# AI-driven web applications

### A PROJECT REPORT

***Submitted by***

## Moradiya Keyurkumar

### (ET21BTAI043)

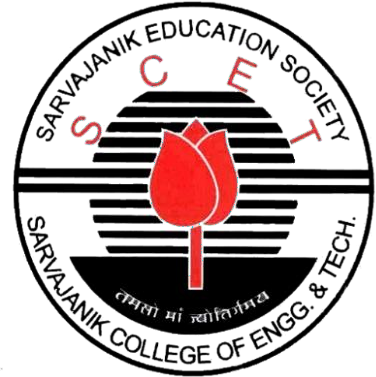
***In fulfillment for the award of the degree of***

## BACHELOR OF TECHNOLOGY

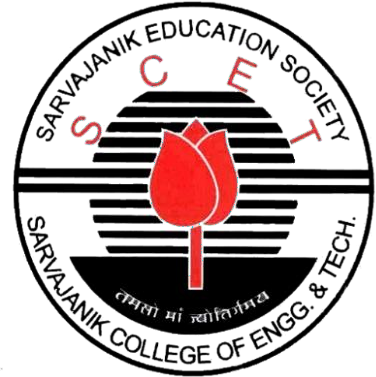
***in***

### ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

**SARVAJANIK COLLEGE OF ENGINEERING AND TECHNOLOGY, SURAT**

****

## Sarvajanik University, Surat May - 2025

### Sarvajanik College of Engineering and Technology, Surat

Dr. R K Desai Marg, Opp. Mission Hospital, Athwalines, Athwa, Surat, Gujarat 395001

# CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Internship** has been carried out by **Moradiya Keyurkumar - ET21BTAI043** under my guidance in partial fulfillment for the degree of Bachelor of Technology in ***Department of Artificial Intelligence and Data Science***, 8th Semester of Sarvajanik University, Surat during the academic year 2024-25.

Prof. Nitya Komalan

#### Internal Guide

Prof. (Dr.) Vivaksha Jariwala

#### Head of the Department

**<LETTER HEAD OF COMPANY>**

Date: 18/04/2025

## TO WHOM IT MAY CONCERN

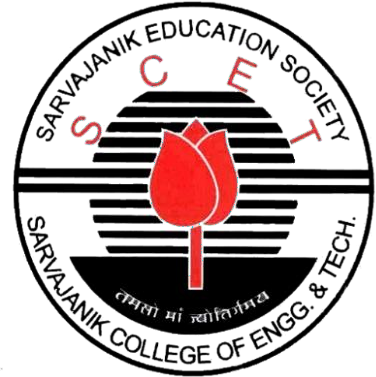
This is to certify that **Moradiya Keyurkumar**, a student of ***Sarvajanik College of Engineering and Technology, Surat*** has successfully completed his internship in the field of **AI Engineer** from **15-01-2025** to **15-04-2025** (Total number of Weeks: 13) under the guidance of **Brijesh Kanani**.

During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in his life and career.

For **Hyeon Infotech,**

Authorized Signature with Industry Stamp

### Sarvajanik College of Engineering and Technology, Surat

Dr. R K Desai Marg, Opp. Mission Hospital, Athwalines, Athwa, Surat, Gujarat 395001

# DECLARATION

We hereby declare that the Internship report submitted along with the Internship

entitled **AI-driven Web Applications Report** submitted in partial fulfillment for the degree of Bachelor of Technology in **Artificial Intelligence and Data Science** to Sarvajanik University, Surat, is a Bonafide record of original project work carried out by me at ***Hyeon Infotech*** under the supervision of Prof. ***Nitya Komalan*** and that no part of this report has been directly copied from any students’ reports or taken from any other source, without providing due reference.

|  |  |  |
| --- | --- | --- |
| Name of the Student | Enrolment No. | Sign of Student |

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Hyeon Infotech for providing me with the opportunity to undertake this internship. I am deeply grateful to my mentor, Brijesh Kanani , for his invaluable guidance, support, and encouragement throughout the internship period. His expertise and insights significantly contributed to my learning and development as a AI Engineer.

I would also like to thank the entire team at Hyeon Infotech for their cooperation and assistance, which made my internship experience both productive and enjoyable.

Furthermore, I extend my appreciation to Sarvajanik College of Engineering and Technology, and my internal guide, Prof. Nitya Komalan for their support and for facilitating this internship opportunity

#### With Sincere Regards

Moradiya Keyurkumar

ET21BTAI043

**ABSTRACT**

# AI Engineer Internship

During my internship at Hyeon Infotech, I was involved in a variety of projects focused on the development and deployment of AI-driven web applications. My work spanned both front-end and back-end development, utilizing technologies such as HTML, CSS, JavaScript, React.js, Python, Flask, and MongoDB. A significant portion of my time was dedicated to developing a plant disease detection system, which involved studying deep learning concepts, CNN architectures, and pre-trained models. I also contributed to projects involving machine learning, such as car selling price prediction and customer segmentation, and explored natural language processing techniques, including sentiment analysis. Furthermore, I gained experience in deploying applications using platforms like Render, Railway, and Vercel, and explored emerging areas like LangChain and AI agents, culminating in the development of a health AI agent chatbot. This internship provided me with practical experience in applying AI and web development skills to real-world problems, enhancing my understanding of the software development lifecycle and industry best practices.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Description** | **Page No** |
| Fig 4.1 | Client-Server interaction using API gateway | 10 |
| Fig 4.2 | PCA Streamlit UI | 11 |
| Fig 4.3 | basic User schema used in MongoDB | 12 |
| Fig 4.4 | PlantAid Use Case Diagram | 15 |
| Fig 4.5 | Youtube Comment Sentiment Analysis Action Diagram | 15 |
| Fig 5.1 | PlantAid Login/Signup Page | 18 |
| Fig 5.2 | PlantAid Home Page | 18 |
| Fig 5.3 | PlantAid Chatbot | 19 |
| Fig 5.4 | Watchinsight Login/Signup Page | 19 |
| Fig 5.5 | Watchinsight Home Page | 19 |
| Fig 5.6 | Watchinsight News Page | 20 |
| Fig 5.7 | Add URL Page | 20 |
| Fig 5.8 | PlantAid Analysis Page | 21 |
| Fig 5.9 | Watchinsight Price Analysis and Price Checker Page | 22 |
| Fig 5.10 | Bar Graph Of Comments | 22 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Description** | **Page No** |
| Table 2.1 | Timeline Table | 6 |
| Table 5.1 | PlantAid Accuracy | 21 |
| Table 5.2 | Watchinsight Accuracy | 22 |
| Table 6.1 | PlantAid Project Test Table | 23 |
| Table 6.2 | Watchinsight Project Test Table | 24 |
| Table 6.3 | Youtube Comment Sentiment Analysis Test Table | 25 |

