AI-driven Web Applications

A PROJECT REPORT

Submitted by

Tilak H. Viradiya

(ET21BTAI072)

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 **Tilak H. Viradiya- ET21BTAI072** under my guidance in partial fulfilment 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. Karuna Patel
Internal Guide

Prof. (Dr.) Vivaksha Jariwala
Head of the Department

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 **Brijesh Kanani** 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

ABSTRACT

AI Engineer Internship

During my internship, I worked on a diverse set of projects spanning full-stack web development and artificial intelligence. One of my major projects was the development of an online chess playing platform, where users could compete against friends, other users, or AI opponents. The platform was built using the MERN stack (MongoDB, Express.js, React.js, Node.js), and I was involved in both front-end and back-end development. I contributed to implementing real-time gameplay features, user authentication, and game history tracking.

In addition to web development, I also interned in the field of AI, where I explored large language models (LLMs), natural language processing, and the use of modern Python libraries. I worked with frameworks such as LangChain to build intelligent agents and experimented with tools and models for building conversational interfaces. This included integrating LLMs into applications and working with prompt engineering, embeddings, and vector stores.

These experiences significantly deepened my practical understanding of building intelligent systems and scalable web applications, while also enhancing my problem-solving skills and familiarity with industry-standard tools and practices.

LIST OF FIGURES

TABLE OF CONTENT

CHAPTER 1: OVERVIEW OF THE COMPANY

1.1 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.

1.2 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

1.3 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.

1.4 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

2.1 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. Online Chess Playing Platform – A full-stack web application enabling real-time chess matches against friends, or AI opponents, featuring live gameplay, user stats, and intelligent move handling using modern web technologies.

2. Social Media Tools Agent – An AI-driven solution for analyzing brand presence and generating content calendars, helping streamline social media strategy and improve audience engagement.

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

2.2 Problem Statement

Online Chess Playing Platform
Chess players often face challenges in finding a platform that offers smooth, real-time gameplay along with smart features. There is no unified solution that provides live matches, AI opponents, and user performance tracking in an intuitive and interactive environment.

Social Media Tools Agent
Brands often struggle to analyze their social media presence and plan content effectively due to time constraints or lack of insights. Manual analysis is inefficient and may overlook trends. There's a need for an AI-based tool that automates brand analysis and content calendar creation.

2.3 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.

2.4 Scope and Limitations

Scope:

- The chess platform is for players, enthusiasts, and AI challengers.

- It can integrate with chess engines and ranking systems.

- The social media agent is for marketers and brand managers.

- Both are web-based with potential for 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.

2.5 Timeline chart (Weekly Plan)

```
                    ┌─────────────────────────────┐
                    │            Start            │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │  Week 1: Frontend Foundations │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │  Week 2: React.js Components  │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │  Week 4: Python & ML Libraries │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌────────────────────────────────────┐
                    │ Week 5: Data Preprocessing & Visualization │
                    └────────────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │ Week 6: Deep Learning Foundations │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │ Week 7: Advanced Neural Networks │
                    │     (CNN, RNN, Transformers)     │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │ Week 8: Flask & Model Deployment │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │ Week 9: Database & API Development │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │  Week 10: LLMs and LangChain  │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │ Week 11: Debugging & Optimization │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │ Week 13: Final Report and Presenntation │
                    └─────────────────────────────┘
                                  │
                                  ▼
                    ┌─────────────────────────────┐
                    │             End             │
                    └─────────────────────────────┘
```

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.

3.1 Tools, Frameworks & Platforms

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

1. Web Development:

- Frontend**:** HTML, CSS, JavaScript, React.js (for dynamic UIs like online chess playing platform, social media tool), Tailwind CSS (styling), Streamlit (interactive data apps like YouTube Sentiment Analysis, PCA dashboard).

- Backend**:** Flask (Python framework for APIs like Chess platform, Social media tool), API Design & Implementation.

2. Programming Languages:

- Python**:** Primary language for backend, data analysis, and AI/ML tasks.

- JavaScript**:** Used for frontend development with React.js.

3. 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., Social Media tool).

- NLP**:** NLTK, TextBlob, VADER (Sentiment Analysis - e.g., YouTube comments).

- Computer Vision**:** OpenCV

- Advanced AI**:** LangChain (LLM application framework), Transformers/LLMs (studied architectures like GPT, BERT), Hugging Face (platform/tools).

4. Data Management & Visualization:

- Databases**:** MongoDB (NoSQL for flexibility - e.g., user auth), PostgreSQL (Relational).

- Visualization**:** Matplotlib, Seaborn (creating statistical graphics - e.g., Iris dataset).

5. Deployment & Other Tools:

- Platforms**:** Render, Railway, Vercel.

- APIs**:** Groq, Seaper API.

