchar | 30 | Текстове поле для назви ролі |

Модель двигуна – таблиця, яка містить інформацію про типи двигунів, моделі автомобілів, на які він встановлений та рік випуску.

Таблиця 3.4 – Таблиця «engine\_model»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 11 | Первинний ключ таблиці |
| 2 | engine\_type | integer | 255 | Текстове поле для назви типу двигуна |
| 3 | models\_used | varchar | 255 | Текстове поле для логіну списку моделей автомобілів на які встановлено модель двигуна |
| 4 | model\_year | integer | 20 | Поле для року створення двигуна |

Посібники – таблиця, яка містить інформацію про посібники для типів двигунів.

Таблиця 3.5 – Таблиця «manuals»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | Id | integer | 11 | Первинний ключ таблиці |
| 2 | manual\_title | integer | 255 | Текстове поле для назви файлу посібника |
| 3 | manual\_path | varchar | 550 | Текстове поле для повного шляху файлу посібника |
| 4 | slug | varchar | 20 | Слаг посібника |
| 5 | model\_id | integer | 11 | Ідентифікатор типу двигуна до якого має відношення посібник |
| 6 | user\_id | integer | 11 | Ідентифікатор користувача |

Таблиця помилок пов’язує таблицю двигуна із таблицями баз знань для експертних систем та містить назву та код помилки.

Таблиця 3.6 – Таблиця «errors\_table»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | Id | integer | 11 | Первинний ключ таблиці |
| 2 | engine\_id | integer | 255 | Зовнішній ключ для таблиці типів двигунів |
| 3 | error\_code | varchar | 255 | Код помилки |
| 4 | error\_name | varchar | 255 | Назва помилки |
| 5 | expert\_data | integer | 11 | Зовнішній ключ таблиці із інформацією для експертної системи |

Таблиці із префіксом «expert\_» містить інформацію для експертної системи певної помилки.

Таблиця 3.7 – Таблиця із префіксом «expert\_»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | Id | integer | 11 | Первинний ключ таблиці |
| 2 | parent\_id | integer | 255 | Ідентифікатор питання-батька |
| 3 | answer | varchar | 255 | Відповідь |
| 4 | question | Varchar | 255 | Питання |
| 5 | result | Varchar | 255 | Результат |

1. **Реалізація програмного продукту**

Для можливості користування додатком, користувачу необхідно увійти в аккаунт за своїм логіном та поштою(рис. 3.11) або пройти реєстрацію (рис. 3.12).

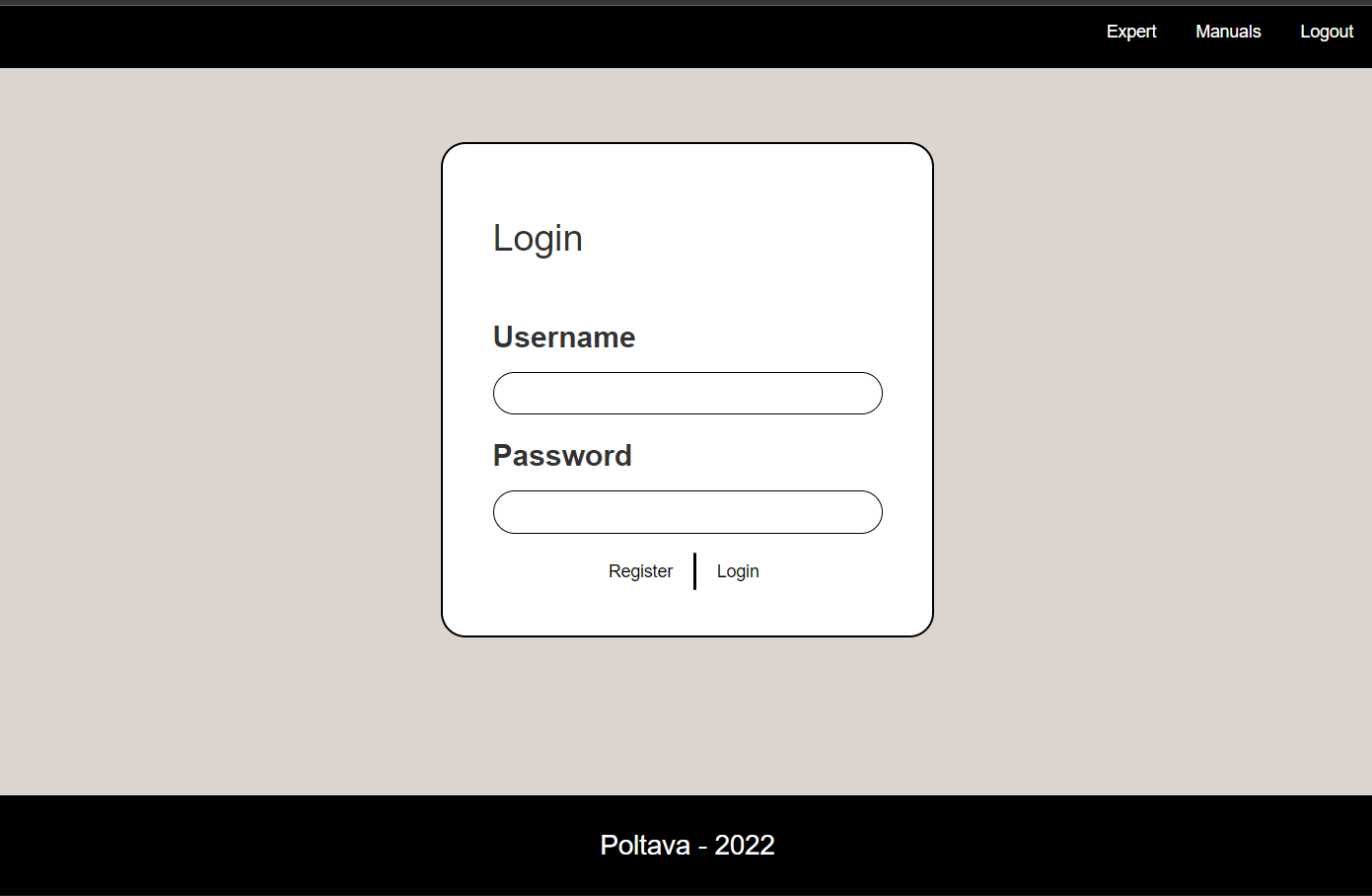


Рисунок 3.10 – Авторизація користувача

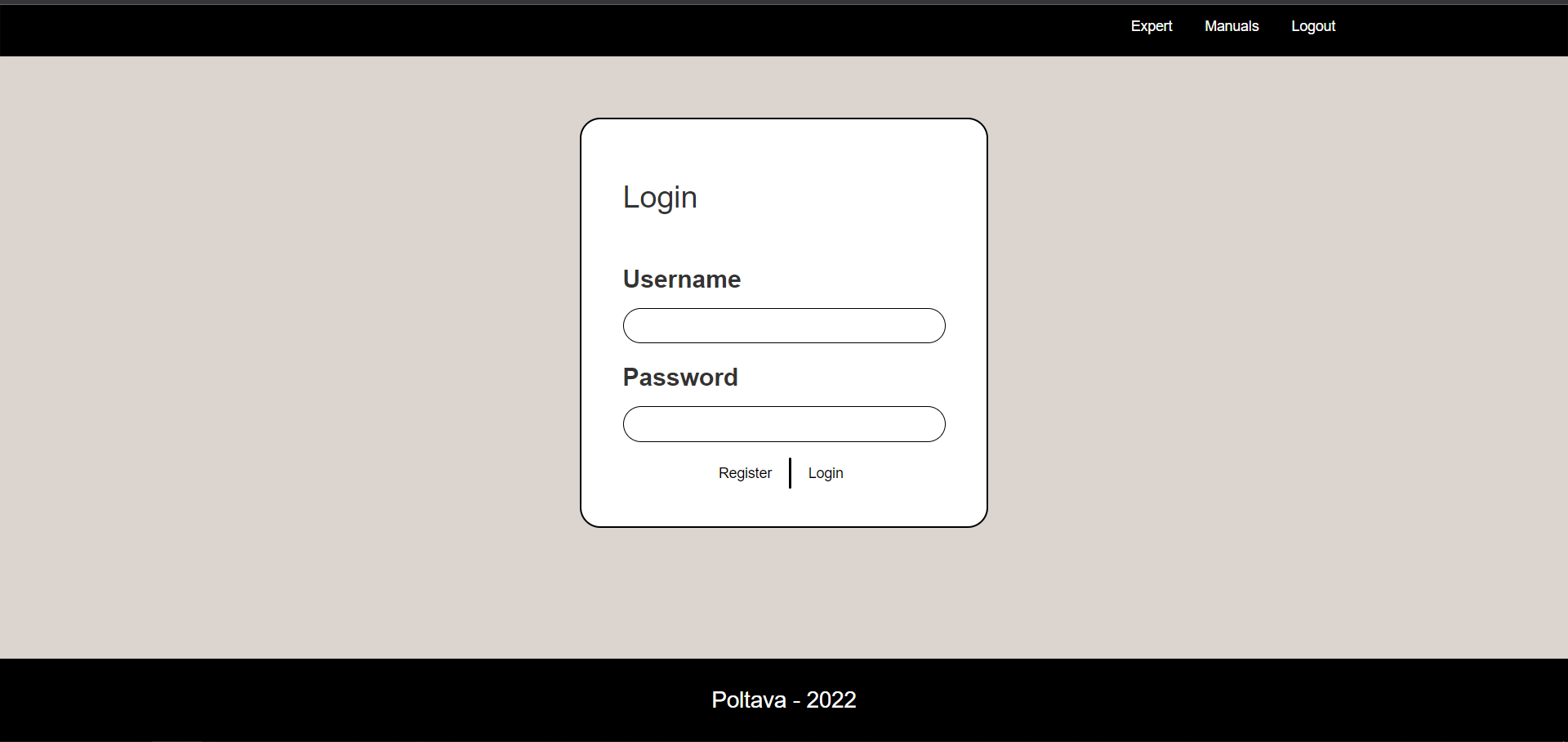


Рисунок 3.11 – Реєстрація нового користувача

Уся інформація про користувачів зберігається в базі даних у таблиці «users»(рис. 3.12).