**TABLE OF CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No** |  | **Chapter Title** | **Page No** |
|  | *Title Page* | | - |
|  | *Certificate (Signed by Head and Institute Mentor)* | | - |
|  | *Certificate from Company / Research Guide* | | - |
|  | Declaration | | i |
|  | Acknowledgement | | ii |
|  | Abstract | | iii |
|  | List of Figures | | iv |
|  | List of Tables | | v |
|  | List of Abbreviations | | vi |
|  | Table of Contents | | vii |
| **Section A:** | **Internship / Project Tracking** | | - |
|  | **A.1** | Bi-Weekly Diaries | - |
|  | **A.2** | Supervisor Feedback Form (Form-4: Evaluation of Intern / Researcher) | - |
| **Chapter 1** | **Overview of the Company/Research Organization** | |  |
|  | 1.1 | Introduction to Company/Research Institute | 1 |
|  | 1.2 | Area of Work (Web, Cloud, Server, AI, etc.) | 1 |
|  | 1.3 | Objectives of the Internship | 1 |
|  | 1.4 | Relevance to IT/AI&DS | 2 |
| **Chapter 2** | **Problem Statement & Objectives** | |  |
|  | 2.1 | Overview of Assigned Project / Research | 3 |
|  | 2.2 | Problem Statement | 3 |
|  | 2.3 | Goals & Learning Outcomes | 4 |
|  | 2.4 | Scope and Limitations | 4 |
|  | 2.5 | Timeline Chart (Gantt or Weekly Plan) | 6 |
| **Chapter 3** | **Technologies & Tools Used** | |  |
|  | 3.1 | Tools, Frameworks & Platforms | 7 |
|  | 3.2 | Methodologies (Agile, DevOps, Research Approach) | 8 |
|  | 3.3 | Justification for Technology Selection | 9 |
| **Chapter 4** | **System / Project Design** | |  |
|  | 4.1 | System / Solution Design (UI, Backend, Architecture) | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4.2 | Database / Data Model / Algorithm Design | 12 |
|  | 4.3 | UML Diagrams (Use Case, Class, Activity, etc.) | 14 |
| **Chapter 5** | **Implementation** | |  |
|  | 5.1 | Implementation Process (Industry or Research) | 16 |
|  | 5.2 | Key Modules / Functions / Experiments | 17 |
|  | 5.3 | Snapshots / Screens / Code (if allowed) | 18 |
|  | 5.4 | Results / Outputs | 20 |
| **Chapter 6** | **Testing & Validation** | |  |
|  | 6.1 | Testing Approach (Unit, Integration, UAT) | 23 |
|  | 6.2 | Sample Test Cases or Research Validation | 23 |
|  | 6.3 | Result Analysis | 26 |
| **Chapter 7** | **Conclusion and Discussion** | |  |
|  | 7.1 | Overall Analysis of Internship / Project Viabilities | 27 |
|  | 7.2 | Skills Acquired | 27 |
|  | 7.3 | Dates of Continuous Evaluation (CE-I and CE-II) | 28 |
|  | 7.4 | Problem Encountered and Possible Solutions | 28 |
|  | 7.5 | Summary of Internship / Project work | 28 |
|  | 7.6 | Limitation and Future Enhancement | 29 |

**CHAPTER 1: OVERVIEW OF THE COMPANY**

### Introduction to Company/Research Institute

Hyeon Infotech is a dynamic and growing IT solutions provider based in Surat, Gujarat, India. Established with the goal of delivering modern and intelligent technology solutions, the company focuses on delivering custom software and AI-driven solutions tailored to the needs of businesses. Hyeon Infotech is committed to innovation and aims to empower companies through technology, making processes smarter, faster, and more efficient. The company operates from its headquarters at Silver Business Point, VIP Circle, Uttran, Surat, and has made a notable presence in the tech space with a strong emphasis on quality and client satisfaction.

### Area of Work

Hyeon Infotech operates in multiple key domains of the IT industry, including:

* + - Web Development
    - Mobile App Design
    - UI/UX Consulting
    - AI Services
    - IOS Development

### Objectives of the Internship

The primary objectives of this internship at Hyeon Infotech were:

* + - To gain practical, hands-on experience in the field of AI Engineering.
    - To apply classroom knowledge of AI and related technologies to real-world projects.
    - To develop proficiency in relevant programming languages and tools.
    - To contribute to the development and deployment of AI-powered applications.
    - To understand the software development lifecycle within a company setting.

### Relevance to IT/AI&DS

This internship was highly relevant to my Bachelor of Technology degree in Information Technology/Artificial Intelligence and Data Science. The work directly involved the application of AI principles and the use of various software development tools and techniques that are central to my degree program. Specifically, the projects I worked on involved machine learning, deep learning, and web development, all of which are core components of my curriculum.

## CHAPTER 2: PROBLEM STATEMENT & OBJECTIVES

### Overview of Assigned Project / Research

During my internship at Hyeon Infotech, I worked on multiple real-world AI and web- based projects. The two primary projects were:

* + 1. **Luxury Watch Market Analysis Platform** – A full-stack AI-powered web application providing real-time watch prices, historical trends, and smart predictions using machine learning.
    2. **Plant Disease Detection System** – A deep learning solution to identify plant diseases from images and assist farmers in taking preventive or curative actions.
    3. **Youtube Comment Sentiment Analysis -** A Python application that analyzes the sentiment of YouTube video comments using natural language processing and visualizes the results in real-time.

These projects combined artificial intelligence, frontend and backend development, and practical deployment skills.

### Problem Statement

#### Luxury Watch Market Analysis Platform:

Collectors, investors, and watch enthusiasts face difficulties in tracking real-time watch prices and market trends, which are often scattered across multiple platforms. There is no centralized platform that offers price tracking, AI-based predictions, and historical insights for luxury watches.

#### Plant Disease Detection System:

Farmers often struggle to detect diseases in crops due to lack of expert availability or visual analysis knowledge. Manual inspection is time-consuming and may lead to inaccurate diagnosis. There’s a need for an AI-based solution that automates disease detection using image classification.

#### YouTube Comment Sentiment Analysis:

Analyzing viewer sentiment from YouTube comments is essential but challenging due to the large volume of user-generated content. This project simplifies the process by using NLP techniques and the VADER sentiment analyzer to automatically classify comments as Positive, Negative, or Neutral. Through a user-friendly Streamlit interface, users can input a YouTube video URL and instantly view sentiment insights and visualizations.

### Goals & Learning Outcomes

#### Goals:

* To design and build full-stack applications powered by artificial intelligence.
* To apply machine learning techniques like NLP, computer vision, and deep learning in real scenarios.
* To integrate real-time data APIs for dynamic analysis.
* To gain experience with deploying scalable AI systems.

#### Learning Outcomes:

* Learned and applied tools such as React.js, Flask, MongoDB, TensorFlow, LangChain, and LLMs.
* Developed skills in data preprocessing, model training, evaluation, and deployment.
* Built working knowledge of frontend-backend integration, API development, and cloud deployment.
* Strengthened problem-solving, team collaboration, and real-time project handling capabilities.

### Scope and Limitations

#### Scope:

* The watch platform can be used by investors, collectors, and retailers.
* It can integrate with platforms like Chrono24, WatchBox, and Watchfinder.
* The plant disease app benefits agriculture professionals, farmers, and researchers.
* The systems are web-based, allowing for wide accessibility and possible mobile extension.