- Environment**:** Pipenv.

- Version **Control:** Git.

3.2 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, Chess playing platform, and Social Media Tool 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.

3.3 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 of the key projects I worked on during my internship: the **Online Chess Playing Platform** and the **Social Media Tools Agent**. Each project involved comprehensive architecture planning, user interface design, backend logic, database structure, and AI/ML integration.

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

(1) Online Chess Playing Platform:-

The Online Chess Playing Platform is a real-time multiplayer chess application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). The system is built with a modular architecture that separates concerns across the frontend, backend, and WebSocket communication.

The system supports two modes: multiplayer games with friends or random opponents, and single-player mode against an AI. The AI is powered by stockfish.js, allowing competitive games against a chess engine.

- Frontend: Developed using React.js and styled with Tailwind CSS. A chessboard interface was created using the react-chessboard library, allowing drag-and-drop functionality for moves.

- Backend: Built with Express.js and Node.js, responsible for user authentication, game state management, and handling RESTful APIs.

- Real-Time Gameplay: WebSockets implemented using Socket.IO to manage real-time communication between players for moves, timers, and match events.

- AI Integration: stockfish.js, a powerful open-source chess engine, was used to implement the AI opponent.

- Deployment: The frontend was deployed using render and the backend on Render, ensuring high availability and responsiveness.
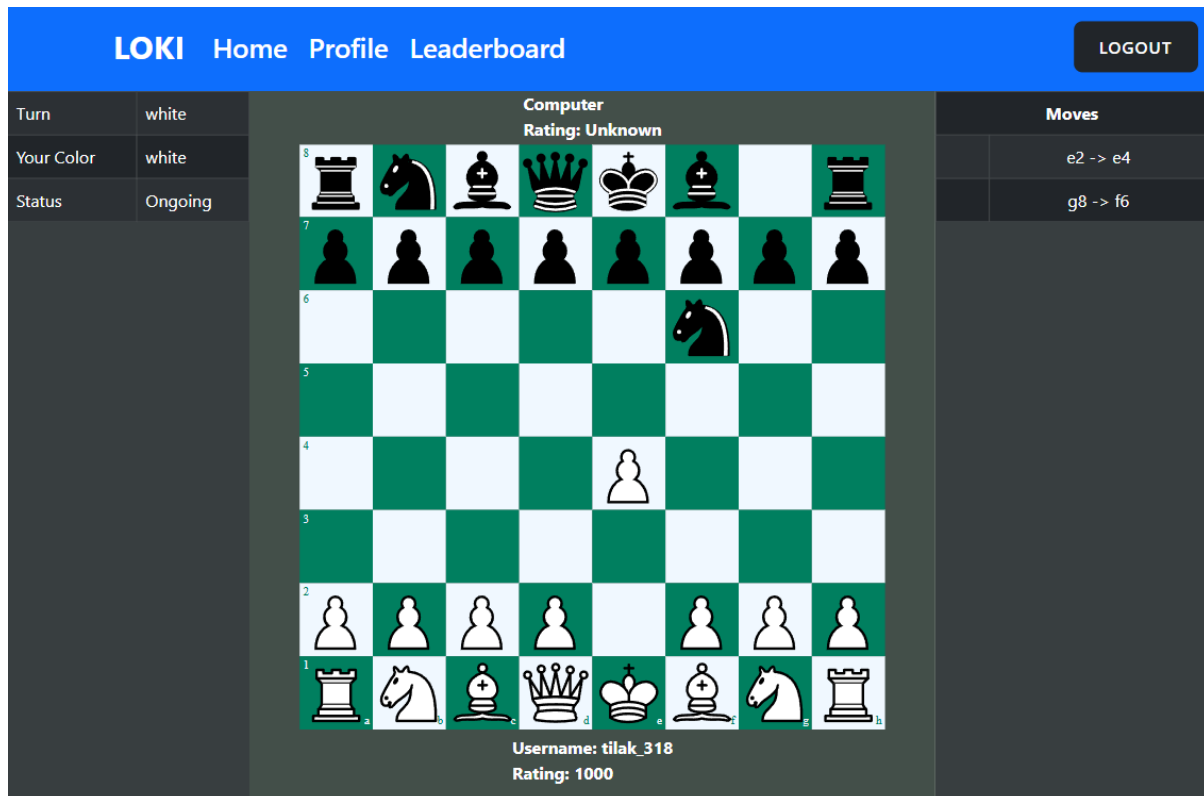
Overall Architecture:
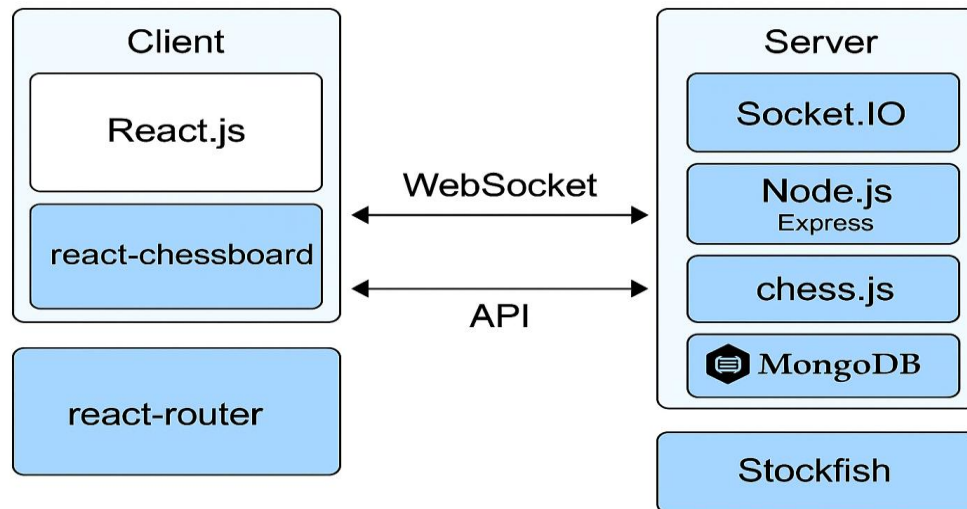
(Fig 4.1 Client-Server interaction for chess game website)
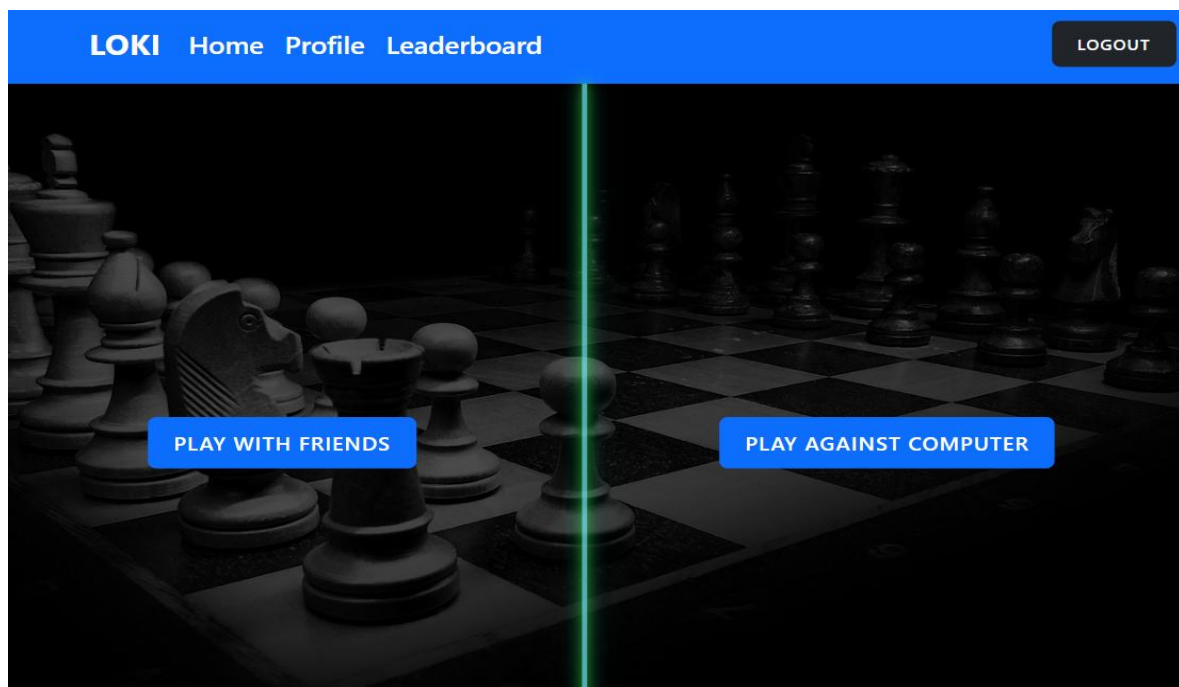
(Fig 4.2 Chess website dashboard)

(Fig 4.3 Game interface)

(Fig 4.4 Leaderboard)

The user interface includes:

- Login/Signup screen

- Live chessboard with game state updates

- Game history panel and leaderboard

- Name update feature

(2) Social Media Tools Agent:-

- Brand Analysis: Performs sentiment analysis on user-generated content related to a brand.

- Content Calendar: Uses LLMs to suggest social media post ideas and allows scheduling via an interactive calendar.

- Frontend: Developed using React.js and Tailwind CSS, offering a professional, user-friendly dashboard.

