Smart Freelancing

Faculty of Computer and Information - Minia University - in Partial Fulfillment of the Requirements for the Degree of Bachelor of Computer Science.

By:

Abdelrahman Ashour abdelrahmanashour758@gmail.com	Ali Jamal Mohammed ag0355121@gmail.com
Abdelrahman Masoud abdelrahmanmasoud206@gmail.com	Ahmed Hassan ahmdhsnbas598@gmail.com
Amr Khaled khamr6524@gmail.com	Mohammed Ahmed m011236320@gmail.com

Supervised By:

Dr. Eman Mamdouh
@mu.edu.eg

Eng. Mostafa Mahmoud
@mu.edu.eg

Abstract

This project presents an intelligent freelance marketplace designed to enhance the experience of both clients and freelancers through advanced AI-driven capabilities...

ACKNOWLEDGEMENT

We are very thankful to everyone who supported us to complete this documentation effectively...

Contents

Chapter 1: Introduction

1.1 Main Idea

AI-powered freelancing platform that facilitates seamless interaction between clients and freelancers while enhancing the overall freelancing process through intelligent automation. The system not only enables users to post projects, submit proposals, and communicate securely, but also incorporates advanced AI features to evaluate freelancers' skills, analyze their profile strengths, and provide personalized recommendations for improvement.

Additionally, the platform empowers clients through smart proposal filtering, insightful analytics, and transparent project-progress tracking, enabling both parties to make better decisions while ensuring a more efficient, fair, and productive freelancing experience.

1.2 Project Scope

- Development of an AI-powered freelancing platform that connects clients and freelancers in a secure and user-friendly environment.
- Implementation of AI-based skill analysis to evaluate freelancers' profiles and assess their compatibility with posted jobs.
- Integration of a recommendation engine that suggests new or trending skills for freelancers to improve their profiles.
- Support for two user types – Client and Freelancer – each with personalized dashboards.
- Features for project posting, proposal submission, payments, and messaging between users.
- Integration of a secure payment gateway using PayPal with an Escrow system.
- Cloud-based file storage for project-related files and portfolio materials.
- AI-supported chat translation for multilingual communication between users.
- **Client Dashboard Insights:** Provides comprehensive project analytics, including the number of received proposals, the average proposal price, and estimated market pricing.

- **Proposal Filtering System:** Clients can filter freelancer proposals based on: Required budget range, Experience level, (Beginner, Intermediate, Expert), Delivery time, and Freelancer rating.
- **Project Progress Tracking:** Allows clients to monitor real-time project status, including task completion percentage, milestone updates, submitted files and approvals, as well as upcoming deadlines and late-task alerts.

Out-of-Scope:

- Advanced plagiarism or originality detection in submitted work.
- Automated code review or design validation beyond basic file checks.
- Blockchain-based payment or identity verification systems.
- Development of a dedicated mobile application (only responsive web supported).
- Training of proprietary deep-learning models (only pre-trained or fine-tuned models will be used).

1.3 Problem Statement

Freelance platforms face several challenges that reduce the efficiency of selecting freelancers and hinder the quality of collaboration between clients and professionals. Freelancers struggle to realistically evaluate and improve their profiles, and there is a lack of accurate mechanisms that enable them to know how suitable they are for different jobs. In addition, there is a clear gap between the skills listed in profiles and the actual skills required in the market, leading to a poor match between client needs and executors' capabilities. Clients also find it difficult to distinguish the best candidates when receiving a large number of applications, due to the absence of smart analytical tools that help them make decisions based on accurate data rather than personal judgment.

1.4 Objectives

This project aims to develop an independent AI-powered platform to improve the efficiency and transparency of the freelancer selection process. The objectives are summarized as follows:

1. Enhance freelancer profiles through intelligent tools that suggest edits, highlight strengths and weaknesses, and provide an accurate assessment of the profile's suitability for various roles.

2. Reduce the gap between stated skills and required skills through skill analysis models and evaluation techniques based on data and modern labor market standards.

3. Help clients make more accurate decisions through matching algorithms and analysis that assess the quality of applications and present the top candidates based on skills, experience, and actual project compatibility.

4. Enhance the experience for both parties (the client and the freelancer) through an easy-to-use interface and a recommendation system based on machine learning.