#### Limitations:

* Real-time APIs used in the watch project have rate limits and data restrictions.
* The plant disease model may struggle with low-quality or unseen images.
* Deployment of transformer models or LLMs may need GPU support.
* Each system is domain-specific and not generalized for all types of users or industries.

### Timeline Table (Weekly Plan)

**Table 2.1 Timeline Table**

|  |  |  |
| --- | --- | --- |
| Week | Task / Module | Description |
| 1 | Frontend Foundations | Learned HTML, CSS, and JavaScript for basic website structure and styling. |
| 2 | React.js Components | Developed reusable components and routing using React.js. |
| 3 | Python & ML Libraries | Practiced NumPy, Pandas, and Scikit-learn for data manipulation and modeling. |
| 4 | Data Preprocessing & Visualization | Cleaned datasets and created visualizations using Matplotlib and Seaborn. |
| 5 | NLP and Text Processing | Built text analysis tools using NLP libraries like NLTK and spaCy. |
| 6 | Deep Learning Foundations | Implemented basic ANN models using TensorFlow and Keras. |
| 7 | Advanced Neural Networks (CNN, RNN, Transformers) | Built custom models for vision and sequence tasks. |
| 8 | Flask & Model Deployment | Integrated ML models into Flask backend for web deployment. |
| 9 | Database & API Development | Worked with MongoDB, designed APIs, and stored real-time data. |
| 10 | LLMs and LangChain | Integrated Large Language Models and explored LangChain agents. |
| 11 | Debugging & Optimization | Finalized projects, resolved bugs, and optimized code and UI/UX. |
| 12 | Testing and Documentation | Systematically tested functionalities and wrote project documentation. |
| 13 | Final Report and Presentation | Prepared report and delivered final project presentation. |

## CHAPTER 3: TECHNOLOGIES & TOOLS USED

This chapter outlines the technologies, frameworks, platforms, and methodologies utilized during the internship at Hyeon Infotech, justifying their selection for the assigned AI Engineer Intern projects and learning goals.

### Tools, Frameworks & Platforms

A diverse technology stack supported web development, data science, AI, and deployment tasks.

#### Web Development:

* **Frontend:** HTML, CSS, JavaScript, React.js (for dynamic UIs like WatchInsight, PlantAid), Tailwind CSS (styling), Streamlit (interactive data apps like YouTube Sentiment Analysis, PCA dashboard).
* **Backend:** Flask (Python framework for APIs like WatchInsight, PlantAid), API Design & Implementation.

#### Programming Languages:

* **Python:** Primary language for backend, data analysis, and AI/ML tasks.
* **JavaScript:** Used for frontend development with React.js.

#### AI, Machine Learning & Data Science:

* **Core Libraries:** NumPy (numerical computation), Pandas (data manipulation).
* **Machine Learning:** Scikit-learn (Linear/Logistic Regression, K-Means, PCA).
* **Deep Learning:** TensorFlow/Keras, PyTorch (ANNs, CNNs, RNNs - e.g., Plant Disease Detection).
* **NLP:** NLTK, TextBlob, VADER (Sentiment Analysis - e.g., YouTube comments).
* **Computer Vision:** OpenCV (Image processing for Plant Disease Detection).
* **Advanced AI:** LangChain (LLM application framework), Transformers/LLMs (studied architectures like GPT, BERT), Hugging Face (platform/tools).

#### Data Management & Visualization:

* **Databases:** MongoDB (NoSQL for flexibility - e.g., user auth), PostgreSQL (Relational).
* **Visualization:** Matplotlib, Seaborn (creating statistical graphics - e.g., Iris dataset).

#### Deployment & Other Tools:

* **Platforms:** Render, Railway, Vercel.
* **APIs:** YouTube Data API.
* **Environment:** Pipenv.
* **Version Control:** Git.

### Methodologies

The internship workflow, while not strictly formal Agile/DevOps, blended several practical approaches:

* + - **Integrated Learning & Application:** Systematically learned new technologies weekly/bi-weekly (per diary) and immediately applied them to projects.
    - **Iterative Project Development:** Built projects like the Portfolio, WatchInsight, and PlantAid incrementally, refining features based on learning.
    - **Research-Focused Implementation:** Dedicated time to understanding AI/ML concepts (Deep Learning, LLMs, LangChain) before practical application.
    - **Project-Driven Skill Building:** Utilized distinct projects to target and develop skills across web development, data visualization, NLP, and AI.

### Justification for Technology Selection

Technology choices were driven by industry relevance, AI/ML suitability, project needs, and learning objectives:

* + - **Industry Standards:** Employed widely-used tools (Python, React, TensorFlow, Flask, Scikit-learn) for valuable skill development.
    - **Python's AI/ML Ecosystem:** Leveraged Python's extensive libraries for efficient data analysis, ML, DL, NLP, and CV tasks.
    - **Frontend Needs:** Used React.js for dynamic UIs and Streamlit for rapid data application prototyping.
    - **Backend Efficiency:** Chose Flask for its lightweight nature in building APIs; selected MongoDB/PostgreSQL based on data structure needs.
    - **Task-Specific Tools:** Utilized libraries like VADER for sentiment analysis, OpenCV for image processing, and LangChain for advanced AI exploration relevant to the AI Engineer role.
    - **Broad Learning Exposure:** The stack provided comprehensive exposure from web fundamentals to advanced AI (LLMs, deployment), meeting internship goals.
    - **Accessible Deployment:** Used platforms like Vercel, Render, and Railway for straightforward project deployment.

## CHAPTER 4: SYSTEM / PROJECT DESIGN

This chapter details the design considerations for the key projects undertaken during the internship. It covers the overall system architecture, user interface (UI) and backend design, database and data models where applicable, core algorithm design for AI/ML components, and outlines the standard Unified Modeling Language (UML) diagrams relevant to these systems.

### System / Solution Design (UI, Backend, Architecture)

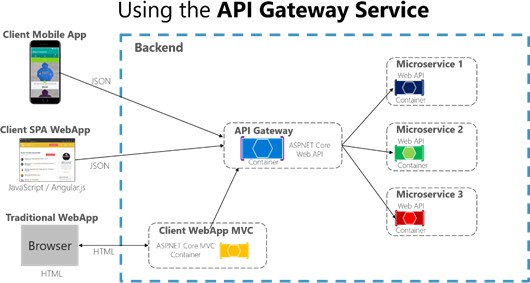
The projects varied in complexity, ranging from standalone data analysis scripts and static websites to full-stack web applications with integrated AI/ML models.

#### Overall Architecture:

* + - **Client-Server Architecture:** Adopted for major web applications like **WatchInsight**

and **PlantAid**. This typically involved:

* + - * A **Frontend (Client)** built with React.js, responsible for rendering the user interface and handling user interactions in the browser.
      * A **Backend (Server)** built with Flask (Python), responsible for handling API requests, processing data, interacting with databases, executing AI/ML models, and returning responses to the frontend.
      * **APIs** served as the communication bridge between the frontend and backend.