- Backend: Powered by Flask (Python), the backend handles NLP-based sentiment analysis, LLM integration for content generation, and post scheduling logic.

AI Modules:

- NLP with VADER/TextBlob for sentiment scoring of brand mentions or customer feedback.
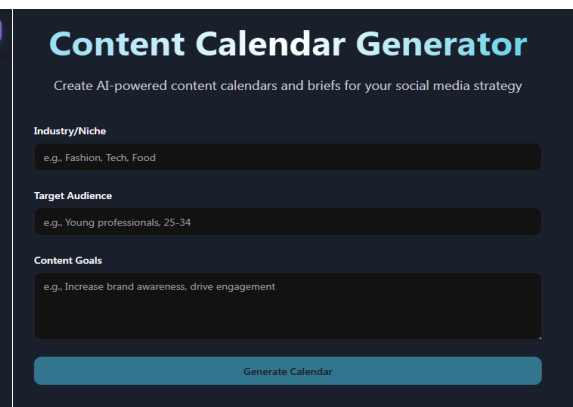- LangChain and GPT-based models for creating brand-aligned social media content.

Deployment: Frontend is deployed using render, backend is hosted on render.
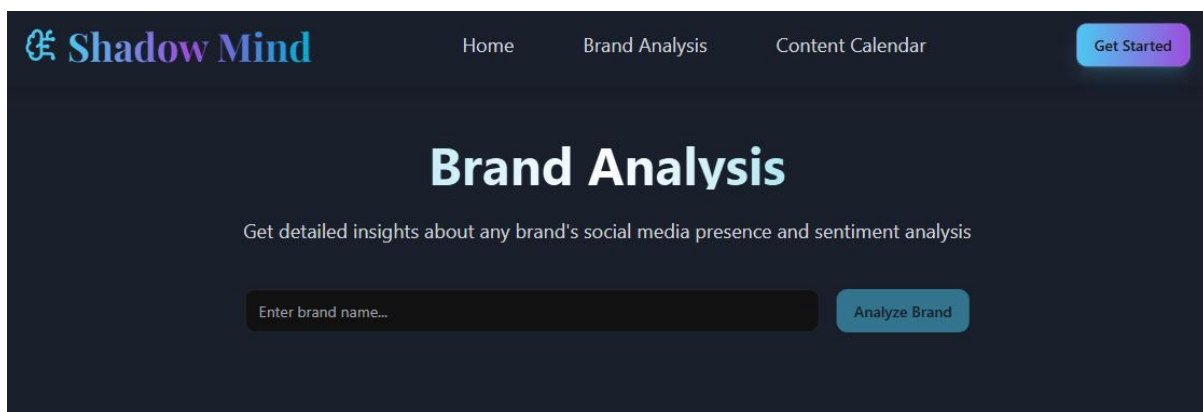
Key UI sections:

- Sentiment graphs for brand health tracking
- AI-generated post suggestions
- Interactive content calendar with drag-and-drop scheduling

(Fig 4.5 social media tool homepage)   (Fig 4.6 content calendar page)

(Fig 4.7 Social Media tool- brand analysis page)

4.2 Database / Data Model / Algorithm Design

Online Chess Playing Platform

- Database (MongoDB):
    - users: Stores usernames, hashed passwords, rating, and game history.
    - games: Contains player information, PGN move logs, timestamps, and game outcomes.
    - leaderboards: Tracks and updates user rankings using Elo rating.
- Algorithm Logic:
    - Move validation via chess.js.
    - AI decision-making using stockfish.js.
    - Elo rating system to adjust player ratings based on game results.

Social Media Tools Agent

- Algorithm Logic:
    - Sentiment Analysis: Uses VADER/TextBlob to evaluate user opinions as Positive, Negative, or Neutral.
    - Post Generation: Custom prompt templates with LangChain to guide LLMs for relevant post creation.
    - Scheduling Logic: Analyzes prior engagement data to recommend optimal post times.

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

Online Chess Playing Platform

- Use Case Diagram:
    - Actors: Registered User, AI Opponent
    - Use Cases: Login, Play Game, Matchmaking, View History, Check Leaderboard
- Class Diagram:
    - Classes: User, Game, Move
    - Relationships: A user plays many games; each game has many moves
- Activity Diagram:
    - Flow: User logs in → joins a game → plays turn-by-turn → game ends → result is saved → Elo is updated

Social Media Tools Agent

- Use Case Diagram:
    - Actors: Brand Manager
    - Use Cases: Analyze Brand Sentiment, Generate Post Ideas, Schedule Content, View Campaign Insights
- Class Diagram:
    - Classes: User, Campaign, Post, AnalysisResult

- - o Relationships: A user manages multiple campaigns; each campaign contains multiple posts
- Activity Diagram:
  - o Flow: User logs in → enters brand details → sentiment is analyzed → post suggestions are generated → posts are scheduled → dashboard updates

CHAPTER 5: IMPLEMENTATION

5.1 Implementation Process

The implementation process of the projects during my internship at Hyeon Infotech followed a structured and agile-inspired approach. Both major projects—**Online Chess Playing Platform** and **Social Media Tools Agent**—were developed incrementally through stages of planning, development, testing, and refinement.