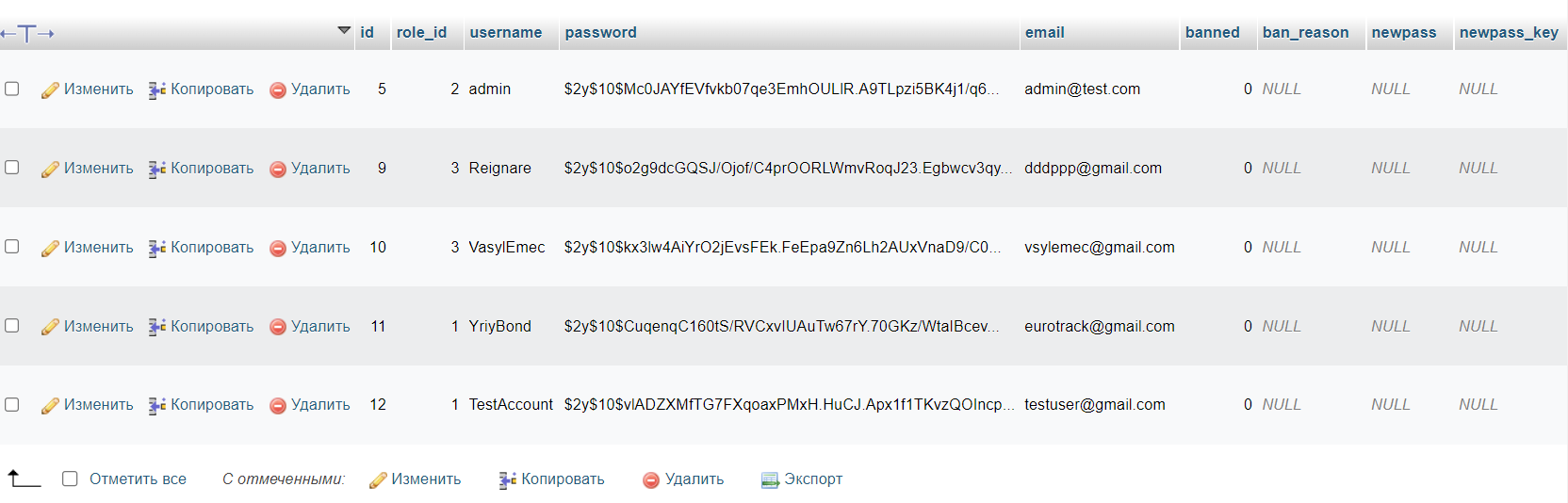


Рисунок 3.12 – Таблиця користувачів

За для міркувань безпеки пароль користувача хешується та в базу даних надсилається вже його шифр. Шифрування проводиться функцією «password\_hash()» із використанням алгоритму «Blowfish» (рис. 3.13);

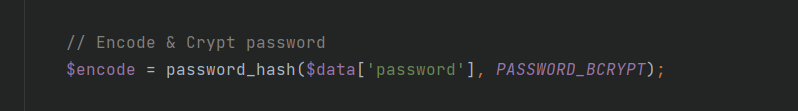


Рисунок 3.13 – Функція хешування паролю

Після етапу реєстрації або авторизації користувач потрапляє на головну сторінку додатку, яка містить в собі пояснення до користування додатком(рис. 3.14.)

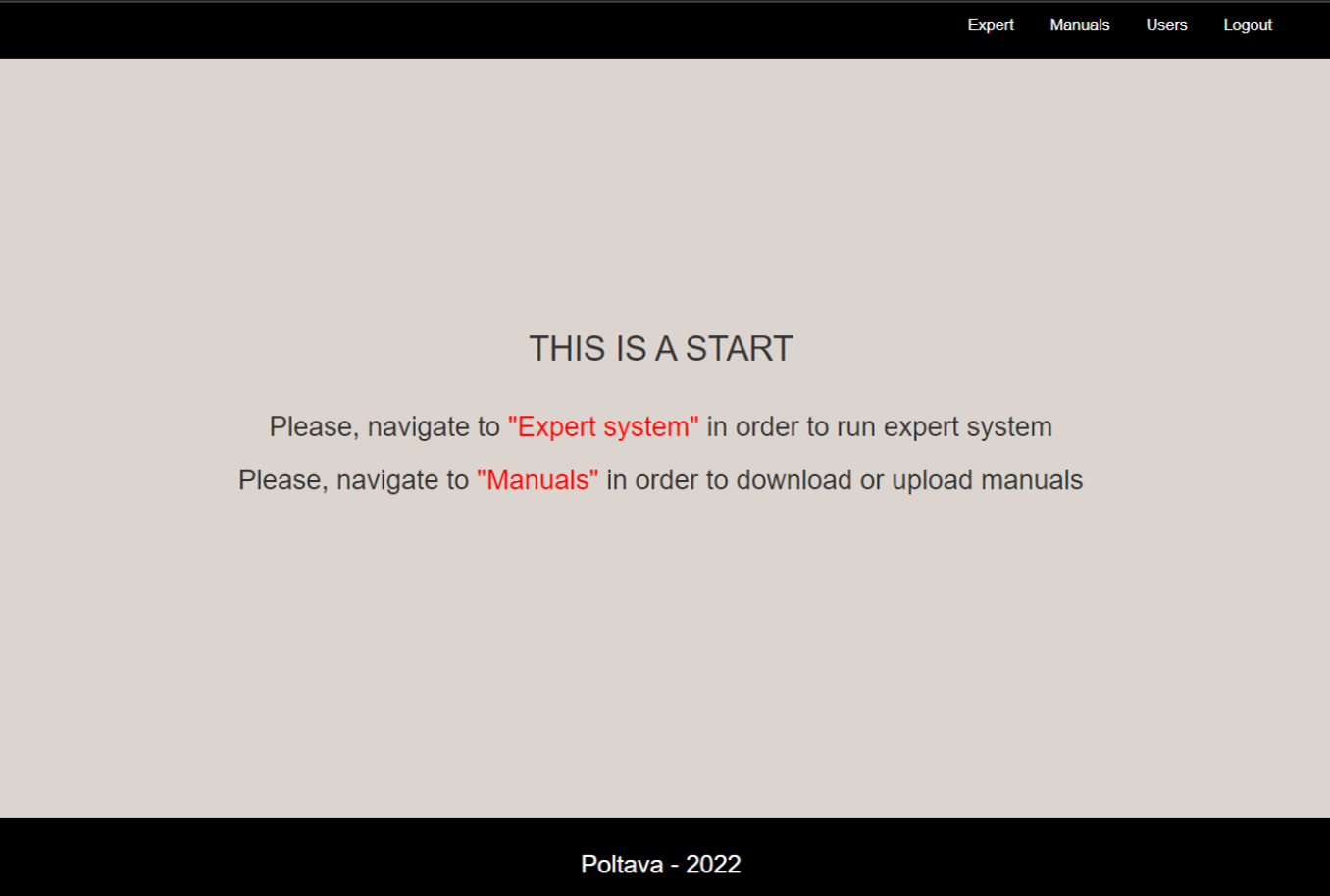


Рисунок 3.14 – Головна сторінка

У випадку, якщо в додаток увійшов акаунт із правами адміністратора, у меню навігації додатку з’явиться посилання до перегляду та управління користувачами(рис. 3.15).

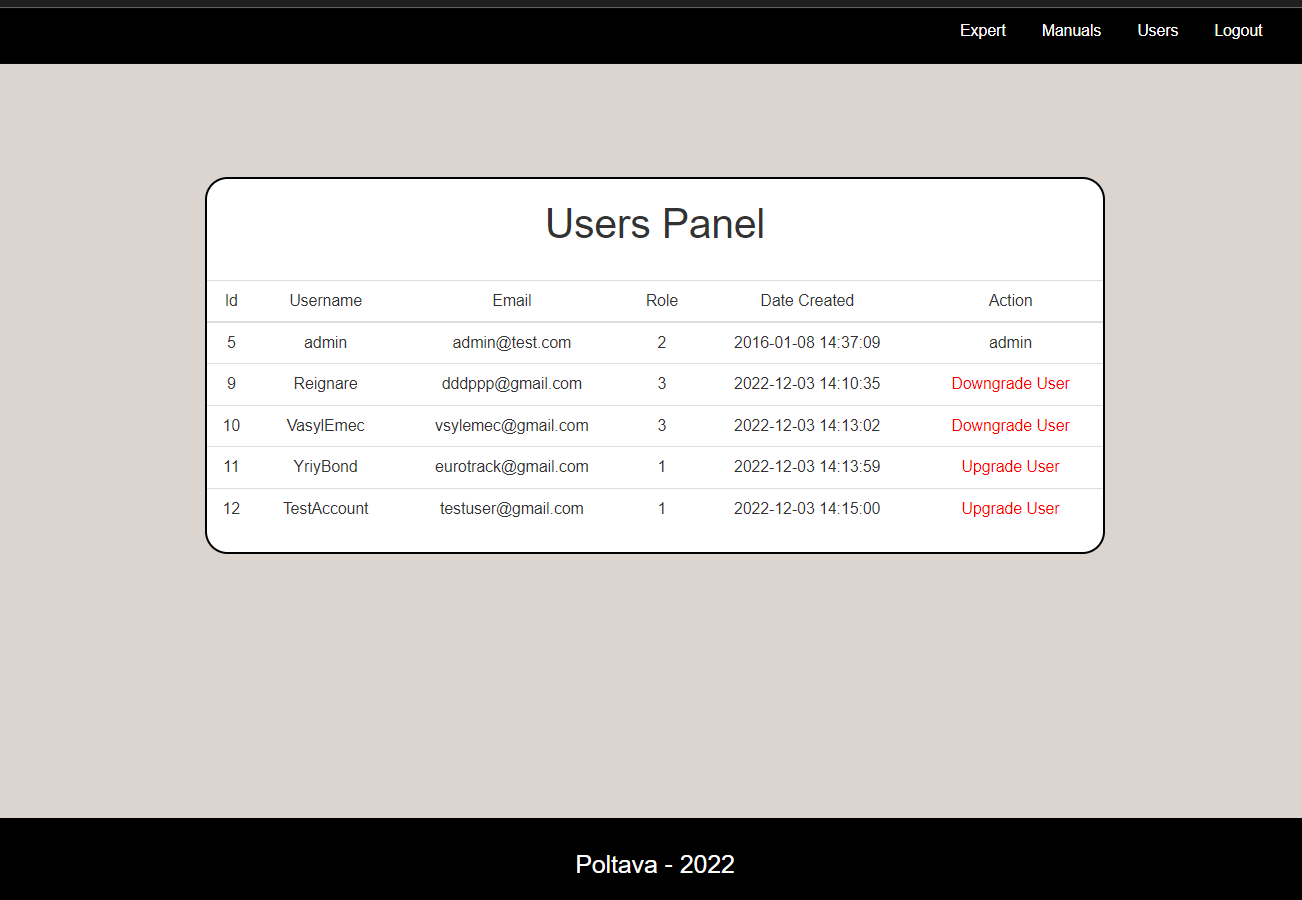


Рисунок 3.15 – Сторінка керування користувачами

Адміністратор ресурсу має можливість надавати або забирати права на завантаження посібників до двигунів, в залежності від поточної ролі користувача.

Розмежування доступом корегується перевірками які відслідковують поточну сесію користувача, витягують з бази даних поле «role\_id» та порівнюють його із значенням прав, які необхідно мати для отримання доступу до контенту(рис. 3.16)

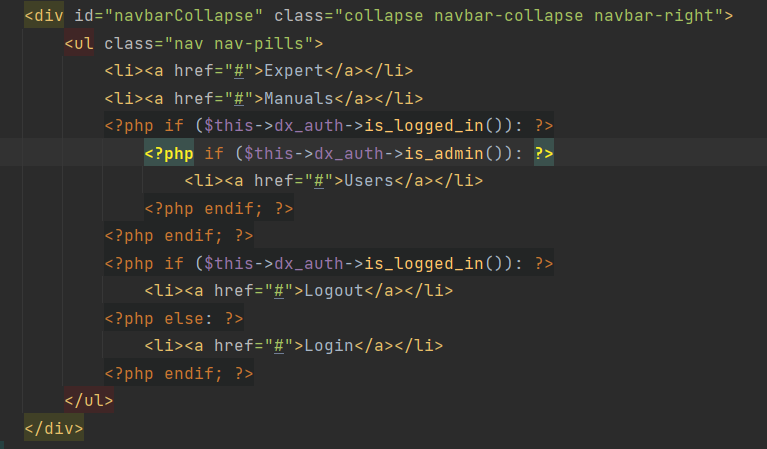


Рисунок 3.16 – Розмежування доступу

Для того щоб перейти до експертної системи, користувач повинен обрати пункт «Expert» в меню навігації. Після цього він перейде на сторінку із вибором типу двигуна. Список також містить рік випуску та моделі автомобілів, на які встановлено даний тип двигуна(рис. 3.17).