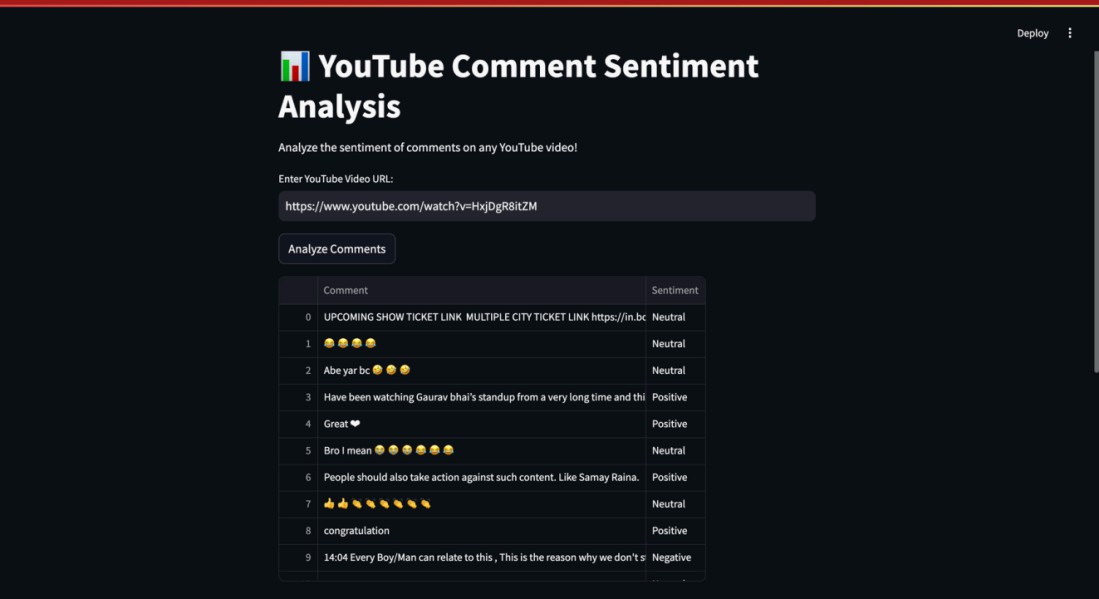


**Fig 4.1 Client-Server interaction using API gateway**

* + - **Standalone Applications/Scripts:** Projects like the **YouTube comment sentiment analysis** and **PCA Analysis** were primarily developed as Python scripts or interactive dashboards using Streamlit. These ran locally or were deployed as self-contained web apps without a separate complex backend.
    - **Static Website:** The **Personal Portfolio** was designed as a static website using HTML, CSS, and JavaScript, focusing purely on frontend presentation.

#### User Interface (UI) Design:

* + - **React-based UIs (WatchInsight, PlantAid):** Leveraged React's component-based structure to create modular and interactive interfaces. Styling was achieved using standard CSS or utility frameworks like Tailwind CSS. Key UI elements included forms for input (e.g., image upload in PlantAid, search queries), data display components, navigation menus, and interactive visualizations (e.g., price charts).
    - **Streamlit Dashboards (Sentiment Analysis, PCA Analysis):** Utilized Streamlit's built-in components to rapidly create interactive data exploration tools. UIs included file uploaders, sliders, buttons, and dynamic plots generated by Matplotlib/Seaborn. The focus was on functional data interaction rather than complex custom styling.



**Fig 4.2 PCA Streamlit UI**

* + - **Static UI (Portfolio):** Standard HTML/CSS/JavaScript were used to create a visually appealing and responsive layout showcasing skills and projects.

#### Backend Design (Flask):

* + - **API Endpoints:** Flask was used to define specific API routes (endpoints) for different functionalities (e.g., /predict for PlantAid's model inference, /login for authentication, /watches for fetching watch data).
    - **Request Handling:** Backend logic was implemented to parse incoming requests (e.g., JSON payloads, form data), validate inputs, and orchestrate necessary actions.
    - **Business Logic:** Contained the core processing logic, including calls to AI/ML models, database operations (CRUD - Create, Read, Update, Delete), and data transformations.
    - **Model Integration:** The Flask backend loaded and interacted with the trained AI/ML models (e.g., TensorFlow/Keras CNN model for PlantAid, NLP models for sentiment analysis) to perform predictions or analysis based on frontend requests.
    - **Database Interaction:** Handled connections and queries to databases like MongoDB (for user data/authentication) and potentially PostgreSQL (for structured watch data).

### Database / Data Model / Algorithm Design

#### Database / Data Model:

* + - **MongoDB (User Authentication):** Used in projects like WatchInsight and PlantAid for storing user credentials (e.g., username, hashed passwords) and potentially user profile information. Being NoSQL, it offered flexibility in schema design.



**Fig 4.3 basic User schema used in MongoDB**

* + - **No Database:** Simpler projects like the initial data visualizations or the portfolio website did not require a database.

#### Algorithm Design:

* + - **Plant Disease Detection (PlantAid):**
      * **Core Algorithm:** Convolutional Neural Network (CNN) implemented using TensorFlow/Keras.
      * **Input:** Image of a plant leaf uploaded by the user.
      * **Preprocessing:** Image resizing, normalization, potentially data augmentation (handled by OpenCV and TensorFlow/Keras image data generators).
      * **Model Architecture:** A typical CNN architecture involving convolutional layers (for feature extraction), pooling layers (for down-sampling), and dense layers (for classification). Specific pre-trained models might have been explored or used (transfer learning).
      * **Output:** Classification of the input image into predefined disease categories (or healthy).

#### YouTube Comment Sentiment Analysis:

* + - * **Core Algorithm:** NLP techniques for sentiment classification. Libraries like NLTK, TextBlob, or VADER were used.
      * **Input:** Text comments fetched via the YouTube Data API.
      * **Prepossessing:** Text cleaning (removing punctuation, converting to lowercase), tokenization, possibly stop-word removal.
      * **Sentiment Calculation:** Applying rule-based lexicons (VADER) or potentially trained classification models (using TextBlob or custom models) to assign a sentiment score (Positive, Negative, Neutral).
      * **Output:** Sentiment classification for each comment and aggregated results (e.g., percentage breakdown, word clouds).