1.5 Methodology

Methodology (Agile Approach)

The development of the proposed system follows the **Agile methodology**, which emphasizes iterative development, continuous user feedback, and incremental improvement through short development cycles (Sprints). The methodology includes the following phases:

1. Planning

- Continuously gathering and analyzing requirements from both clients and freelancers.
- Identifying existing gaps in current freelancing platforms and establishing a prioritized backlog for each Sprint.

2. Design

- Designing the system architecture and decomposing it into core modules, including:
    - Skill Analysis Module
    - Profile Enhancement Module
    - Client–Freelancer Matching Module
- Creating data flow diagrams, database structure, and initial UI/UX prototypes.

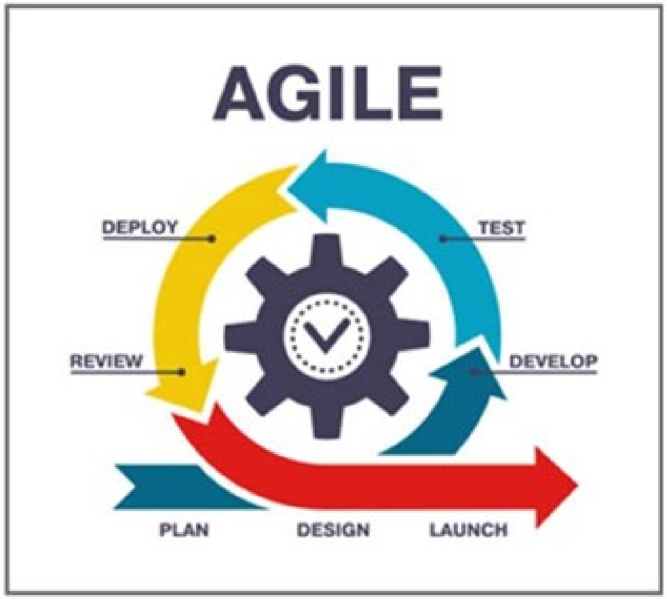


Figure 1.1 Agile Methodology.

3. Develop

- Implementing the features planned for each Sprint.
- Developing the AI components, including NLP-based skill extraction, profile analysis algorithms, and recommendation/classification models.
- Building and refining the user interface in parallel with backend development.

4. Test

- Conducting continuous testing throughout each Sprint.
- Evaluating:
    - The accuracy of AI recommendations
    - The performance of the skill analysis module
    - The usability and responsiveness of the interface
- Applying unit testing, integration testing, and scenario-based evaluation.

5. Deploy

- Deploying incremental versions of the system at the end of each Sprint.
- Ensuring that newly added features are smoothly integrated without disrupting existing functionality.

6. Review

- Reviewing Sprint outcomes with stakeholders and test users.

- Collecting feedback on performance, usability, and accuracy.
- Updating the backlog and defining improvement tasks for the next Sprint.

7. Launch (Incremental Release)

- Publishing improved and stable versions after completing multiple Sprints.
- Continuously enhancing the system to ensure high reliability, efficiency, and alignment with user needs.

Tools & Technologies:

- **Frontend:** Next.js (React)
- **Backend:** ASP.NET Core Web API
- **Database:** SQL Server
- **AI Models:** Python (scikit-learn / TensorFlow) – integrated via REST API
- **Authentication:** JWT (JSON Web Tokens)
- **Payment Gateway:** PayPal (Escrow system)
- **Cloud Storage:** Firebase / AWS S3
- **Deployment:** Docker + Azure Cloud Hosting

Architecture:

User (Client / Freelancer) → Next.js Frontend → .NET API Backend → SQL Server Database
↓
AI Module (Skill Analysis & Recommendation – Python API)
↓
Payment Gateway + Cloud Storage + Chat System

Chapter 2: Background and related work

2.1 Background

In recent years, freelancing platforms have grown rapidly as more professionals shift toward flexible and independent work models. However, despite this expansion, many platforms still rely on traditional keyword-based search, general ratings, or manual filtering to match freelancers with clients. These methods often fail to reflect true skill levels, understand the context of competencies, or accurately represent the alignment between a freelancer's profile and the actual requirements of the job market.