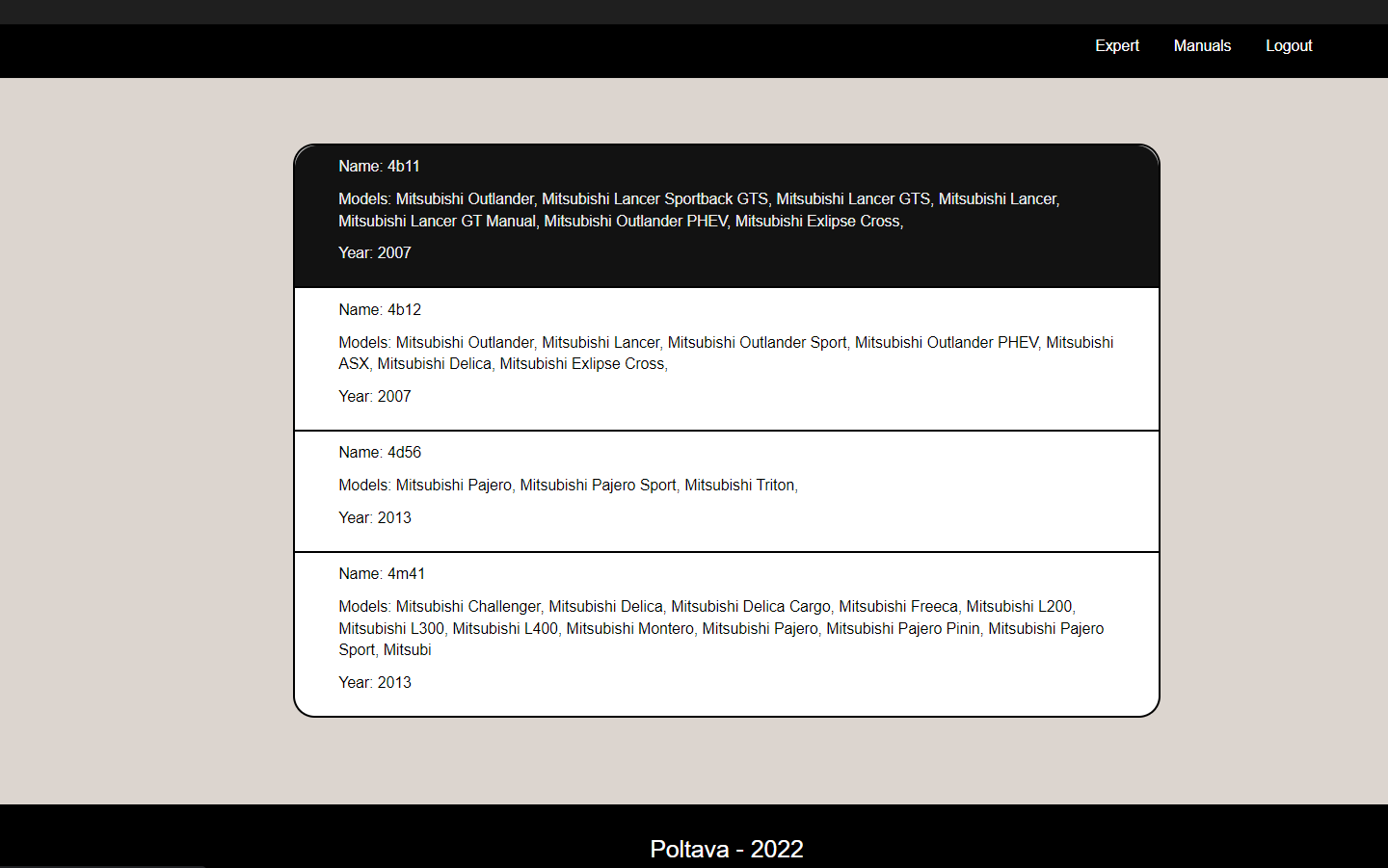


Рисунок 3.17 – Сторінка вибору типу двигуна

Далі, користувач повинен обрати помилку за кодом або назвою для зменшення радіусу поломки(рис. 3.18).

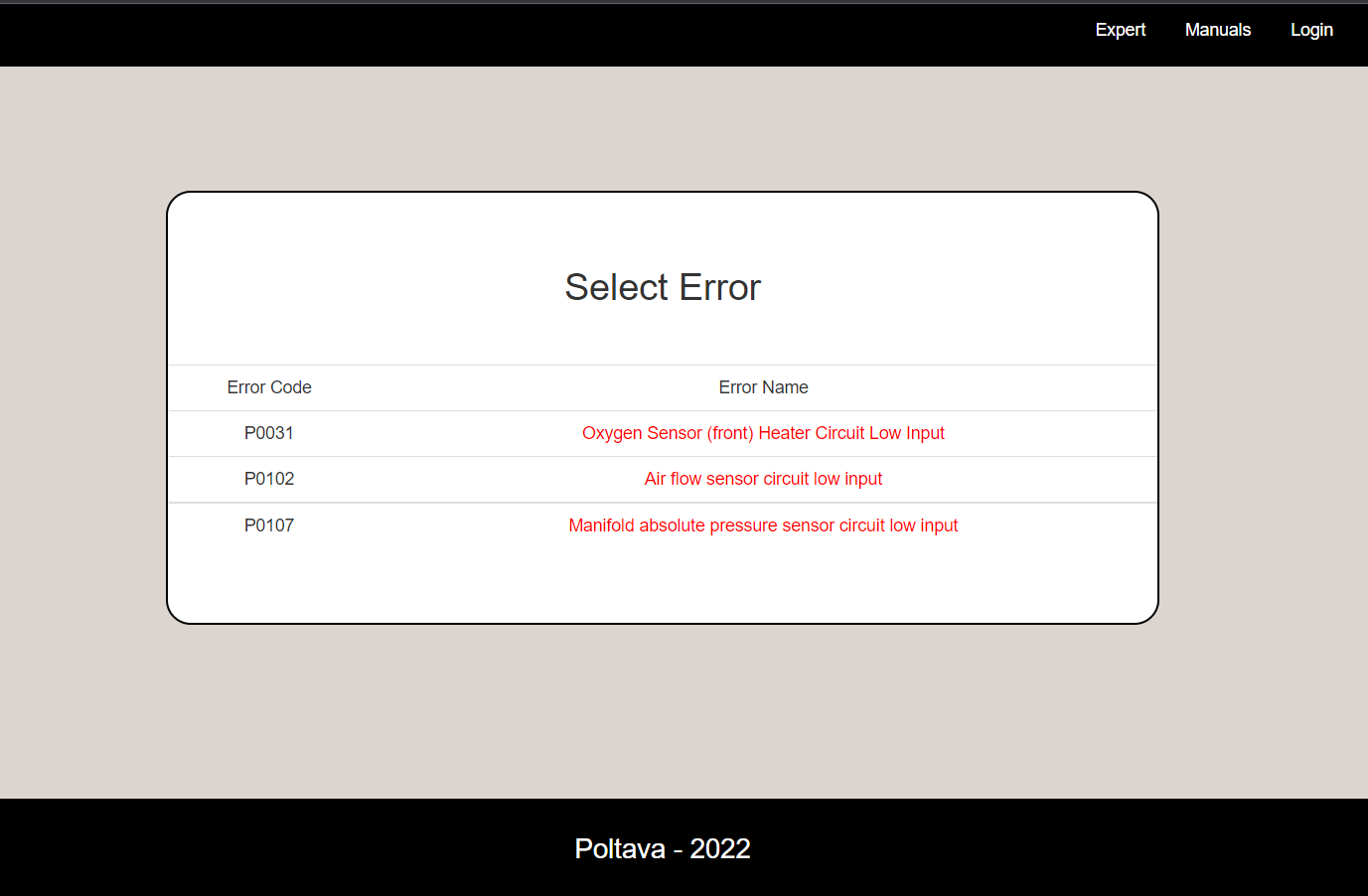


Рисунок 3.18 – Сторінка вибору помилки

Усі помилки були взяті із ресурсу «Mitsubishi motors», звідти також була взята база знань за допомогою якої встановлюється логіка експертної системи.

Сама експертна система працює на базі евристичних методів та дає рішення в залежності від відповідей користувача на поставлені запитання системі. Кількість питань залежить від логіки бази знань для конкретної проблеми.

Кожен наступний крок береться із бази даних із використанням методології «Ajax»(рис. 3.19), для того щоб не перенавантажувати великим об’ємом інформації, яка може не знадобитися без оновлення поточної сторінки, що позитивно впливає на швидкодію додатку.

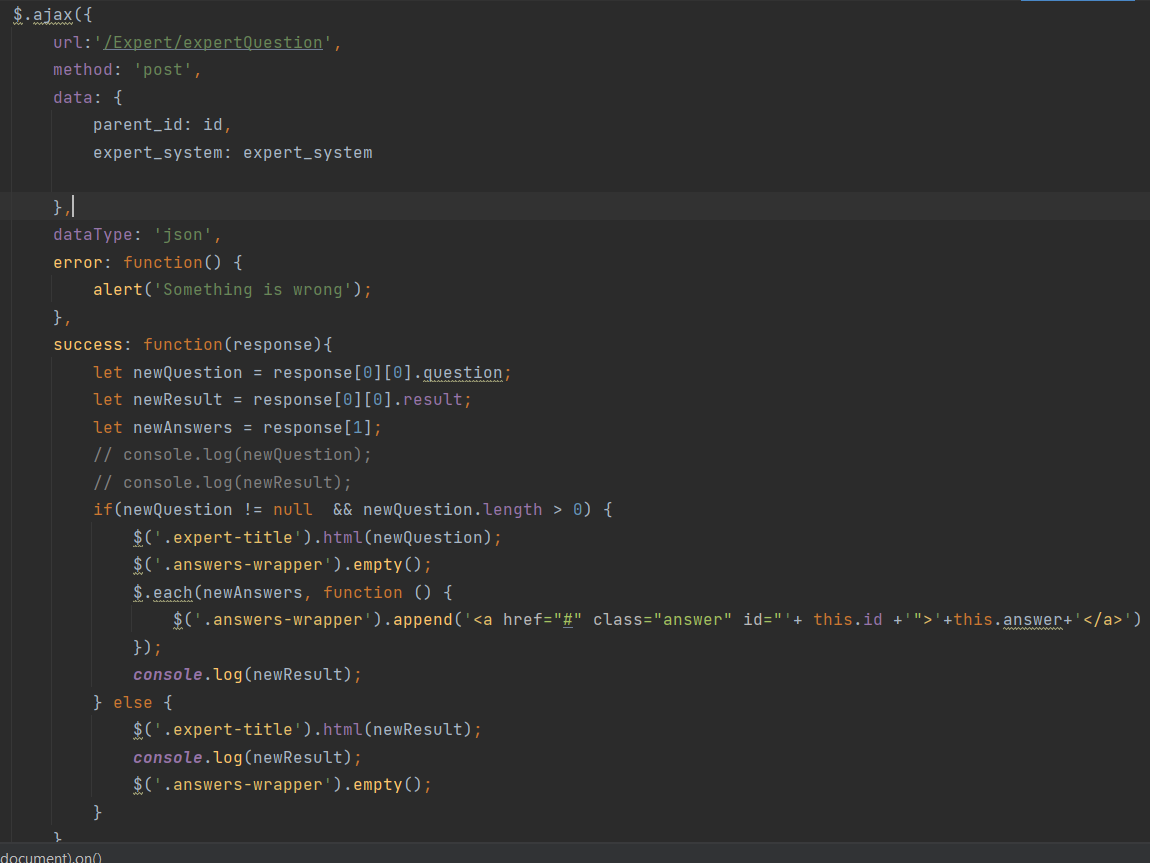


Рисунок 3.19 – Ajax-запит

Нижче представлений повний цикл експертної системи(рис 3.20 – 3.23).