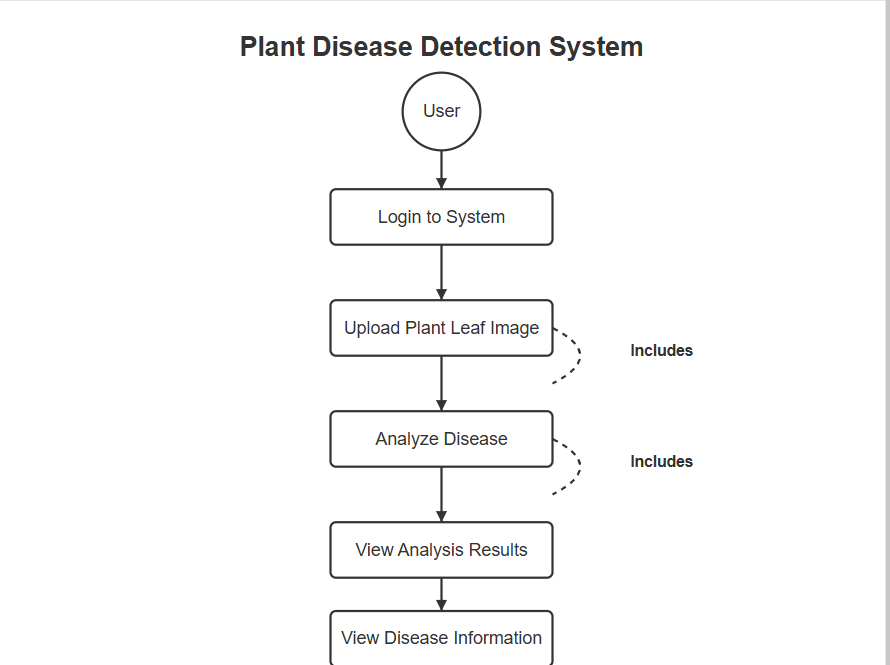
#### Watch Price Prediction :

* + - * **Core Algorithm:** Likely involved regression models (e.g., Linear Regression, Gradient Boosting, or potentially time-series models) trained on historical watch price data. Feature engineering based on watch attributes (brand, model, age, condition) would be critical.
      * **Input:** watch details provided by user such as brand name, model name, reference number, condition etc..
      * **Prepossessing:** data cleaning, feature extraction, normalization.
      * **Output**: Predict current market price, and last 5 years price history.

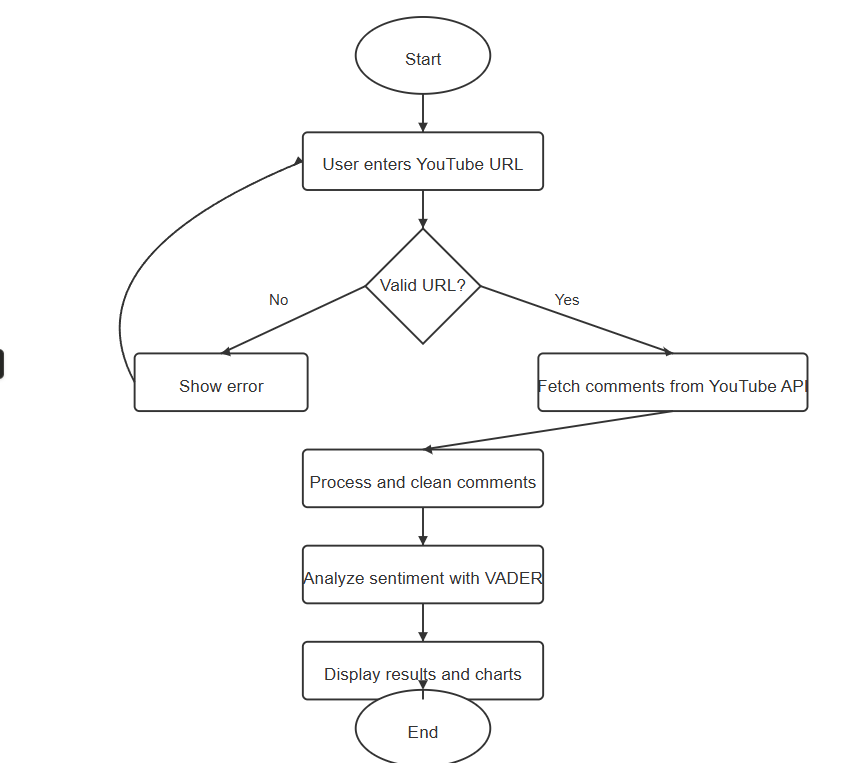
### UML Diagrams (Use Case, Class, Activity, etc.)

Unified Modeling Language (UML) diagrams are standard tools for visually representing the design and structure of software systems. While specific diagrams were not included in the provided internship materials, the following types would be relevant for documenting the projects developed:

* + - **Use Case Diagrams:** To illustrate the interactions between users (actors) and the system, showing the different functionalities available .



**Fig 4.4 PlantAid Use Case Diagram**

* + - **Activity Diagrams:** To represent workflows and process flows within the system (e.g., the step-by-step process of analyzing a plant image from upload to displaying results, or the user authentication flow).

**Fig 4.5 YouTube Comment Sentiment Analysis Activity Diagram**

## CHAPTER 5: IMPLEMENTATION

### Implementation Process (Industry or Research)

During my internship at Hyeon Infotech, the implementation process for projects followed a structured approach, integrating agile methodologies to ensure flexibility and efficiency. The process can be summarized as follows:

#### Requirement Gathering and Analysis:

* + - * The initial phase involved understanding the project goals, deliverables, and constraints. For instance, the Plant Disease Detection project required a system that could accurately identify plant diseases from user-uploaded images.
      * Tools like detailed project briefs and client meetings were utilized.

#### Design and Planning:

* + - * A detailed design phase followed, where the system architecture, database schema, and user interface were defined.
      * For the Plant Disease Detection project, this involved designing the architecture for image processing, model training, and user interaction.
      * Frameworks like React.js for the frontend and Flask for the backend were selected.

#### Development and Testing:

* + - * The development phase involved writing code, integrating different modules, and conducting regular testing.
      * For the Plant Disease Detection project, this included developing the user interface, implementing the backend logic for handling image uploads, and integrating the AI model for disease prediction.
      * I was involved in both front-end and back-end development, using React.js and Python.

#### Deployment and Integration:

* + - * The deployment phase involved deploying the application to a production environment.
      * For the web applications, this involved deploying the Flask backend and the React.js frontend.

#### Evaluation and Maintenance:

* + - * The final phase involved evaluating the system's performance and making necessary adjustments.
      * This also included ongoing maintenance and updates to ensure the system's reliability and functionality.

### Key Modules / Functions / Experiments

The internship involved working on several key modules and functions across different projects:

#### PlantAid - Plant Disease Detection Project:

* + **Image Upload and Processing:** This module handled the uploading of plant leaf images by users and pre-processed them for analysis.
    - Key functions included: File upload handling, image resizing, and format conversion.
  + **Disease Prediction:** This module used a trained AI model to analyze the processed image and predict the type of plant disease.
    - Key functions included: Model loading, image feature extraction, and disease classification.
  + **User Interface (UI) Module:** This module provided a user-friendly interface for uploading images, viewing results, and interacting with the system.
    - Key technologies: HTML, CSS, Python, MongoDB, LLM.

#### Watchinsight - Luxury Watch Website:

* + **AI-Driven Price Prediction:** This module used machine learning models to predict future watch prices.
  + **User Authentication:** Implementation of login and signup features, including email verification.
    - Key technologies: React, Flask, MongoDB, Python, ML .