With advancements in modern technologies, ready-made AI APIs for text analysis and semantic understanding have become widely accessible. These tools enable deeper insight into user profiles, more accurate skill extraction, and improved evaluation processes without requiring developers to build NLP or machine-learning models from scratch. Such APIs provide a practical and efficient way to implement intelligent features within systems while significantly reducing development complexity.

This project leverages these capabilities by using pre-built AI services to analyze freelancer profiles, assess skill relevance, and extract semantic information from textual data. This approach allows the system to enhance matching accuracy, offer relevant insights, and improve the overall decision-making experience for both freelancers and clients.

2.2 Related Work

Several existing systems and research efforts have explored methods to enhance talent matching and skill assessment within freelancing and recruitment platforms. These works can generally be categorized in the following directions:

1. Keyword-based Matching in Traditional Platforms:
Platforms such as Upwork and Freelancer primarily rely on matching keywords between job descriptions and freelancer skills. Although simple, this method lacks semantic understanding and does not accurately measure skill quality or relevance.

2. Behavior-based Recommendation Systems:
Some studies have introduced recommendation systems based on user behavior, such as previous projects, browsing patterns, or client preferences. While effective in certain contexts, these systems often overlook the actual content and quality of freelancer skills.

3. Semantic Analysis for Job–Skill Matching:
A number of studies apply semantic processing to compare job requirements with freelancer skills. Although these methods improve analysis quality, they often focus solely on matching and rarely provide a comprehensive system that evaluates, scores, and improves the freelancer's profile.

The following table illustrates a comparison between the most popular traditional platforms:

Website-Name	**Advantages**	**Disadvantages**
Upwork	- Comprehensive system for project management and payments.- Large global community of clients and freelancers.- Supports long-term and specialized projects.	- Matching relies mainly on manual search and keywords.- High competition makes it difficult for new freelancers to be discovered.- No advanced analysis of freelancer skills or personalized recommendations.
Fiverr	- Simple interface for posting and browsing fixed-price services (Gigs).- Suitable for small and medium-sized tasks.- Clear categorization of services.	- No intelligent mechanism to match clients with the most suitable freelancer.- Relies heavily on ratings and sales history, not true skill assessment.- Hard for clients to evaluate quality without trial and error.

Freelancer.com	- Wide range of project categories.- Offers contests and bidding tools to attract freelancers.- Provides secure payment and milestone systems.	- Search and discovery are not enhanced by AI.- Competition is often unstructured, making selection difficult for clients.- Lacks tools for automatic profile improvement or performance analytics.

Contribution of This Project

This project distinguishes itself from previous work through:

- Integrating **ready-made AI APIs** to perform text analysis, skill extraction, and compatibility assessment efficiently.
- Focusing on **actual skill–job alignment**, not just textual similarity.
- Providing **profile evaluation and improvement suggestions**, offering actionable feedback for freelancers.
- Designing a **simple yet extensible system** suitable for academic work while delivering practical real-world value.

2.3 Summary

This chapter provided an overview of the theoretical foundations underlying the project, including the challenges faced by freelancing platforms and the limitations of traditional matching techniques. It highlighted the shift toward using ready-made AI tools to analyze text, extract skills, and understand user profiles with greater accuracy—offering a practical alternative to developing NLP or ML models from scratch.

The related-work review explored several approaches, such as keyword-based matching, behavior-driven recommendation systems, and AI-powered profile analysis. While each approach contributes valuable insights, most existing solutions address only part of the problem and often lack a fully integrated system for evaluating, improving, and matching freelancer profiles.

This chapter sets the foundation for the system design and methodology that follow, where the proposed solution integrates AI APIs to deliver more accurate skill analysis, improved profile recommendations, and enhanced matching for both clients and freelancers.

Chapter 3: Requirement analysis

3.1 Feasibility Study

Technical Feasibility:

- The project is technically feasible using the proposed technologies:
    - **Frontend:** Next.js with Redux Toolkit for state management, providing a responsive and dynamic user interface.
    - **Backend:** ASP.NET Core Web API for handling business logic and secure client-freelancer interactions.
    - **Database:** SQL Server to store user profiles, project details, proposals, and chat history.
    - **AI/ML:** Python-based APIs (scikit-learn, TensorFlow) for skill analysis, profile evaluation, and recommendation engine.
    - **Deployment:** Docker and Azure Cloud Hosting ensure scalability and smooth deployment.