Рисунок 3.20 – Крок експертної системи

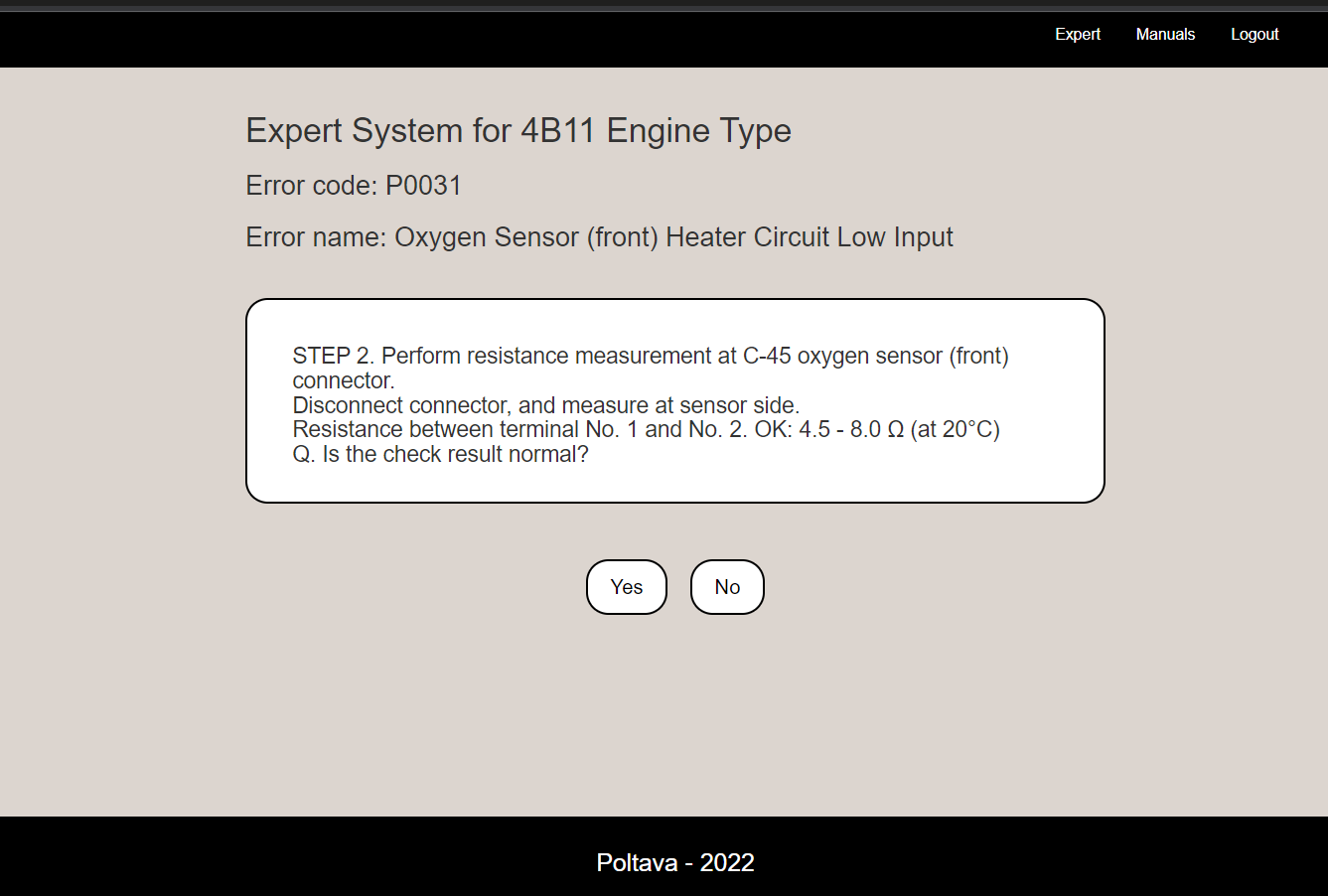


Рисунок 3.21 – Крок експертної системи

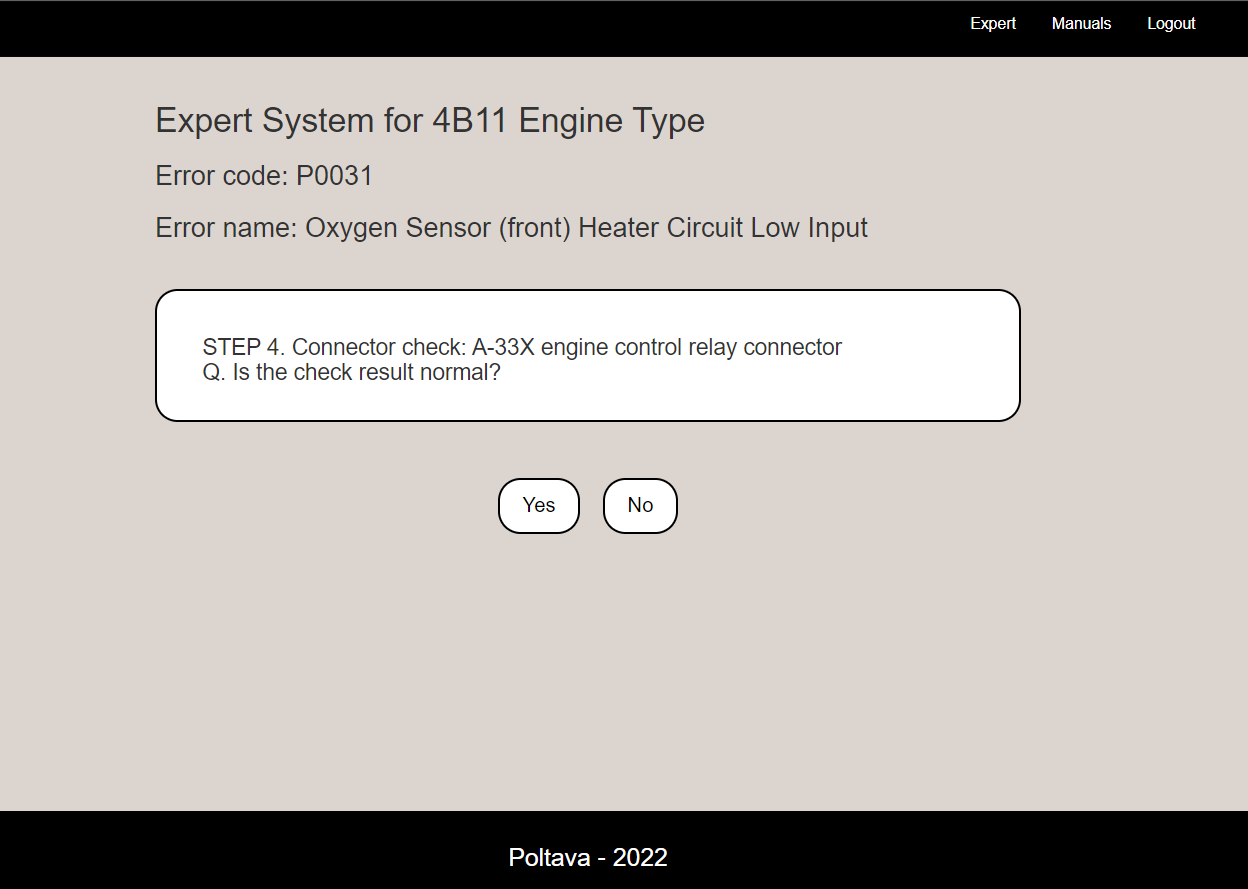


Рисунок 3.22 – Крок експертної системи



Рисунок 3.23 – Ajax-запит

Оскільки база знань була взята із даних «Mitsubishi» в багатьох випадках вона опирається на певні норми та їх кодові імена, які прописані у готових посібниках, тому для цього додано можливість перегляду цих посібників всередині додатку та їх завантаження локально. Для цього, користувачу необхідно перейти по посиланню «Manuals», обрати тип двигуна, який цікавить користувача(рис. 3.24).

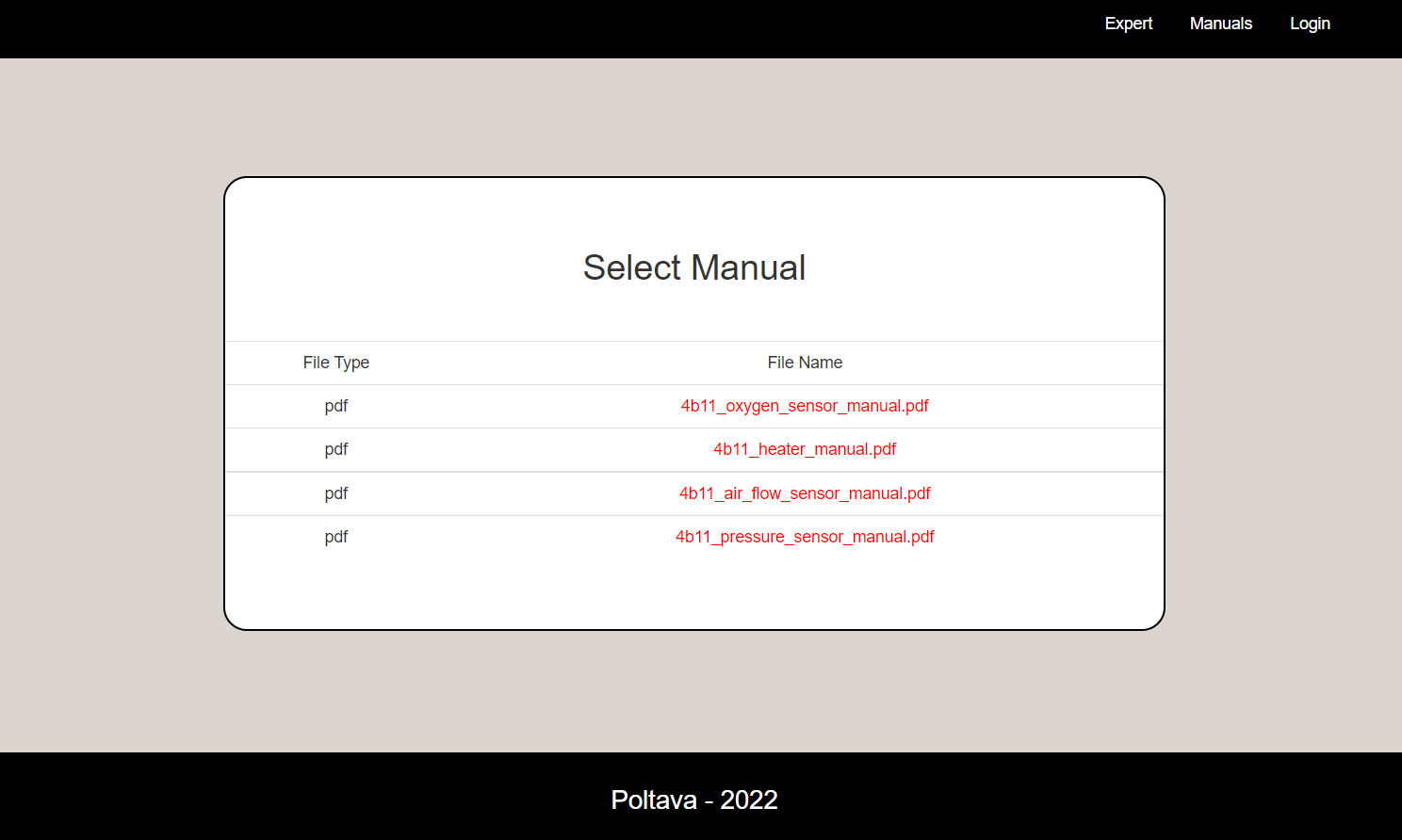


Рисунок 3.24 – Сторінка вибору посібника

Для того щоб завантажити посібник на сервер, необхідно мати права користувача обрати пункт «Upload manual», обрати тип двигуна та обрати файл, який необхідно завантажити(рис. 3.25).