### Snapshots / Screens / Code

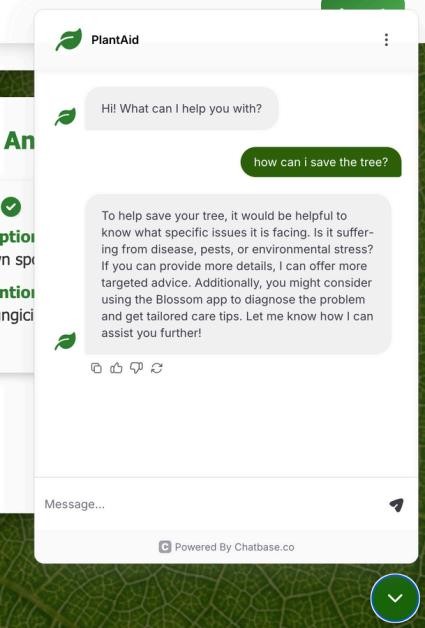
#### PlantAid - Plant Disease Detection Project:

****

**Fig 5.1 PlantAid Login/Signup Page**

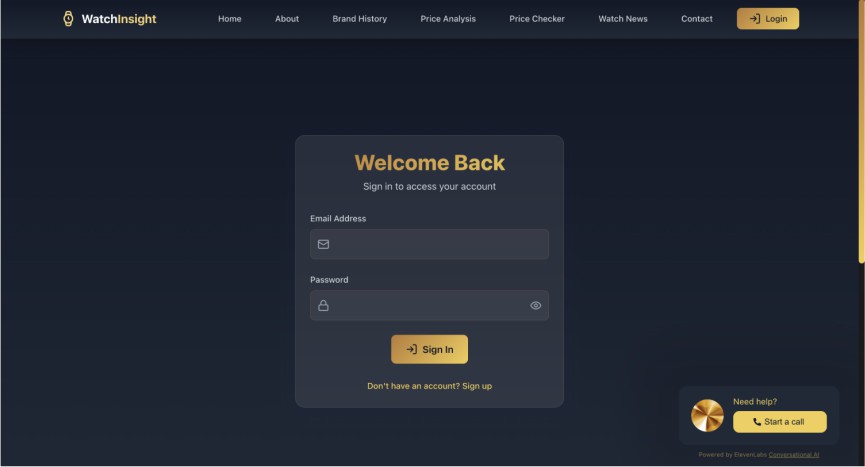
****

**Fig 5.2 PlantAid Home Page**



**Fig 5.3 PlantAid Chatbot**

#### Watchinsight - Luxury Watch Market Project:

****

**Fig 5.4 Watchinsight Login/Signup Page**

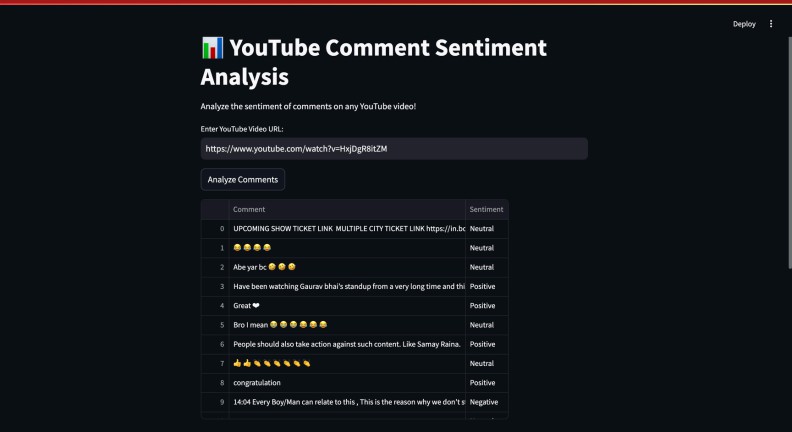
****

**Fig 5.5 Watchinsight Home Page**



**Fig 5.6 Watchinsight News Page**

#### 4. YouTube Comment Sentiment Analysis Project:

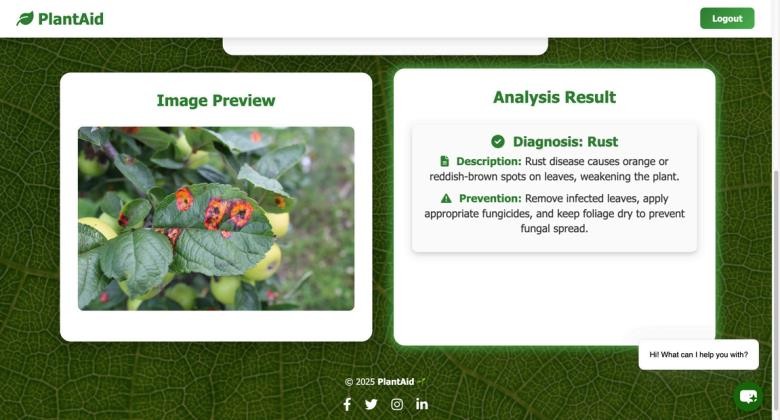
****

**Fig 5.7 Add URL Page**

### Results / Outputs

#### Plant Disease Detection Project:

* + A functional web application that allows users to upload plant leaf images and receive a diagnosis of potential diseases.



**Fig 5.8 PlantAid Analysis Page**

* + The application was able to correctly identify several common plant diseases with a certain level of accuracy.

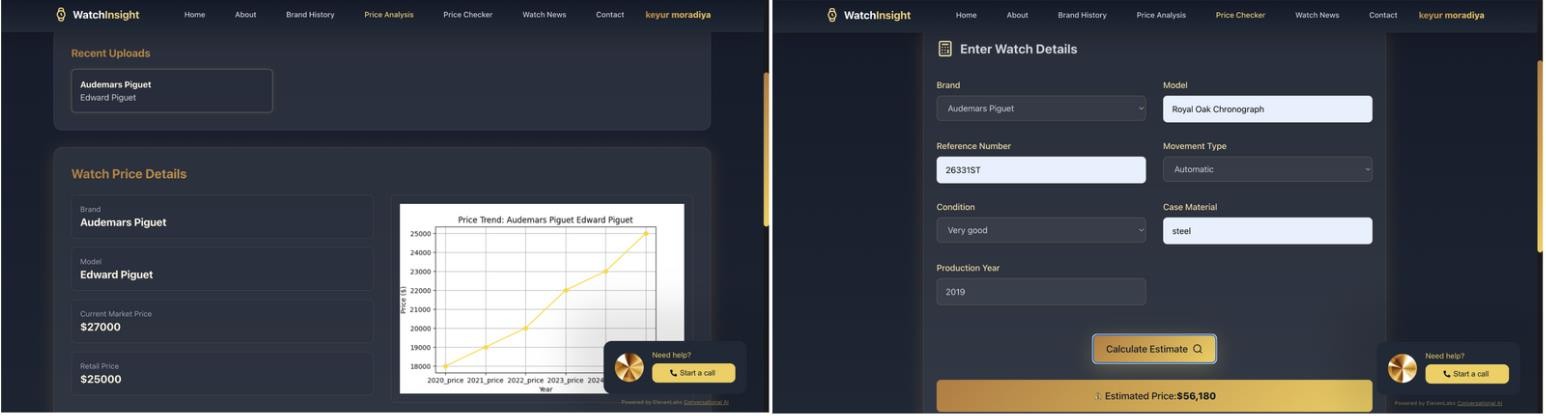
**Table 5.1 PlantAid Accuracy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Healthy | 0.94 | 0.96 | 0.95 | 157 |
| Powdery | 0.93 | 0.91 | 0.92 | 155 |
| Rust | 0.94 | 0.96 | 0.95 | 167 |
| accuracy |  |  | 0.94 | 479 |
| macro avg | 0.94 | 0.94 | 0.94 | 479 |
| weighted avg | 0.94 | 0.94 | 0.94 | 479 |

* + The project demonstrated the application of AI in agriculture and its potential to improve crop health.