Operational Feasibility:

- Users (clients and freelancers) can easily interact with the platform through a web interface.
- AI-based recommendations improve the decision-making process for both freelancers and clients.
- Secure payment integration with PayPal (Escrow) ensures trust and smooth transactions.

Economic Feasibility:

- Development costs are manageable within a student project scope.

- Using pre-trained AI models reduces training costs and computational resources.
- Cloud services and open-source frameworks lower infrastructure expenses.

3.2 Development Costs

Item	**Estimated Cost**	**Notes**
Frontend Development	Low	Next.js and Redux are open-source
Backend Development	Low	ASP.NET Core is free for students
AI/ML Model Integration	Medium	Using pre-trained models via Python APIs
Database & Hosting	Medium	SQL Server and Azure hosting (student plans)
UI/UX Design	Low	Figma
Payment Integration	Low	PayPal developer account is free
Total Estimated Cost	Low-Medium	Affordable for a graduation project

3.3 Organizational Study

- **Team Roles:**
  - Frontend Developers: Build UI components, integrate Redux state management.
  - Backend Developers: Create APIs, handle authentication, integrate AI and payment systems.

  - AI/ML Specialists: Implement skill analysis, profile evaluation, and recommendation engine.
  - UI/UX Designers: Design intuitive dashboards and responsive interfaces.
- **Communication Tools:** Slack or Microsoft Teams for coordination.
- **Project Management:** Agile methodology with 2-week sprints.

3.4 Proposed Features

Freelancer Features:

- Profile creation and skill tagging.
- AI-based skill evaluation and improvement suggestions.
- Proposal submission and tracking.
- Portfolio upload and file management.
- Chat with clients with AI-powered translation support.

Client Features:

- Project posting with detailed requirements.
- Proposal filtering by budget, experience, delivery time, and rating.
- Dashboard for analytics: average proposal price, number of proposals, top candidates.
- Real-time project progress tracking with milestones and deadlines.

Admin Features:

- User management and reporting.

- Monitoring AI recommendations and system performance.
- Payment oversight and dispute management.

3.5 Risk Management

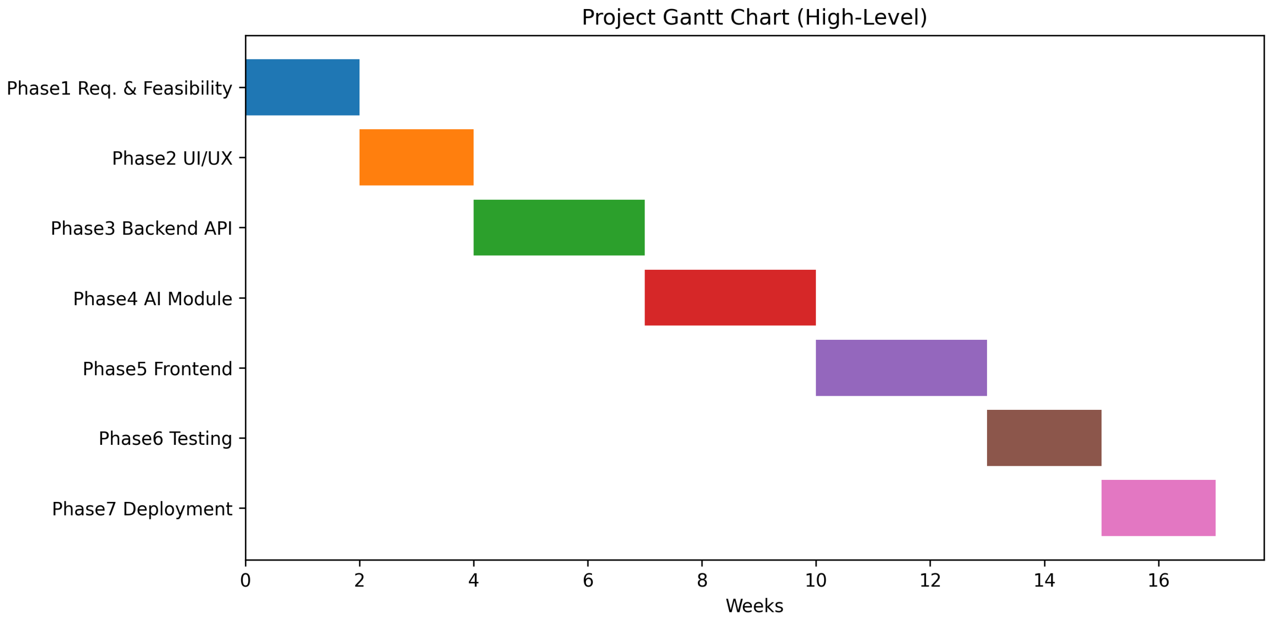
Risk	**Impact**	**Mitigation**
Delays in AI integration	High	Use pre-trained models and APIs
Cloud service downtime	Medium	Use redundant hosting and backup storage
Payment errors	High	Test PayPal sandbox and Escrow thoroughly
Feature creep	Medium	Stick to MVP scope for initial release
Security breaches	High	Implement JWT, encryption, and secure coding practices