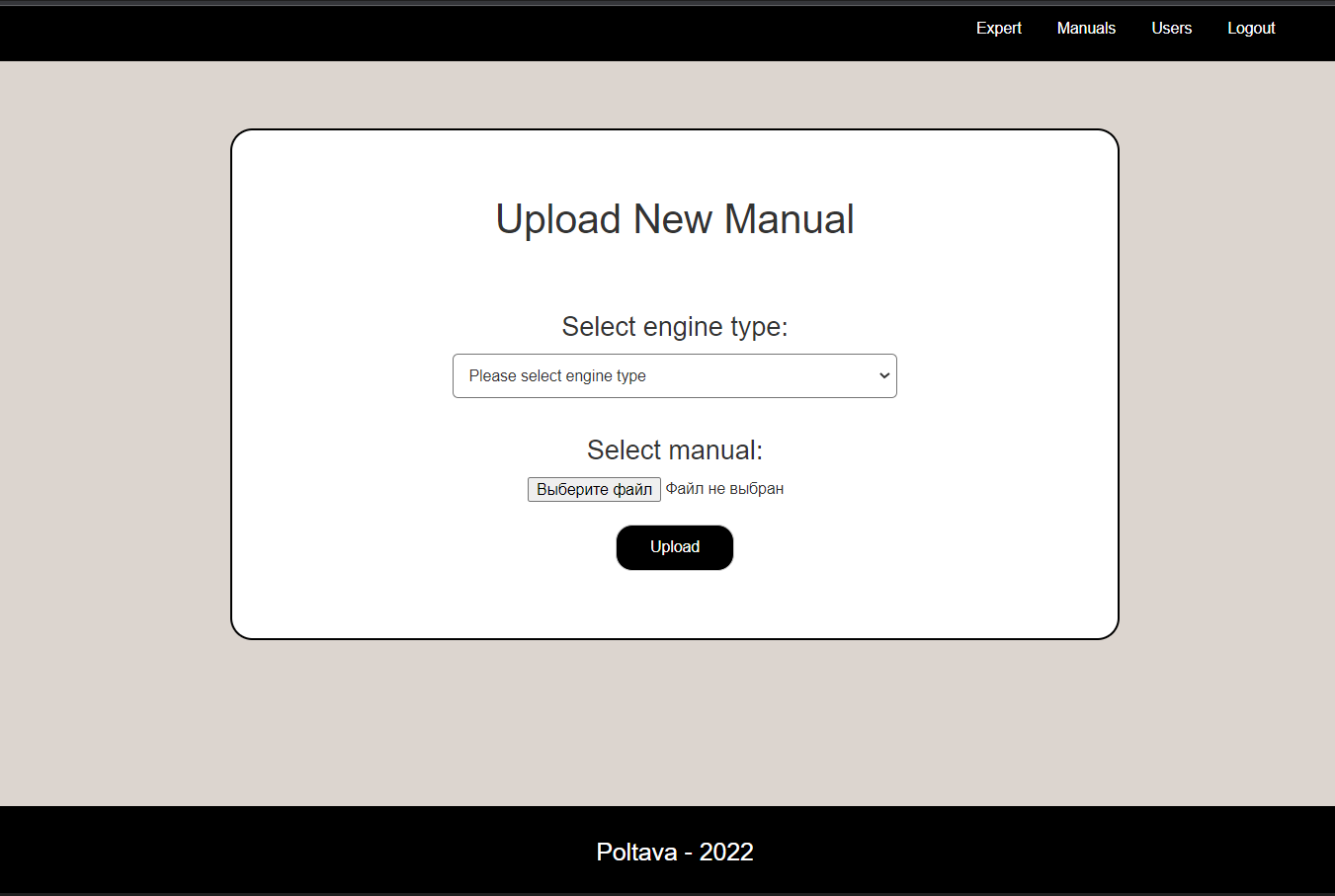


Рисунок 3.25 – Завантаження посібника

**ВИСНОВКИ**

У ході виконання роботи магістра, було спроєктовано та розроблено веб додаток із експертною системою для діагностики несправностей у лінійці автомобілів марки «Mitsubishi». Експертна система дає можливість отримати рішення по ремонту обраної помилки обраного двигуна. Базою знань слугують онлайн посібники у відкритому доступі компанії «Mitsubishi». Додаток містить також систему авторизації та реєстрації, за допомогою якої також проводиться розмежування доступу у додатку. Адміністратор ресурсу має змогу переглядати та змінювати ролі користувачів. Також додаток містить у собі посібники, які відсортовані за типом двигунів. Реалізовано їх перегляд, завантаження, та завантаження нових посібників адміністратором, або користувачами із відповідними правами

Під час розробки додатку були використані сучасні технології та методології, які посприяли швидкодії додатка.

Додаток призначений для компаній, які займаються ремонтом автомобілів лінійки «Mitsubishi».

До основних переваг використання додатку належать:

* можливість покрокового перегляду кожного етапу дій;
* можливість перегляду посібників, відсортованих за типом двигуна.

Розроблена експертна система виконує усі вимоги, які були поставленні на етапі визначення задачі. До вдосконалення системи можна віднести збільшення кількості підтримуваних системою типів двигунів.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Newell, A., Shaw, J. C., & Simon, H. A. (1959, January). Report on a general problem solving program. In IFIP congress (Vol. 256, p. 64).
2. Petrick, S. R. (1971). Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation (pp. 23-25). March.
3. Minksy, M. (1975). A framework for representing knowledge. The psychology of computer vision, 73, 211-277.
4. Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.
5. Aikins, J. S. (1983). Prototypical knowledge for expert systems. Artificial Intelligence, 20, 163–210.  
   Arocha, J. F. & Patel, V. L. (1995). Novice diagnostic reasoning in medicine: Accounting for evidence. The Journal of the Learning Sciences, 4, 355–384.
6. Bennett, J. S. (1985). ROGET: A knowledge-based system for acquiring the conceptual structure of a diagnostic expert system. Journal of Automated Reasoning, 1, 49–74.
7. Баклан І.В. Експертні системи. Навчальний посібник. К.: НАУ, 2012. 132 с.
8. Вергунова І.М.Основи математичного моделювання для аналізу та прогнозу агрономічних процесів: Навч. посіб. для студ. агрономічних спец. К.: НОРА-ПРІНТ. 2016. 146 с.
9. Касьяненко В.О., Старченко Л.В. Моделювання та прогнозування економічних процесів: навч. посіб.для студ. вузів. Суми: Унів. книга. 2017. 184
10. Катренко А.В. Теорія прийняття рішень. Підручник. К.: BHV. 2009. 448с.
11. Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи. Львів: Вид-во Львівської політехніки, 2012. 168 с.
12. Питер Джексон Введение в экспертные системы // Introduction to Expert Systems./ Питер Джексон – 3-е изд. – М.: Вильямс, 2001. – 624 с.
13. Погудина О.К. Информационная система обоснования характеристик высокотехнологичных проектов / О.К. Погудина, Д.В. Божко // 114 Радiоелектроннi i комп'ютернi системи. 2008. – № 3 (30). – 102-108 c.
14. Россиев Д.А. Нейросетевые самообучающиеся экспертные системы в медицине. / Молодые ученые – практическому здравоохранению. / Россиев Д.А. – Красноярск, – 1994. –17 c.
15. Рыбина Г.В. Основы построения интеллектуальных систем: уч. пособ./ Рыбина Г.В. – М.: Финансы и статистика; ИНФРА-М, 2010. –65 c
16. Смірнов О.А. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. – М.: Финансы и статистика; ИНФРА-М, 2010. – 65 c
17. Смолин Д.В. Введение в искусственный интеллект. / Смолин Д.В. – М.: ФИЗМАТЛИТ, 2004. –208 с.
18. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник./ Субботін С. О. — Запоріжжя: ЗНТУ, 2008. — 341 с.
19. Таран Т.А. Штучний інтелект. Теорія і застосування: навч. посіб. / Т.А. Таран, Д.А. Зубов. – Луганськ: Вид-во СНУ ім. В. Даля, 2006. – 240 с.
20. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ: Пер. с англ. В. А. Кондратенко, С. В.Трубицына.—М.: Финансы и статистика, 1990. — 320 с.
21. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
22. Уотермен Д. Руководство по экспертным системам. / Уотермен Д.: Пер. с англ. под ред. В.Л. Стефанюка. — М.: «Мир», 1989: — 388 с.
23. Хайкин С. Нейронные сети: полный курс. / Хайкин С. — М.: Вильямс, 2006. —1104 с.