#### Watchinsight - Luxury Watch Website:

* + A web platform providing real-time market insights, historical price trends, and AI-driven predictions for luxury watches.



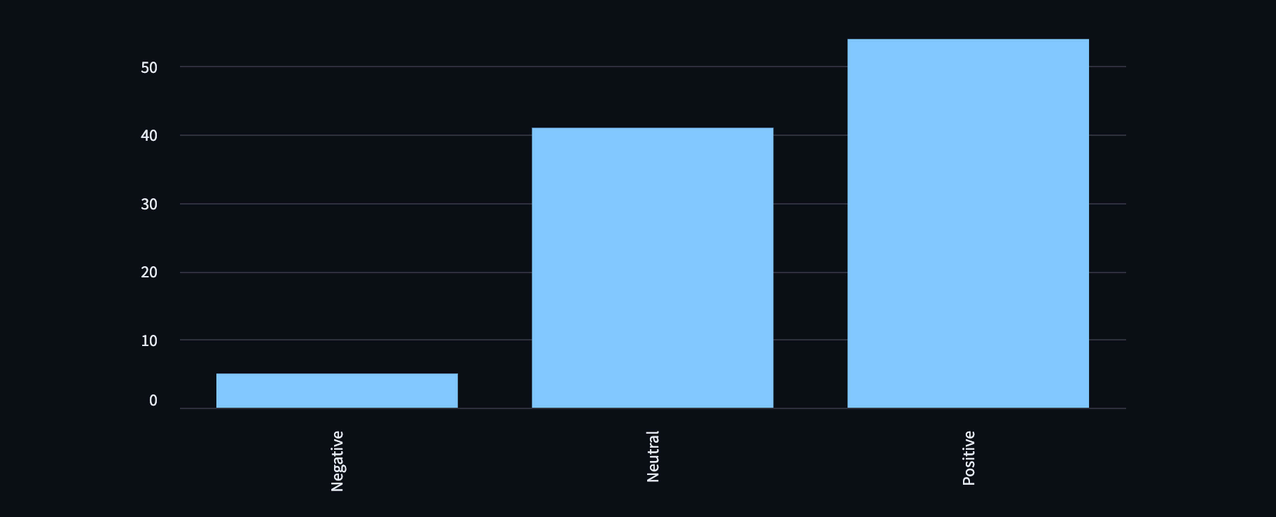
**Fig 5.9 Watchinsight Price Analysis and Price Checker Page**

* + Implemented features such as user authentication, watch database search, and integration with marketplace APIs.

**Table 5.2 Watchinsight Accuracy**

|  |  |
| --- | --- |
| Metric | Value |
| Mean Absolute Error (MAE) | 320.45 |
| Mean Squared Error (MSE) | 164589.34 |
| Root Mean Squared Error (RMSE) | 405.70 |
| R² Score | 0.88 |

#### YouTube Comment Sentiment Analysis :

****

**Fig 5.10 Bar graph of comments**

## CHAPTER 6: TESTING & VALIDATION

This chapter outlines the testing and validation methodologies employed to ensure the reliability, performance, and effectiveness of the developed projects. The following sections detail the testing approaches, sample test cases, and the analysis of the results.

### Testing Approach

The following testing approaches were utilized during the development phase of the projects:

* + - **Unit Testing:** Individual components and functions were tested to ensure they performed as expected. This approach helped in identifying and rectifying bugs early in the development cycle.
    - **Integration Testing:** The interaction between different modules and systems was tested to verify the seamless integration of various components. This ensured that the combined functionalities of the system worked correctly.
    - **User Acceptance Testing (UAT):** The final applications were tested by end-users to ensure they met the required specifications and were user-friendly. This phase validated the system's usability and its ability to function in a real-world environment.

### Sample Test Cases or Research Validation

#### PlantAid - Plant Disease Detection:

**Table 6.1 PlantAid Project Test Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Steps** | **Expected Result** | **Test Type** |
| Upload a clear image of diseased leaf | 1. Open the app 2. Upload a clear leaf image 3. Tap on "Detect Disease" | Disease is detected and correctly classified | Functional |
| Upload an image with no leaf | 1. Open the app 2. Upload an image without a plant leaf | Error: "Invalid input. Please upload a clear leaf | Validation |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3. Tap "Detect" | image." |  |
| Upload a blurry or low-res image | 1. Upload a low-quality image 2. Tap on "Detect Disease" | Warning or error message for unclear image | Error Handling |
| Upload a healthy leaf | 1. Upload a healthy plant leaf 2. Tap on "Detect Disease" | Output: "Healthy" or “No disease detected” | Functional |
| Input image size boundary test | 1. Upload an image below the minimum resolution 2. Tap on "Detect" | Error: "Image resolution too low" | Boundary Value Analysis |
| Multiple images upload | 1. Try uploading more than one image | Error: "Please upload only one image at a time" | Negative Testing |

#### WatchInsight – Luxury Watch Price Analysis:

**Table 6.2 Watchinsight Project Test Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Steps** | **Expected Result** | **Test Type** |
| Upload valid watch data | 1. Open the app 2. Enter brand and model 3. Click "Analyze Price" | Displays current market price, retail price, and historical trend graph | Functional |
| Upload empty brand/model fields | 1. Leave brand or model field blank 2. Click "Analyze Price" | Error: "Please enter both brand and model." | Validation |
| Enter non-existent watch details | 1. Enter random or invalid brand/model 2. Click "Analyze Price" | Error or warning: "Watch not found in database." | Negative Testing |
| Upload watch | 1. Upload watch image | Error: "Please enter | Validation |

|  |  |  |  |
| --- | --- | --- | --- |
| image instead of details | without entering details  2. Click "Analyze" | watch brand and model for analysis." |  |
| Submit special characters in input fields | 1. Enter brand as "@#&%" and model as "\*&^!" 2. Click "Analyze Price" | Error: "Invalid characters in input." | Validation/Input Handling |
| Input long text in model field | 1. Enter an excessively long model name (200+ characters) 2. Click "Analyze" | Error or truncation warning: "Input too long. Please enter a valid model name." | Boundary Value Analysis |