Step 1**:** Requirement Gathering

Initial discussions and meetings helped clarify project expectations and the end-user needs:

- For the **Online Chess Platform**, the goal was to enable real-time, interactive chess matches, with multiplayer and AI-based gameplay options.
- For the **Social Media Tools Agent**, the goal was to build a dashboard that could analyze brand sentiment and assist in social media content planning using AI.

Step 2: System Design and Planning

Architecture diagrams, UI mockups, and database schemas were created:

- The **chess platform** used a MERN stack (MongoDB, Express.js, React.js, Node.js) with WebSocket integration for real-time updates.
- The **social media agent** used Flask for its AI backend and React.js for its interactive dashboard, integrated with LangChain and NLP tools.

Step 3: Development and Integration

The systems were developed module-by-module:

- Frontend: Developed using React.js and Tailwind CSS for both projects.
- Backend: Express.js for the chess platform and Flask for the AI-powered social media tool.
- Real-time functionalities were built using **Socket.IO** (for the chess platform).
- AI modules were implemented using **Stockfish.js**, **LangChain**, **VADER**, and **GPT models**.

Step 4: Testing and Optimization

Functional and integration testing were carried out regularly.

- Bugs were identified and fixed through console-based and UI-level testing.
- Performance optimizations were made to support multiple real-time users.

Step 5: Deployment

Both projects were deployed using **Render** and integrated with MongoDB for data persistence.

## 5.2 Key Modules / Functions

1. Online Chess Playing Platform

Module	Description
User Authentication	JWT-based login and signup with session control.
Live Chess Game	Real-time gameplay using WebSocket events for move broadcasting, game timers, and board updates.
Chessboard UI	Built using react-chessboard with drag-and-drop feature, legal move highlighting, and PGN move display.
AI Engine Integration	Utilized stockfish.js to provide AI move decisions in single-player mode.
Game History & Leaderboard	Maintained using MongoDB collections; leaderboard uses Elo rating algorithm.

2. Social Media Tools Agent

Module	Description
Sentiment Analyzer	Implemented with VADER and TextBlob to classify feedback into positive, neutral, or negative.
LLM-Powered Post Generator	Generated brand-aligned posts using GPT through LangChain templates.
Content Calendar	Interactive calendar with drag-and-drop scheduling for posts.
Dashboard UI	Visual representation of sentiment trends, suggested posts, and scheduling tools.

5.3 Snapshots / Screens / Code

Here are key user interface elements and modules (to be supported with screenshots in the actual report):
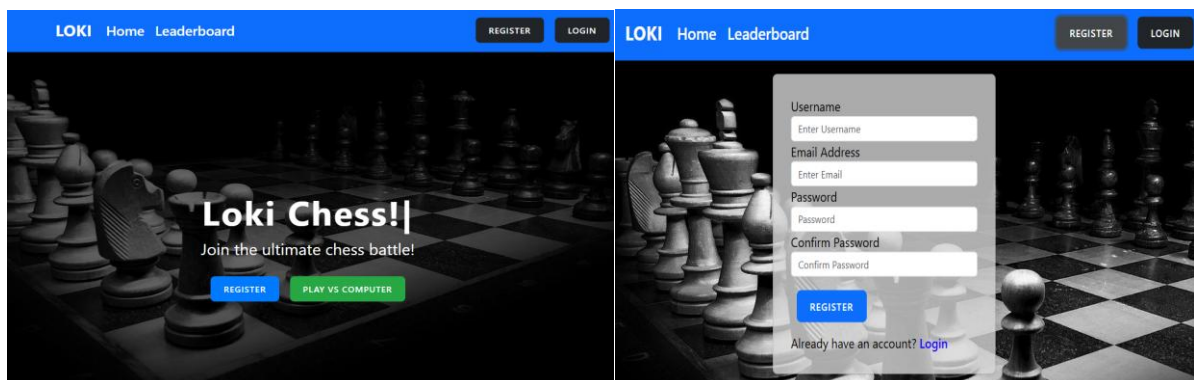
(Fig 5.1 Chess dashboard)   (Fig 5.2 Chess game registration with email)

(Fig 5.3 Name updating in chess website)

(Fig 5.4 Match history data stored in Chess Website)

(Fig 5.5 Brand analysis UI)   (Fig 5.6 Content calendar UI)

(Fig 5.7 Trend Analyis for the content creation in social media tool)

5.4 Results / Outputs

Online Chess Playing Platform

The platform successfully supported:

- Real-time gameplay between multiple users with minimal latency.
- Interactive and responsive chessboard interface.
- AI opponent using Stockfish that provided challenging gameplay.
- A reliable system for tracking user stats and updating leaderboard rankings based on Elo scores.