**ДОДАТОК А  
ВИХІДНИЙ КОД ОСНОВНИХ КОМПОНЕНТІВ**

<?php  
  
defined('BASEPATH') or exit('No direct script access allowed');  
  
class Expert extends CI\_Controller  
{  
  
 public function \_\_construct()  
 {  
 parent::\_\_construct();  
 $this->load->model('expert\_model');  
 }  
  
 public function index()  
 {  
 $data['models'] = $this->expert\_model->getModels();  
 $this->load->view('templates/header', $data);  
 $this->load->view('expert/index');  
 $this->load->view('templates/footer');  
 }  
  
  
 public function expert\_system()  
 {  
 if ( isset($\_GET['model']) ) {  
 $model = $\_GET['model'];  
 $data['expert'] = $this->expert\_model->getQuestions($model);  
 $data['first\_answers'] = $this->expert\_model->getFirstQuestion($model);  
 $data['model'] = $model;  
 $this->load->view('templates/header');  
 $this->load->view('expert/index', $data);  
 $this->load->view('templates/footer');  
 }  
 }  
  
 public function expertQuestion()  
 {  
 $postData = $this->input->post();  
 $data = $this->expert\_model->getNextQuestion($postData);  
 echo json\_encode($data);  
 }  
}

<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class Backend extends CI\_Controller  
{  
 function \_\_construct()  
 {  
 parent::\_\_construct();  
   
 $this->load->library('Table');  
 $this->load->library('Pagination');  
 $this->load->library('DX\_Auth');  
   
 $this->load->helper('form');  
 $this->load->helper('url');  
   
 // Protect entire controller so only admin,   
 // and users that have granted role in permissions table can access it.  
 $this->dx\_auth->check\_uri\_permissions();  
 }  
   
 function index()  
 {  
 $this->users();  
 }  
   
 function users()  
 {  
 $this->load->model('dx\_auth/users', 'users');   
   
 // Search checkbox in post array  
 foreach ($\_POST as $key => $value)  
 {  
 // If checkbox found  
 if (substr($key, 0, 9) == 'checkbox\_')  
 {  
 // If ban button pressed  
 if (isset($\_POST['ban']))  
 {  
 // Ban user based on checkbox value (id)  
 $this->users->ban\_user($value);  
 }  
 // If unban button pressed  
 else if (isset($\_POST['unban']))  
 {  
 // Unban user  
 $this->users->unban\_user($value);  
 }  
 else if (isset($\_POST['reset\_pass']))  
 {  
 // Set default message  
 $data['reset\_message'] = 'Reset password failed';  
   
 // Get user and check if User ID exist  
 if ($query = $this->users->get\_user\_by\_id($value) AND $query->num\_rows() == 1)  
 {   
 // Get user record   
 $user = $query->row();  
   
 // Create new key, password and send email to user  
 if ($this->dx\_auth->forgot\_password($user->username))  
 {  
 // Query once again, because the database is updated after calling forgot\_password.  
 $query = $this->users->get\_user\_by\_id($value);  
 // Get user record  
 $user = $query->row();  
   
 // Reset the password  
 if ($this->dx\_auth->reset\_password($user->username, $user->newpass\_key))  
 {   
 $data['reset\_message'] = 'Reset password success';  
 }  
 }  
 }  
 }  
 }   
 }  
   
 /\* Showing page to user \*/  
   
 // Get offset and limit for page viewing  
 $offset = (int) $this->uri->segment(3);  
 // Number of record showing per page  
 $row\_count = 10;  
   
 // Get all users  
 $data['users'] = $this->users->get\_all($offset, $row\_count)->result();  
   
 // Pagination config  
 $p\_config['base\_url'] = '/backend/users/';  
 $p\_config['uri\_segment'] = 3;  
 $p\_config['num\_links'] = 2;  
 $p\_config['total\_rows'] = $this->users->get\_all()->num\_rows();  
 $p\_config['per\_page'] = $row\_count;  
   
 // Init pagination  
 $this->pagination->initialize($p\_config);   
 // Create pagination links  
 $data['pagination'] = $this->pagination->create\_links();  
   
 // Load view  
 $this->load->view('backend/users', $data);  
 }  
   
 function unactivated\_users()  
 {  
 $this->load->model('dx\_auth/user\_temp', 'user\_temp');  
   
 /\* Database related \*/  
   
 // If activate button pressed  
 if ($this->input->post('activate'))  
 {  
 // Search checkbox in post array  
 foreach ($\_POST as $key => $value)  
 {  
 // If checkbox found  
 if (substr($key, 0, 9) == 'checkbox\_')  
 {  
 // Check if user exist, $value is username  
 if ($query = $this->user\_temp->get\_login($value) AND $query->num\_rows() == 1)  
 {  
 // Activate user  
 $this->dx\_auth->activate($value, $query->row()->activation\_key);  
 }  
 }   
 }  
 }  
   
 /\* Showing page to user \*/  
   
 // Get offset and limit for page viewing  
 $offset = (int) $this->uri->segment(3);  
 // Number of record showing per page  
 $row\_count = 10;  
   
 // Get all unactivated users  
 $data['users'] = $this->user\_temp->get\_all($offset, $row\_count)->result();  
   
 // Pagination config  
 $p\_config['base\_url'] = '/backend/unactivated\_users/';  
 $p\_config['uri\_segment'] = 3;  
 $p\_config['num\_links'] = 2;  
 $p\_config['total\_rows'] = $this->user\_temp->get\_all()->num\_rows();  
 $p\_config['per\_page'] = $row\_count;  
   
 // Init pagination  
 $this->pagination->initialize($p\_config);   
 // Create pagination links  
 $data['pagination'] = $this->pagination->create\_links();  
   
 // Load view  
 $this->load->view('backend/unactivated\_users', $data);  
 }  
   
 function roles()  
 {   
 $this->load->model('dx\_auth/roles', 'roles');  
   
 /\* Database related \*/  
   
 // If Add role button pressed  
 if ($this->input->post('add'))  
 {  
 // Create role  
 $this->roles->create\_role($this->input->post('role\_name'), $this->input->post('role\_parent'));  
 }  
 else if ($this->input->post('delete'))  
 {   
 // Loop trough $\_POST array and delete checked checkbox  
 foreach ($\_POST as $key => $value)  
 {  
 // If checkbox found  
 if (substr($key, 0, 9) == 'checkbox\_')  
 {  
 // Delete role  
 $this->roles->delete\_role($value);  
 }   
 }  
 }  
  
 /\* Showing page to user \*/  
   
 // Get all roles from database  
 $data['roles'] = $this->roles->get\_all()->result();  
   
 // Load view  
 $this->load->view('backend/roles', $data);  
 }  
   
 function uri\_permissions()  
 {  
 function trim\_value(&$value)   
 {   
 $value = trim($value);   
 }  
   
 $this->load->model('dx\_auth/roles', 'roles');  
 $this->load->model('dx\_auth/permissions', 'permissions');  
   
 if ($this->input->post('save'))  
 {  
 // Convert back text area into array to be stored in permission data  
 $allowed\_uris = explode("\n", $this->input->post('allowed\_uris'));  
   
 // Remove white space if available  
 array\_walk($allowed\_uris, 'trim\_value');  
   
 // Set URI permission data  
 // IMPORTANT: uri permission data, is saved using 'uri' as key.  
 // So this key name is preserved, if you want to use custom permission use other key.  
 $this->permissions->set\_permission\_value($this->input->post('role'), 'uri', $allowed\_uris);  
 }  
   
 /\* Showing page to user \*/   
   
 // Default role\_id that will be showed  
 $role\_id = $this->input->post('role') ? $this->input->post('role') : 1;  
   
 // Get all role from database  
 $data['roles'] = $this->roles->get\_all()->result();  
 // Get allowed uri permissions  
 $data['allowed\_uris'] = $this->permissions->get\_permission\_value($role\_id, 'uri');  
   
 // Load view  
 $this->load->view('backend/uri\_permissions', $data);  
 }  
   
 function custom\_permissions()  
 {  
 // Load models  
 $this->load->model('dx\_auth/roles', 'roles');  
 $this->load->model('dx\_auth/permissions', 'permissions');  
   
 /\* Get post input and apply it to database \*/  
   
 // If button save pressed  
 if ($this->input->post('save'))  
 {  
 // Note: Since in this case we want to insert two key with each value at once,  
 // it's not advisable using set\_permission\_value() function   
 // If you calling that function twice that means, you will query database 4 times,  
 // because set\_permission\_value() will access table 2 times,   
 // one for get previous permission and the other one is to save it.  
   
 // For this case (or you need to insert few key with each value at once)   
 // Use the example below  
   
 // Get role\_id permission data first.   
 // So the previously set permission array key won't be overwritten with new array with key $key only,   
 // when calling set\_permission\_data later.  
 $permission\_data = $this->permissions->get\_permission\_data($this->input->post('role'));  
   
 // Set value in permission data array  
 $permission\_data['edit'] = $this->input->post('edit');  
 $permission\_data['delete'] = $this->input->post('delete');  
   
 // Set permission data for role\_id  
 $this->permissions->set\_permission\_data($this->input->post('role'), $permission\_data);  
 }  
   
 /\* Showing page to user \*/   
   
 // Default role\_id that will be showed  
 $role\_id = $this->input->post('role') ? $this->input->post('role') : 1;  
   
 // Get all role from database  
 $data['roles'] = $this->roles->get\_all()->result();  
 // Get edit and delete permissions  
 $data['edit'] = $this->permissions->get\_permission\_value($role\_id, 'edit');  
 $data['delete'] = $this->permissions->get\_permission\_value($role\_id, 'delete');  
   
 // Load view  
 $this->load->view('backend/custom\_permissions', $data);  
 }  
}

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');  
/\*\*  
 \* Files management class created by CodexWorld  
 \*/  
class Manuals extends CI\_Controller {  
  
 function \_\_construct() {  
 parent::\_\_construct();  
 $this->load->model('manuals\_model');  
 $this->load->helper('form');  
 }  
  
 public function index($section=null){  
// $this->data['files'] = $this->file->getRows($section);  
// $this->data['news'] = $this->news\_model->getNews();  
  
 //load the view  
 $this->data['title'] = "доповіді";  
 $this->load->view('templates/header', $this->data);  
 $this->load->view('manuals');  
 $this->load->view('templates/footer');  
  
 }  
  
 public function view($slug=NULL){  
 $this->data['title'] = "Доповідь";  
 $this->data['files\_item'] = $this->file->getFiles($slug);  
 $files\_slug = $this->file->getFiles($slug);  
 $files = $this->file->getFiles();  
 $data['files'] = $this->file->getRows($slug);  
 $this->data['news'] = $this->news\_model->getNews();  
  
  
 if(empty($this->data['files\_item'])){  
 show\_404();  
 }  
 $this->data['id'] = $files\_slug['id'];  
 $this->load->model('comments\_model');  
 $this->data['comments'] = $this->comments\_model->getComments($files\_slug['id'],100);  
  
 $files['files'] = $this->file->getFiles($slug);  
 $this->data['content'] = $this->data['files\_item']['file\_name'];  
 $this->load->view('templates/header', $this->data);  
 $this->load->view('files/view', $files);  
 $this->load->view('templates/footer');  
 }  
  
  
 public function load($slug){  
 if(!empty($slug)){  
 //load download helper  
 $this->load->helper('download');  
  
 //get file info from database  
 $fileinfo = $this->file->download($slug);  
  
 //file path  
 $file = 'uploads/'.$fileinfo['file\_name'];  
  
 //download file from directory  
 force\_download($file, NULL);  
 }  
 }  
}

<?php  
  
  
class Upload extends CI\_Controller  
{  
  
 public function \_\_construct()  
 {  
 parent::\_\_construct();  
 $this->load->helper(array('form', 'url'));  
 $this->load->database();  
 $this->load->library('form\_validation');  
 $this->load->helper('form');  
 $this->load->model('manuals\_model');  
  
 }  
  
 public function index()  
 {  
 $this->data['news'] = $this->manuals\_model->getFiles();  
 $this->data['title'] = "додати доповідь";  
 $this->load->view('templates/header', $this->data);  
 $this->load->view('manuals', $this->data);  
 $this->load->view('templates/footer');  
  
  
 }  
  
 public function do\_upload()  
 {  
 $config['upload\_path'] = './uploads/';  
 $config['allowed\_types'] = 'doc|docx|pdf';  
 $config['max\_size'] = 10000;  
  
 $this->load->library('upload', $config);  
  
 if (!$this->upload->do\_upload('userfile')) {  
 $error = array('error' => $this->upload->display\_errors());  
  
 $this->load->view('files/upload\_file', $error);  
 } else {  
 $uniqid = uniqid();  
 $data = array(  
 'manual\_title' => $this->upload->data('file\_name'),  
 'manual\_path' => $this->upload->data('full\_path'),  
 'slug' => $uniqid,  
 );  
 return $this->db->insert('manuals', $data);  
 $this->load->view('templates/header');  
 $this->load->view('files/upload\_success', $data);  
 $this->load->view('templates/footer');  
 }  
  