3.6 Work plan

Phase	**Description**	**Duration**
Phase1	Requirement gathering, feasibility study, and backlog creation	2 weeks

Phase	**Description**	**Duration**
Phase2	UI/UX design, prototyping, and user feedback	2 weeks
Phase3	Backend API development (user management, project management, payment integration)	3 weeks
Phase 4	AI module integration (skill analysis, profile recommendations, proposal ranking)	3 weeks
Phase 5	Frontend implementation with Redux state management and API integration	3 weeks
Phase 6	Testing and quality assurance (unit, integration, and scenario-based testing)	2 weeks
Phase 7	Deployment, documentation, and final review	1–2 weeks

3.7 Gantt chart



Chapter 4: Requirement Gathering

4.1 Functional Requirements

- Users can register and log in securely (JWT authentication).
- Freelancers can create and update profiles with skills and portfolio files.
- Clients can post projects with detailed specifications.
- AI module evaluates freelancer skills and recommends improvements.
- Proposal system allows freelancers to submit offers for projects.
- Clients can filter proposals based on multiple criteria.
- Real-time chat system with AI translation support.
- Payment system via PayPal Escrow for secure transactions.
- Dashboard analytics for clients and freelancers.

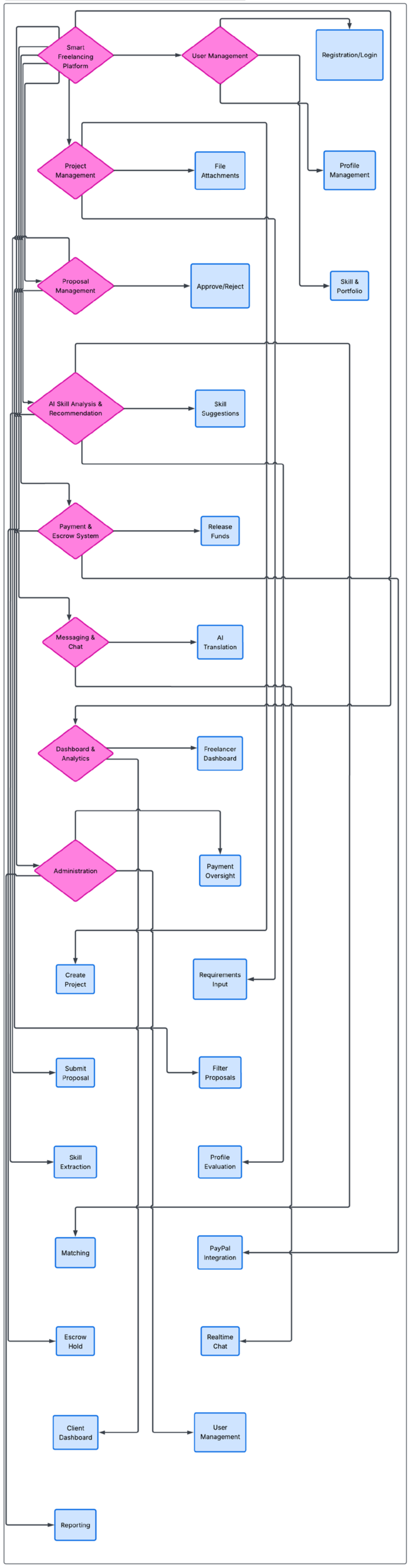
4.2 Non-functional Requirements

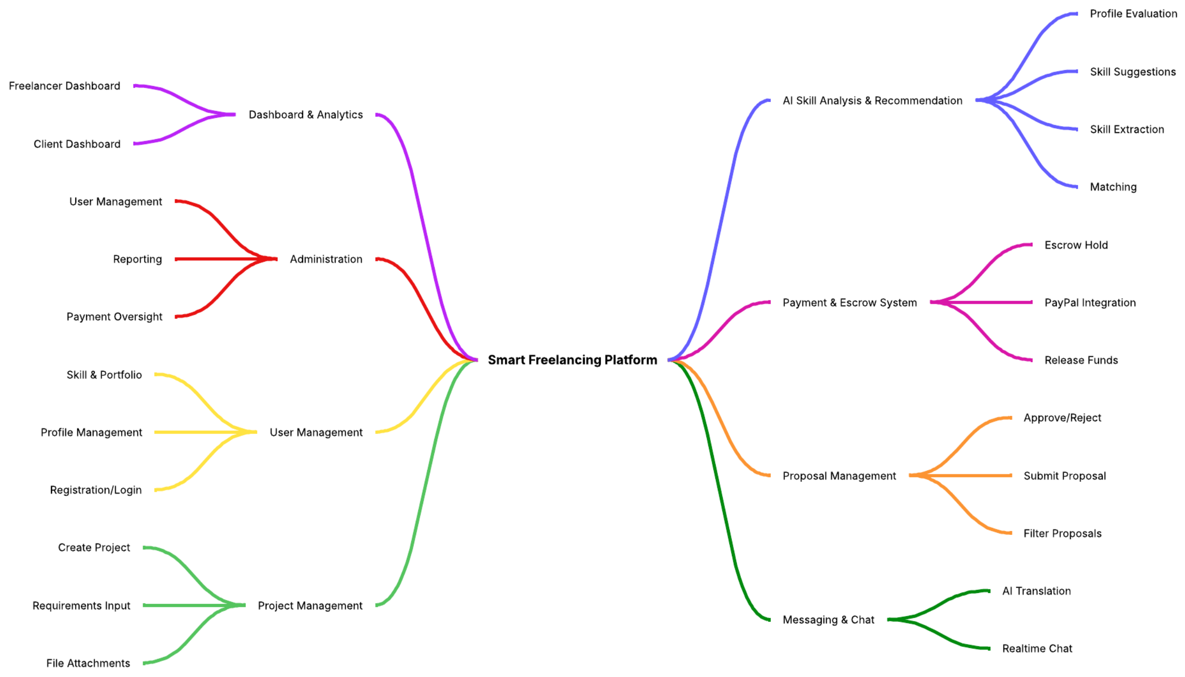