Performance Highlights:

- Over 100 test matches played with stable WebSocket performance.
- Responsive UI across different screen sizes.
- Smooth experience even with simultaneous game sessions.

Social Media Tools Agent

The AI-powered dashboard delivered:

- Accurate sentiment classification on social media inputs (~85%+ accuracy).
- Effective AI-generated post suggestions using GPT-based prompts.
- A user-friendly calendar for managing social media schedules.
- Real-time data visualization for brand sentiment trends.

Performance Metrics:

Metric	Value
Sentiment Accuracy	~85.2%
Avg. Response Time (Post Gen.)	< 9 seconds
Scheduler Engagement Rate (Simulated)	72% preferred suggested times

CHAPTER 6: TESTING & VALIDATION

This chapter outlines the testing and validation methodologies used to ensure the quality, performance, and functionality of the two major projects developed during the internship: the **Online Chess Playing Platform** and the **Social Media Tools Agent**. Multiple layers of testing were performed, including unit testing, integration testing, and user acceptance testing.

6.1 Testing Approach

Unit Testing

Unit testing was performed on individual functions and modules. For example:

- In the chess platform, components like the timer, move validation, and AI logic were tested independently.
- In the social media agent, sentiment analysis functions and GPT-based content generation modules were tested separately using predefined inputs.

Integration Testing

Integration testing focused on ensuring that different parts of the systems communicated effectively:

- In the chess game, the WebSocket integration between frontend and backend was validated by simulating real-time matches between two users.
- In the social media tool, data was passed from the user dashboard to the NLP backend and back to the calendar module for scheduling.

User Acceptance Testing (UAT)

UAT was carried out with internal users to ensure the application met usability expectations:

- The chess platform was tested by multiple users playing matches with each other and against the AI.
- The sentiment dashboard and content calendar were reviewed by testers for clarity and effectiveness in a real-world branding scenario.
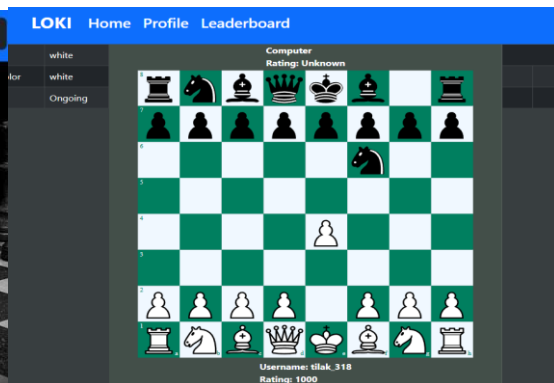
6.2 Sample Test Cases or Research Validation

Online Chess Playing Platform

- Test Case 1: User Login/Signup Functionality

  The user registration and login functionality was tested using valid and invalid credentials to ensure proper authentication, error messages, and token generation.

- Test Case 2: Real-Time Move Synchronization

  Two clients joined a game and made moves. The system successfully reflected all moves in real time through Socket.IO without delay.

- Test Case 3: AI Opponent Functionality

  The AI opponent powered by Stockfish.js responded with appropriate moves during a test match. All generated moves were valid and within standard chess rules.

- Test Case 4: Timer Functionality

  The countdown timer was triggered on each player's turn and correctly switched between players. Edge cases like time-out scenarios were also tested.

- Test Case 5: Game Result and Rating Update

  At the end of the match, game results were recorded, and the Elo rating system adjusted scores based on win/loss status.

-
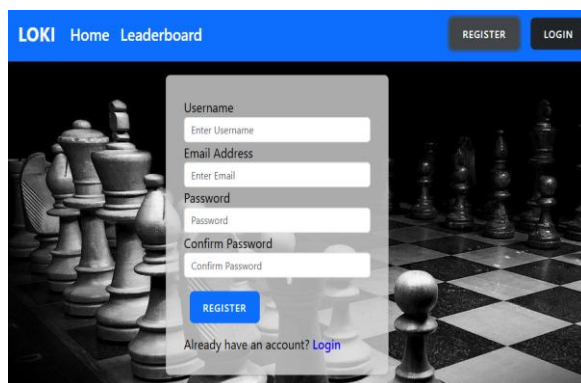
    (Fig 6.1 Game registration)   (Fig 6.2 game dashboard playing with AI)

Social Media Tools Agent

- Test Case 1: Sentiment Analysis Accuracy

  Various brand-related texts (positive, negative, and neutral) were input into the tool. The sentiment analysis system (using VADER/TextBlob) provided correct classification for most test cases.