 $this->load->view('templates/header', $this->data);  
 $this->load->view('files/upload\_success', $data);  
 $this->load->view('templates/footer');  
  
 }  
}  
?>

<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class Auth extends CI\_Controller  
{  
 // Used for registering and changing password form validation  
 var $min\_username = 4;  
 var $max\_username = 20;  
 var $min\_password = 4;  
 var $max\_password = 20;  
  
 function \_\_construct()  
 {  
 parent::\_\_construct();  
   
 $this->load->library('Form\_validation');  
 $this->load->library('DX\_Auth');  
   
 $this->load->helper('url');  
 $this->load->helper('form');  
 }  
   
 function index()  
 {  
 $this->login();  
 }  
   
 /\* Callback function \*/  
   
 function username\_check($username)  
 {  
 $result = $this->dx\_auth->is\_username\_available($username);  
 if ( ! $result)  
 {  
 $this->form\_validation->set\_message('username\_check', 'Username already exist. Please choose another username.');  
 }  
   
 return $result;  
 }  
  
 function email\_check($email)  
 {  
 $result = $this->dx\_auth->is\_email\_available($email);  
 if ( ! $result)  
 {  
 $this->form\_validation->set\_message('email\_check', 'Email is already used by another user. Please choose another email address.');  
 }  
   
 return $result;  
 }  
  
   
 function recaptcha\_check()  
 {  
 $result = $this->dx\_auth->is\_recaptcha\_match();   
 if ( ! $result)  
 {  
 $this->form\_validation->set\_message('recaptcha\_check', 'Your confirmation code does not match the one in the image. Try again.');  
 }  
   
 return $result;  
 }  
   
 /\* End of Callback function \*/  
   
   
 function login()  
 {  
 if ( ! $this->dx\_auth->is\_logged\_in())  
 {  
 $val = $this->form\_validation;  
   
 // Set form validation rules  
 $val->set\_rules('username', 'Username', 'trim|required');  
 $val->set\_rules('password', 'Password', 'trim|required');  
 $val->set\_rules('remember', 'Remember me', 'integer');  
  
   
 if ($val->run() AND $this->dx\_auth->login($val->set\_value('username'), $val->set\_value('password'), $val->set\_value('remember')))  
 {  
 // Redirect to homepage  
 redirect('', 'location');  
 }  
 else  
 {  
 // Check if the user is failed logged in because user is banned user or not  
 if ($this->dx\_auth->is\_banned())  
 {  
 // Redirect to banned uri  
 $this->dx\_auth->deny\_access('banned');  
 }  
 else  
 {   
 // Default is we don't show captcha until max login attempts eceeded  
 $data['show\_captcha'] = FALSE;  
   
 // Load login page view  
 $this->load->view('templates/header');  
 $this->load->view($this->dx\_auth->login\_view, $data);  
 $this->load->view('templates/footer');  
  
 }  
 }  
 }  
 else  
 {  
 $data['auth\_message'] = 'You are already logged in.';  
 $this->load->view($this->dx\_auth->logged\_in\_view, $data);  
 }  
 }  
   
 function logout()  
 {  
 $this->dx\_auth->logout();  
   
 $data['auth\_message'] = 'You have been logged out.';   
 $this->load->view($this->dx\_auth->logout\_view, $data);  
 }  
   
   
 function register()  
 {  
 if ( ! $this->dx\_auth->is\_logged\_in() AND $this->dx\_auth->allow\_registration)  
 {   
 $val = $this->form\_validation;  
   
 // Set form validation rules  
 $val->set\_rules('username', 'Username', 'trim|required|min\_length['.$this->min\_username.']|max\_length['.$this->max\_username.']|callback\_username\_check|alpha\_dash');  
 $val->set\_rules('password', 'Password', 'trim|required|min\_length['.$this->min\_password.']|max\_length['.$this->max\_password.']|matches[confirm\_password]');  
 $val->set\_rules('confirm\_password', 'Confirm Password', 'trim|required');  
 $val->set\_rules('email', 'Email', 'trim|required|valid\_email|callback\_email\_check');  
   
 // Is registration using captcha  
 if ($this->dx\_auth->captcha\_registration)  
 {  
 // Set recaptcha rules.  
 // IMPORTANT: Do not change 'recaptcha\_response\_field' because it's used by reCAPTCHA API,  
 // This is because the limitation of reCAPTCHA, not DX Auth library  
 $val->set\_rules('recaptcha\_response\_field', 'Confirmation Code', 'trim|required|callback\_recaptcha\_check');  
 }  
  
 // Run form validation and register user if it's pass the validation  
 if ($val->run() AND $this->dx\_auth->register($val->set\_value('username'), $val->set\_value('password'), $val->set\_value('email')))  
 {   
 // Set success message accordingly  
 if ($this->dx\_auth->email\_activation)  
 {  
 $data['auth\_message'] = 'You have successfully registered. Check your email address to activate your account.';  
 }  
 else  
 {   
 $data['auth\_message'] = 'You have successfully registered. '.anchor(site\_url($this->dx\_auth->login\_uri), 'Login');  
 }  
   
 // Load registration success page  
 $this->load->view($this->dx\_auth->register\_success\_view, $data);  
 }  
 else  
 {  
 // Load registration page  
 $this->load->view('templates/header');  
 $this->load->view('Auth/register\_form');  
 $this->load->view('templates/footer');  
 }  
 }  
 elseif ( ! $this->dx\_auth->allow\_registration)  
 {  
 $data['auth\_message'] = 'Registration has been disabled.';  
 $this->load->view($this->dx\_auth->register\_disabled\_view, $data);  
 }  
 else  
 {  
 $data['auth\_message'] = 'You have to logout first, before registering.';  
 $this->load->view($this->dx\_auth->logged\_in\_view, $data);  
 }  
 }  
   
 function activate()  
 {  
 // Get username and key  
 $username = $this->uri->segment(3);  
 $key = $this->uri->segment(4);  
  
 // Activate user  
 if ($this->dx\_auth->activate($username, $key))   
 {  
 $data['auth\_message'] = 'Your account have been successfully activated. '.anchor(site\_url($this->dx\_auth->login\_uri), 'Login');  
 $this->load->view($this->dx\_auth->activate\_success\_view, $data);  
 }  
 else  
 {  
 $data['auth\_message'] = 'The activation code you entered was incorrect. Please check your email again.';  
 $this->load->view($this->dx\_auth->activate\_failed\_view, $data);  
 }  
 }  
   
 function forgot\_password()  
 {  
 $val = $this->form\_validation;  
   
 // Set form validation rules  
 $val->set\_rules('login', 'Username or Email address', 'trim|required');  
  
 // Validate rules and call forgot password function  
 if ($val->run() AND $this->dx\_auth->forgot\_password($val->set\_value('login')))  
 {  
 $data['auth\_message'] = 'An email has been sent to your email with instructions with how to activate your new password.';  
 $this->load->view($this->dx\_auth->forgot\_password\_success\_view, $data);  
 }  
 else  
 {  
 $this->load->view($this->dx\_auth->forgot\_password\_view);  
 }  
 }  
   
 function reset\_password()  
 {  
 // Get username and key  
 $username = $this->uri->segment(3);  
 $key = $this->uri->segment(4);  
  
 // Reset password  
 if ($this->dx\_auth->reset\_password($username, $key))  
 {  
 $data['auth\_message'] = 'You have successfully reset you password, '.anchor(site\_url($this->dx\_auth->login\_uri), 'Login');  
 $this->load->view($this->dx\_auth->reset\_password\_success\_view, $data);  
 }  
 else  
 {  
 $data['auth\_message'] = 'Reset failed. Your username and key are incorrect. Please check your email again and follow the instructions.';  
 $this->load->view($this->dx\_auth->reset\_password\_failed\_view, $data);  
 }  
 }  
   
 function change\_password()  
 {  
 // Check if user logged in or not  
 if ($this->dx\_auth->is\_logged\_in())  
 {   
 $val = $this->form\_validation;  
   
 // Set form validation  
 $val->set\_rules('old\_password', 'Old Password', 'trim|required|min\_length['.$this->min\_password.']|max\_length['.$this->max\_password.']');  
 $val->set\_rules('new\_password', 'New Password', 'trim|required|min\_length['.$this->min\_password.']|max\_length['.$this->max\_password.']|matches[confirm\_new\_password]');  
 $val->set\_rules('confirm\_new\_password', 'Confirm new Password', 'trim|required');  
   
 // Validate rules and change password  
 if ($val->run() AND $this->dx\_auth->change\_password($val->set\_value('old\_password'), $val->set\_value('new\_password')))  
 {  
 $data['auth\_message'] = 'Your password has successfully been changed.';  
 $this->load->view($this->dx\_auth->change\_password\_success\_view, $data);  
 }  
 else  
 {  
 $this->load->view($this->dx\_auth->change\_password\_view);  
 }  
 }  
 else  
 {  
 // Redirect to login page  
 $this->dx\_auth->deny\_access('login');  
 }  
 }   
   
 function cancel\_account()  
 {  
 // Check if user logged in or not  
 if ($this->dx\_auth->is\_logged\_in())  
 {   
 $val = $this->form\_validation;  
   
 // Set form validation rules  
 $val->set\_rules('password', 'Password', "trim|required");  
   
 // Validate rules and change password  
 if ($val->run() AND $this->dx\_auth->cancel\_account($val->set\_value('password')))  
 {  
 // Redirect to homepage  
 redirect('', 'location');  
 }  
 else  
 {  
 $this->load->view($this->dx\_auth->cancel\_account\_view);  
 }  
 }  
 else  
 {  
 // Redirect to login page  
 $this->dx\_auth->deny\_access('login');  
 }  
 }  
  
 // Example how to get permissions you set permission in /backend/custom\_permissions/  
 function custom\_permissions()  
 {  
 if ($this->dx\_auth->is\_logged\_in())  
 {  
 echo 'My role: '.$this->dx\_auth->get\_role\_name().'<br/>';  
 echo 'My permission: <br/>';  
   
 if ($this->dx\_auth->get\_permission\_value('edit') != NULL AND $this->dx\_auth->get\_permission\_value('edit'))  
 {  
 echo 'Edit is allowed';  
 }  
 else  
 {  
 echo 'Edit is not allowed';  
 }  
   
 echo '<br/>';  
   
 if ($this->dx\_auth->get\_permission\_value('delete') != NULL AND $this->dx\_auth->get\_permission\_value('delete'))  
 {  
 echo 'Delete is allowed';  
 }  
 else  
 {  
 echo 'Delete is not allowed';  
 }  
 }  
 }  
  
  
}

<?php  
  
  
class Users extends CI\_Controller  
{  
  
 public function \_\_construct()  
 {  
 parent::\_\_construct();  
 $this->load->helper(array('form', 'url'));  
 $this->load->model('users\_model');  
  
 }  
  
 public function index()  
 {  
 $data['users'] = $this->users\_model->getUsers();  
 $this->load->view('templates/header', );  
 $this->load->view('users', $data);  
 $this->load->view('templates/footer');  
 }  
  
 public function update\_user()  
 {  
 $user = $this->uri->segment(3);  
 $role = $this->uri->segment(4);  
 $this->users\_model->upUser($user, $role);  
 $data['users'] = $this->users\_model->getUsers();  
 $this->load->view('templates/header', );  
 $this->load->view('users', $data);  
 $this->load->view('templates/footer');  
 }  
}