- **Performance:** Fast response time for API requests (<2 seconds).
- **Scalability:** Handle up to 10,000 concurrent users initially.
- **Reliability:** 99% uptime via Azure cloud hosting.
- **Security:** JWT authentication, encrypted communication (HTTPS), secure file storage.
- **Usability:** Responsive web design, intuitive navigation.

- **Maintainability:** Modular code for frontend, backend, and AI modules.

4.3 Functional (System) Decomposition Diagram

Smart Freelancing Platform Decomposition




Smart Freelancing Platform
Dashboard & Analytics
Freelancer Dashboard
Client Dashboard
Administration
User Management
Reporting
Payment Oversight
User Management
Skill & Portfolio
Profile Management
Registration/Login
Project Management
Create Project
Requirements Input
File Attachments
AI Skill Analysis & Recommendation
Profile Evaluation
Skill Suggestions
Skill Extraction
Matching
Payment & Escrow System
Escrow Hold
PayPal Integration
Release Funds
Proposal Management
Approve/Reject
Submit Proposal
Filter Proposals
Messaging & Chat
AI Translation
Realtime Chat

Chapter 5: System Architecture

5.1 Use Case Model

5.2 Use Case Diagram

5.3 Use Case Formats

5.4 Sequence Diagram

5.5 Class Diagram

Chapter 6: Database Design

6.1 Entity Relationship Diagram

6.2 Schema

Chapter 7: Implementation

7.1 UI design

7.2 Front End Code

7.3 Back End Code

7.4 Machine Learning Model

Chapter 8: References