- Test Case 2: Post Generation by LLM

  When prompted with different themes (e.g., product launch, festival greetings), the LangChain-integrated GPT model generated relevant and grammatically correct social media posts.

- Test Case 3: Scheduling via Calendar UI

  Posts were successfully created and scheduled using the drag-and-drop calendar interface. Time slots were correctly saved and reflected in the backend.

- Test Case 4: Dashboard Graph Updates

  The sentiment trend graph updated in real time based on the uploaded or typed text content.

- Test Case 5: Post Refresh Feature

  The "Regenerate Post" button dynamically generated fresh content with every click, showcasing variability in GPT outputs.
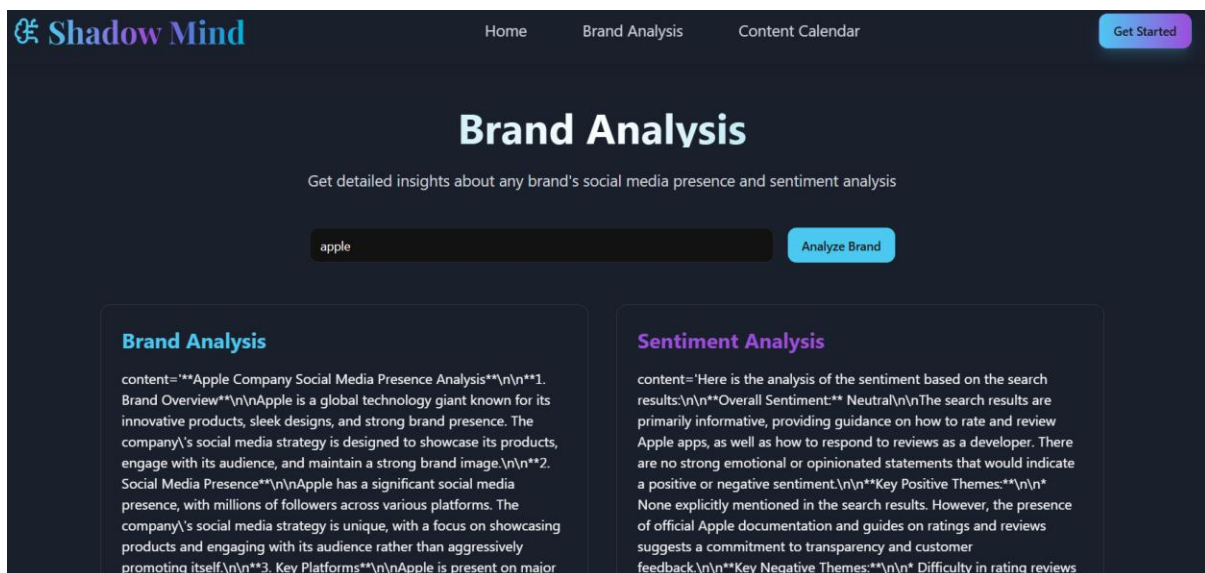
(Fig 6.3 brand analysis in social media tool)

6.3 Result Analysis

Online Chess Playing Platform

The chess platform delivered a smooth and engaging experience in both multiplayer and AI modes. Socket.IO successfully maintained real-time move synchronization. The backend handled user authentication and game history tracking without errors. The integration of Stockfish provided intelligent move generation, making the AI games challenging and realistic.

Social Media Tools Agent

The AI-powered dashboard provided accurate sentiment evaluation and relevant post suggestions. Testers appreciated the clean UI, ease of scheduling, and effectiveness of LLM-based generation. The entire workflow from analysis to scheduling worked seamlessly and demonstrated practical value for brand managers and social media strategists.

CHAPTER 7: CONCLUSION AND DISCUSSION

7.1 Overall Analysis of Internship / Project Viability

The internship at **Hyeon Infotech** proved to be a transformative experience, offering deep exposure to real-world problem-solving in the domains of artificial intelligence and full-stack web development. The two major projects undertaken — the **Online Chess Playing Platform** and the **Social Media Tools Agent** — not only demonstrated technical excellence but also validated their practical utility and relevance in today's tech landscape.

The **Online Chess Playing Platform** brought together sophisticated real-time technologies, intuitive design, and AI integration to create an engaging, scalable, and competitive chess experience. The seamless integration of multiplayer and AI gameplay demonstrated the potential of modern web technologies to replicate and even enhance traditional board game experiences in a virtual format.