<?php  
  
class Expert\_model extends CI\_Model {  
 public function \_\_construct()  
 {  
 $this->load->database();  
 }  
 public function getQuestions ($model) {  
 $query = $this->db->get($model);  
 return $query -> result\_array();  
 }  
  
 public function getModels()  
 {  
 $query = $this->db->get('engine\_model');  
 return $query->result\_array();  
 }  
  
 function getFirstQuestion($model){  
  
 $this->db->select('\*');  
 $this->db->where('id', 1);  
 $records = $this->db->get($model);  
 $question = $records->result\_array();  
  
 $this->db->select('\*');  
 $this->db->where('parent\_id', 1);  
 $records = $this->db->get($model);  
 $ansers = $records->result\_array();  
  
 return array($question, $ansers);  
 }  
  
 function getNextQuestion($postData){  
  
 $this->db->select('\*');  
 $this->db->where('id', $postData['parent\_id']);  
 $records = $this->db->get($postData['expert\_system']);  
 $question = $records->result\_array();  
// $question = str\_replace('\n', '<br>', $question);  
  
 $this->db->select('\*');  
 $this->db->where('parent\_id', $postData['parent\_id']);  
 $records = $this->db->get($postData['expert\_system']);  
 $ansers = $records->result\_array();  
  
 return array($question, $ansers);  
 }  
}

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');  
class Manuals\_model extends CI\_Model{  
 /\*  
 \* get rows from the files table  
 \*/  
 public function \_\_construct(){  
  
 $this->load->database();  
 }  
  
 function getRows($section=null,$params = array()){  
 $this->db->select('\*');  
 $this->db->from('manuals');  
 if(array\_key\_exists('id',$params) && !empty($params['id'])){  
 $this->db->where('id',$params['id']);  
 //get records  
 $query = $this->db->get();  
 $result = ($query->num\_rows() > 0)?$query->row\_array():FALSE;  
 }else{  
 //set start and limit  
 if(array\_key\_exists("start",$params) && array\_key\_exists("limit",$params)){  
 $this->db->limit($params['limit'],$params['start']);  
 }elseif(!array\_key\_exists("start",$params) && array\_key\_exists("limit",$params)){  
 $this->db->limit($params['limit']);  
 }  
 //get records  
 $query = $this->db->get();  
 $result = ($query->num\_rows() > 0)?$query->result\_array():FALSE;  
 }  
 //return fetched data  
 return $result;  
 }  
 public function getFiles($slug=FALSE){  
 if($slug===FALSE){  
 $query = $this->db->get('manuals');  
 return $query->result\_array();  
 }  
 $query = $this->db->get\_where('files', array('slug'=> $slug));  
 return $query->row\_array();  
 }  
  
  
 public function download($slug){  
 $query = $this->db->get\_where('files',array('slug'=>$slug));  
 return $query->row\_array();  
 }  
}

<?php  
  
class Expert\_model extends CI\_Model {  
 public function \_\_construct()  
 {  
 $this->load->database();  
 }  
 public function getQuestions ($model) {  
 $query = $this->db->get($model);  
 return $query -> result\_array();  
 }  
  
 public function getModels()  
 {  
 $query = $this->db->get('engine\_model');  
 return $query->result\_array();  
 }  
  
 function getFirstQuestion($model){  
  
 $this->db->select('\*');  
 $this->db->where('id', 1);  
 $records = $this->db->get($model);  
 $question = $records->result\_array();  
  
 $this->db->select('\*');  
 $this->db->where('parent\_id', 1);  
 $records = $this->db->get($model);  
 $ansers = $records->result\_array();  
  
 return array($question, $ansers);  
 }  
  
 function getNextQuestion($postData){  
  
 $this->db->select('\*');  
 $this->db->where('id', $postData['parent\_id']);  
 $records = $this->db->get($postData['expert\_system']);  
 $question = $records->result\_array();  
// $question = str\_replace('\n', '<br>', $question);  
  
 $this->db->select('\*');  
 $this->db->where('parent\_id', $postData['parent\_id']);  
 $records = $this->db->get($postData['expert\_system']);  
 $ansers = $records->result\_array();  
  
 return array($question, $ansers);  
 }  
}

<?php  
$username = array(  
 'name' => 'username',  
 'id' => 'username',  
 'size' => 30,  
 'value' => set\_value('username')  
);  
  
$password = array(  
 'name' => 'password',  
 'id' => 'password',  
 'size' => 30,  
 'value' => set\_value('password')  
);  
  
$confirm\_password = array(  
 'name' => 'confirm\_password',  
 'id' => 'confirm\_password',  
 'size' => 30,  
 'value' => set\_value('confirm\_password')  
);  
  
$email = array(  
 'name' => 'email',  
 'id' => 'email',  
 'maxlength' => 80,  
 'size' => 30,  
 'value' => set\_value('email')  
);  
?>  
  
  
<section class="login">  
 <div class="login-wrapper">  
 <h2>Register</h2>  
 <?php echo form\_open($this->uri->uri\_string())?>  
 <br>  
 <?php echo $this->dx\_auth->get\_auth\_error(); ?>  
 <div class="username">  
 <h3><?php echo form\_label('Username', $username['id']);?></h3>  
 <?php echo form\_input($username)?>  
 <?php echo form\_error($username['name']); ?>  
 </div>  
 <div class="password">  
 <h3><?php echo form\_label('Password', $password['id']);?></h3>  
 <?php echo form\_password($password)?>  
 <?php echo form\_error($password['name']); ?>  
 </div>  
 <div class="password-confirm">  
 <h3><?php echo form\_label('Confirm Password', $confirm\_password['id']);?></h3>  
 <?php echo form\_password($confirm\_password);?>  
 <?php echo form\_error($confirm\_password['name']); ?>  
 </div>  
 <div class="email">  
 <h3><?php echo form\_label('Email Address', $email['id']);?></h3>  
 <?php echo form\_input($email);?>  
 <?php echo form\_error($email['name']); ?>  
 </div>  
 <div class="lower-nav">  
 <?php echo form\_submit('register','Register', 'class="register-link"');?>  
 <?php echo anchor($this->dx\_auth->login\_uri, 'Login', 'class="login-submit"'); ?>  
 <?php  
 if ($this->dx\_auth->allow\_registration) {  
  
 };  
 ?>  
 </div>  
 <!-- --><?php //echo anchor($this->dx\_auth->forgot\_password\_uri, 'Forgot password', 'class="forgot-link"');?>  
 </div>  
</section>  
<?php echo form\_close()?>  
</fieldset>

<?php  
$username = array(  
 'name' => 'username',  
 'id' => 'username',  
 'size' => 30,  
 'value' => set\_value('username')  
);  
  
$password = array(  
 'name' => 'password',  
 'id' => 'password',  
 'size' => 30  
);  
  
$remember = array(  
 'name' => 'remember',  
 'id' => 'remember',  
 'value' => 1,  
 'checked' => set\_value('remember'),  
 'style' => 'margin:0;padding:0'  
);  
  
$confirmation\_code = array(  
 'name' => 'captcha',  
 'id' => 'captcha',  
 'maxlength' => 8  
);  
  
?>  
<section class="login">  
 <div class="login-wrapper">  
 <h2>Login</h2>  
 <?php echo form\_open($this->uri->uri\_string())?>  
 <br>  
 <?php echo $this->dx\_auth->get\_auth\_error(); ?>  
 <div class="username">  
 <h3><?php echo form\_label('Username', $username['id']);?></h3>  
 <?php echo form\_input($username)?>  
 <?php echo form\_error($username['name']); ?>  
 </div>  
 <div class="password">  
 <h3><?php echo form\_label('Password', $password['id']);?></h3>  
 <?php echo form\_password($password)?>  
 <?php echo form\_error($password['name']); ?>  
 </div>  
 <div class="lower-nav">  
 <?php  
 if ($this->dx\_auth->allow\_registration) {  
 echo anchor($this->dx\_auth->register\_uri, 'Register', 'class="register-link"');  
 };  
 ?>  
 <?php echo form\_submit('login','Login', 'class="login-submit"');?>  
 </div>  
<!-- --><?php //echo anchor($this->dx\_auth->forgot\_password\_uri, 'Forgot password', 'class="forgot-link"');?>  
 </div>  
</section>  
<!-- --><?php //echo form\_checkbox($remember);?><!-- --><?php //echo form\_label('Remember me', $remember['id']);?><!-- -->  
<?php echo form\_close()?>  
</fieldset>

<section class="model-list-layout">  
 <div class="container">  
 <div class="engine-type-layout">  
 <ul>  
 <?php foreach ( $models as $model): ?>  
 <a href="/expert/expert\_system?model=<?php echo 'expert\_'.strtolower($model['engine\_type']).'\_'.strtolower($model['model\_year']); ?>">  
 <li>  
 <p>  
 <span>Name:</span> <?php echo $model['engine\_type']; ?>  
 </p>  
 <p>  
 <span>Models:</span> <?php echo $model['models\_used']; ?>  
 </p>  
 <p>  
 <span>Year:</span> <?php echo $model['model\_year']; ?>  
 </p>  
 </li>  
 </a>  
 <?php endforeach; ?>  
 </ul>  
 </div>  
 </div>  
</section>

<div class="container">  
 <div class="users-layout">  
 <table class="table">  
 <tbody>  
 <h1>Users Panel</h1>  
 </br>  
 <tr>  
 <!-- <td>Id</td> -->  
 <td>Id</td>  
 <td>Username</td>  
 <td>Email</td>  
 <td>Role</td>  
 <td>Date Created</td>  
 <td>Action</td>  
 </tr>  
 <?php foreach ($users as $user): ?>  
 <tr>  
 <td><?php echo $user['id']; ?></td>  
 <td><?php echo $user['username']; ?></td>  
 <td><?php echo $user['email']; ?></td>  
 <td><?php echo $user['role\_id']; ?></td>  
 <td><?php echo $user['created']; ?></td>  
 <td>  
  
 <?php if ($user['username'] == 'admin'): ?>  
 admin  
 <?php elseif ($user['role\_id'] == 3 || $user['role\_id'] == 2): ?>  
 <a href="/users/update\_user/<?php echo $user['id']; ?>/1">Downgrade User</a>  
 <?php else: ?>  
 <a href="/users/update\_user/<?php echo $user['id']; ?>/3">Upgrade User</a>  
 <?php endif; ?>  
 </td>  
  
 </tr>  
 <?php endforeach; ?>  
 </tbody>  
 </table>  
 </div>  
</div>

$active\_group = 'default';  
$query\_builder = TRUE;  
  
$db['default'] = array(  
 'dsn' => '',  
 'hostname' => 'localhost',  
 'username' => 'root',  
 'password' => '',  
 'database' => 'diplom',  
 'dbdriver' => 'mysqli',  
 'dbprefix' => '',  
 'pconnect' => FALSE,  
 'db\_debug' => (ENVIRONMENT !== 'production'),  
 'cache\_on' => FALSE,  
 'cachedir' => '',  
 'char\_set' => 'utf8',  
 'dbcollat' => 'utf8\_general\_ci',  
 'swap\_pre' => '',  
 'encrypt' => FALSE,  
 'compress' => FALSE,  
 'stricton' => FALSE,  
 'failover' => array(),  
 'save\_queries' => TRUE  
);