#### YouTube Comment Sentiment Analysis:

**Table 6.3 YouTube Comment Sentiment Analysis Project Test Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Steps** | **Expected Result** | **Test Type** |
| Analyze valid YouTube video URL | 1. Open the app 2. Enter a valid YouTube video URL 3. Click "Analyze Comments" | Displays sentiment analysis results and a bar chart of Positive/Neutral/Negative | Functional |
| Submit empty input | 1. Leave the input field blank 2. Click "Analyze Comments" | Error: "Please enter a YouTube video URL." | Validation |
| Enter invalid YouTube URL | 1. Enter an incorrect or broken YouTube link 2. Click "Analyze Comments" | Error: "Invalid YouTube URL. Please try again." | Validation |
| Video with no comments | 1. Enter a valid YouTube URL with no comments 2. Click "Analyze Comments" | Warning: "No comments found on this video." | Functional |

|  |  |  |  |
| --- | --- | --- | --- |
| API quota limit reached | 1. Enter URL 2. Simulate exceeding API quota | Error: "Error fetching comments: quota exceeded." | Error Handling |
| Sentiment classification boundary test | 1. Use a comment with mixed sentiment 2. Observe classification result | Expected classification: Neutral (if compound score near 0) | Boundary Value Analysis |

### Result Analysis

* + - **Luxury Watch Website:** The real-time price updates and AI-driven price predictions were well-received by users, providing valuable insights.
    - **Plant Disease Detection:** The deep learning model demonstrated high accuracy in identifying plant diseases, enabling timely intervention.
    - **Comment sentiment analysis:** The sentiment analysis tool provided a clear understanding of public opinion, with effective graphical representations.

## CHAPTER 7: CONCLUSION AND DISCUSSION

### Overall Analysis of Internship / Project Viabilities

The internship at Hyeon Infotech proved to be a highly enriching experience, offering hands- on exposure to cutting-edge technologies in Artificial Intelligence, Machine Learning, and Full-Stack Web Development. The primary projects — Luxury Watch Website (WatchInsight), Plant Disease Detection System, and YouTube Comment Sentiment Analyzer — showcased practical implementations of advanced tech to solve real-world problems.

Each project demonstrated clear viability:

* + - The WatchInsight platform meets market demands for price tracking and AI-powered predictions in the luxury watch domain.
    - The Plant Disease Detection tool offers significant benefits for agriculture by enabling early disease identification.
    - The YouTube Sentiment Analyzer provides valuable insights into public opinion through natural language processing.

### Skills Acquired

Throughout the internship, I gained expertise across multiple domains and tools:

* + - Frontend Development: HTML, CSS, JavaScript, React.js, Tailwind CSS
    - Backend Development: Python, Flask
    - Database Management: PostgreSQL, MongoDB, SQLite
    - Machine Learning & Deep Learning: Scikit-learn, TensorFlow, Keras, NumPy, Pandas
    - Data Visualization: Matplotlib, Seaborn, Chart.js
    - Natural Language Processing: NLTK, TextBlob, VADER
    - API Integration: Google APIs, ScraperAPI, RapidAPI
    - Project Management & Collaboration: GitHub, Trello, Agile methodologies

These skills were honed by solving real technical challenges and working on end-to-end project pipelines.

### Dates of Continuous Evaluation (CE-I and CE-II)

* + - Continuous Evaluation – I: 01/03/2025
    - Continuous Evaluation – II: 05/04/2025

### Problems Encountered and Possible Solutions

* + - Problem 1: Difficulty in fetching accurate real-time watch prices due to the lack of public APIs.

Solution: Integrated ScraperAPI and Google Search API to extract relevant watch data from platforms like Chrono24 and WatchBox.

* + - Problem 2: Image classification accuracy for plant diseases was initially low due to poor-quality datasets.

Solution: Improved data preprocessing, applied image augmentation, and fine-tuned a CNN model for better performance.

* + - Problem 3: Handling sarcasm and ambiguity in sentiment analysis for YouTube comments.

Solution: Combined multiple NLP libraries (TextBlob, VADER) and trained a custom sentiment classifier using labeled datasets.

### Summary of Internship / Project Work

The internship focused on solving practical problems through AI-driven applications. Key projects include:

* + - Luxury Watch Website (WatchInsight): Developed a modern web platform with features like real-time price tracking, historical trend graphs, AI-based price prediction, and data from multiple luxury watch platforms. The site also included pages for watch news, database, and user-friendly analytics tools.
    - Plant Disease Detection System: Designed a web tool that identifies diseases in plants using image classification with deep learning. The solution aids farmers and agriculture experts in quick and accurate disease diagnosis.
    - YouTube Comment Sentiment Analysis: Built a Flask-based application to fetch YouTube video comments and analyze them for public sentiment using natural language processing and machine learning algorithms.

### Limitations and Future Enhancement

#### Limitations:

* + - Some third-party APIs used in the projects had rate limits or required premium access.
    - The plant disease model was trained on limited datasets, which may affect its generalizability to rare diseases.
    - Sentiment analysis faced challenges in understanding context, sarcasm, and multilingual comments.

#### Future Enhancements:

* + - Integrate premium APIs or develop proprietary scraping methods to improve watch data accuracy and real-time updates.
    - Expand the plant disease detection dataset with more diverse and high-resolution images.
    - Incorporate advanced NLP techniques like BERT or GPT-based models for more accurate sentiment classification.
    - Add user authentication, notification systems, and mobile responsiveness to all platforms for a better user experience.

## REFERENCES

1. **Google API Documentation** – YouTube Data API v3 https://developers.google.com/youtube/v3
2. **ScraperAPI** – Web Scraping API for Real-Time Data Extraction https:/[/www.scraperapi.com/documentation](http://www.scraperapi.com/documentation)
3. **Scikit-learn** – Machine Learning in Python https://scikit-learn.org/stable/
4. **TensorFlow** – An Open-Source Machine Learning Framework <https://www.tensorflow.org/>
5. **Keras** – Deep Learning API for TensorFlow <https://keras.io/>
6. **Pandas** – Python Data Analysis Library https://pandas.pydata.org/
7. **NumPy** – Fundamental Package for Scientific Computing <https://numpy.org/>
8. **Matplotlib** – Python Plotting Library <https://matplotlib.org/>
9. **NLTK (Natural Language Toolkit)** – Platform for NLP in Python <https://www.nltk.org/>
10. **TextBlob** – Simplified Text Processing <https://textblob.readthedocs.io/en/dev/>
11. **VADER Sentiment Analysis** – Lexicon and Rule-Based Sentiment Analysis Tool <https://github.com/cjhutto/vaderSentiment>
12. **Flask** – Python Web Framework https://flask.palletsprojects.com/
13. **React.js** – A JavaScript Library for Building User Interfaces <https://reactjs.org/>
14. **PostgreSQL** – Open Source Relational Database <https://www.postgresql.org/>
15. **MongoDB** – NoSQL Database for Modern Applications <https://www.mongodb.com/>
16. **Chrono24** – Luxury Watch Marketplace (Data Reference) <https://www.chrono24.com/>
17. **WatchBox** – Buy, Sell & Trade Pre-Owned Luxury Watches <https://www.thewatchbox.com/>
18. **Google News API** – Search and Retrieve News Articles <https://newsapi.org/>
19. **GitHub** – Code Hosting Platform for Version Control <https://github.com/>
20. **Trello** – Project Management Tool <https://trello.com/>