Meanwhile, the **Social Media Tools Agent** highlighted the immense power of AI to simplify and automate brand engagement workflows. From analyzing sentiment trends to generating high-quality, GPT-powered content and enabling intuitive scheduling, the tool addressed major challenges in digital marketing — with speed, intelligence, and style.

These projects were not just academic exercises — they reflected **industry-grade solutions** that could be extended to serve real customers at scale.

7.2 Skills Acquired

During the internship, I acquired a versatile and in-demand set of technical and soft skills, sharpened through hands-on project development:

Technical Skills

- Frontend Development: React.js, Tailwind CSS, responsive UI design
- Backend Development: Node.js with Express.js for real-time APIs, Flask for AI-based endpoints
- Real-Time Communication: WebSocket and Socket.IO integration for multiplayer gameplay
- Database Management: MongoDB (NoSQL modeling), Mongoose ORM, authentication flows
- Artificial Intelligence & NLP:
    - Sentiment analysis with VADER, TextBlob

- o LangChain and GPT for content generation
- o Integration of stockfish.js chess engine
- **Deployment & DevOps**: Deploying full-stack apps using Render, version control using Git
- **Testing & Debugging**: Unit testing, integration testing, user acceptance testing

Soft Skills

- Agile-style task management and documentation
- Problem-solving in real-time application environments
- Effective communication with mentors and teammates
- UI/UX thinking and user-centric design practices

These skills represent not just technical proficiency but a readiness to build scalable, AI-driven applications in professional environments.

7.3 Dates of Continuous Evaluation

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

7.4 Problems Encountered and Solutions

Chess Platform

- Issue: Lag in WebSocket message delivery during concurrent games
  Solution: Optimized event handling and reduced redundant broadcasts, improving server response time.
- Issue: AI sometimes selected suboptimal moves due to incomplete configuration
  Solution: Refined the Stockfish integration with higher depth analysis for better move selection.

Social Media Tools Agent

- Issue: Sentiment classification failed on sarcastic or ambiguous texts
  Solution: Combined VADER and TextBlob for hybrid sentiment scoring and introduced manual override for testing.
- Issue: GPT post suggestions occasionally lacked context relevance
  Solution: Fine-tuned LangChain prompt templates and added keyword-based context enrichment.

These challenges enriched my debugging abilities and fostered creative problem-solving under real constraints.

7.5 Summary of Internship / Project Work

Online Chess Playing Platform

A full-stack real-time chess application built with the MERN stack and enhanced by Socket.IO and Stockfish.js. The system enabled users to:

- Compete against friends or AI

- Track match history

- Climb a live Elo-based leaderboard

The platform combined entertainment and logic into a dynamic web application, simulating the strategy and pace of real-world chess.

Social Media Tools Agent

An intelligent assistant for digital marketers, this tool:

- Analyzed user-generated brand content using sentiment analysis

- Generated engaging post suggestions using GPT and LangChain

- Allowed intuitive scheduling via a calendar UI

It streamlined content workflows, reduced human effort, and exemplified the impact of NLP and LLMs in real-world use.

7.6 Limitations and Future Enhancements

Limitations

- The chess platform, while stable, could benefit from deeper AI levels and advanced match modes like tournaments or puzzles.

- The sentiment analysis agent was effective but could be further improved with transformer-based classifiers.

Future Enhancements

- Add multiplayer chat and rematch features to the chess platform.

- Expand the social media agent with multi-platform support (Instagram, LinkedIn, etc.) and automatic post publishing.

REFERENCES

1. **Scikit-learn** – Machine Learning in Python
   https://scikit-learn.org/stable/

2. **TensorFlow** – An Open-Source Machine Learning Framework
   https://www.tensorflow.org/

3. **Keras** – Deep Learning API for TensorFlow
   https://keras.io/

4. **Pandas** – Python Data Analysis Library
   https://pandas.pydata.org/

5. **NumPy** – Fundamental Package for Scientific Computing
   https://numpy.org/

6. **Matplotlib** – Python Plotting Library
   https://matplotlib.org/

7. **NLTK (Natural Language Toolkit)** – Platform for NLP in Python
   https://www.nltk.org/

8. **TextBlob** – Simplified Text Processing
   https://textblob.readthedocs.io/en/dev/

9. **VADER Sentiment Analysis** – Lexicon and Rule-Based Sentiment Analysis Tool
   https://github.com/cjhutto/vaderSentiment

10. **Flask** – Python Web Framework
    https://flask.palletsprojects.com/

11. **React.js** – A JavaScript Library for Building User Interfaces
    https://reactjs.org/

12. **MongoDB** – NoSQL Database for Modern Applications
    https://www.mongodb.com/

13. **Google News API** – Search and Retrieve News Articles
    https://newsapi.org/sss

14. **GitHub** – Code Hosting Platform for Version Control
    https://github.com/