AI – POWERED SKILL & PLACEMENT TRACKER

A Project Report Submitted in partial fulfillment
for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

PARDEEP KUMAR SINGH, AKASH GAUR

(23010001903, 23010001901)

Under the Supervision of

Mr. Ashish Aggarwal

Assistant Professor, DCS, GIET

GATEWAY INSTITUTE OF
ENGINEERING & TECHNOLOGY
DELHI - NCR, SONIPAT, INDIA

Department of Computer Science & Engineering
Gateway Institute of Engineering & Technology
Delhi NCR, Sonipat, Haryana, India

November, 2025

DECLARATION

I hereby, declare that the Project entitled "AI-POWERED SKILL & PLACEMENT TRACKER" submitted for the degree of Bachelor of Technology in Computer Science & Engineering is my original work. This Project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Student

Name: Pardeep Kumar Singh, Akash Gaur

Rollno: 23010001903, 23010001901

Place: Sonipat

Date:

CERTIFICATE

This is to certify that the Project entitled "AI-POWERED SKILL & PLACEMENT TRACKER" is the bonafide work carried out by "PARDEEP KUMAR SINGH, AKASH GAUR", 7th semester student of Bachelor of Technology in Computer Science & Engineering of Gateway Institute of Engineering & Technology (GIET), Sonepat affiliated to DCRUST, Murthal during the academic year 2023-26, in partial fulfillment of the requirements for the award of the degree. This Project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Guide
Guide Name: Mr. Ashish Aggarwal
Assistant Professor, GIET

Place: Sonipat
Date:

ACKNOWLEDGEMENT

I express my sincere gratitude to my project guide Mr. Ashish Aggarwal, for his/her able guidance, continuous support and cooperation throughout my project, without which the present work would not have been possible.

I would also like to give special thanks to my department Project Coordinator Mr. Anil Arora and HOD Dr. Aakash Gupta for the guidance he provided throughout the project. I shall be failing in my works if I didn't convey my thanks to my parents and friends, who provided me all the thoughts and insights.

A special word of thanks to all the respondents for sharing their valuable time with me.

ABSTRACT

Finding suitable employment after graduation is a major challenge for students, often due to a disconnect between the skills gained in academics and those demanded by industries. Many students lack clarity on what employers expect, while placement officers struggle to measure and improve overall placement readiness. Addressing this issue requires a smarter, data-driven approach that connects students' skill development with industry needs.

This project introduces an **AI-powered skill development and placement tracking system** that supports both students and placement officers. Students can upload their resumes to receive AI-driven feedback, skill gap analysis, and personalized course recommendations tailored to enhance their employability. Placement officers, on the other hand, are equipped with an interactive dashboard that provides real-time analytics on student readiness, enabling them to make informed decisions and design targeted training strategies.

The primary objective of this system is to significantly boost student employability and increase job placement success rates. By continuously monitoring progress at both individual and group levels, the platform ensures a clear alignment between academic learning outcomes and industry expectations. Ultimately, this project aims to close the gap between education and employment, preparing graduates with the right skills to succeed in the competitive job market.

TABLE OF CONTENTS

Contents **Page No**

LIST OF FIGURES

LIST OF TABLES

CHAPTER - 1
INTRODUCTION TO PROJECT

1.1 Introduction

Landing a job after graduation is tough for many students who don't know which skills employers want. Our **AI-powered skill development & placement tracker** fixes this by giving students personalized learning plans and helping universities forecast placement readiness, closing the gap between academics and industry.

1.2 Description

Our platform uses AI to help students and placement officers. Students upload their resumes, and the AI suggests ways to improve their skills. Placement officers use a dashboard to track readiness, helping them boost job placement rates. This system connects student skills directly with what employers are looking for.

1.3 Objective

The main objective of this project is to boost student employability and improve placement success. Using AI tools, the system analyzes skills, identifies gaps, and provides personalized course recommendations. It also tracks skill development at both individual and group levels while offering placement officers data-driven dashboards to make informed decisions. Overall, the project aims to align student skills with industry needs and increase successful job placements.

1.4 Scope of the Project

The current face AI-Powered Skills & Placement Tracker is designed to provide students with an intelligent platform to evaluate, enhance, and showcase their skills for better placement opportunities. The project's scope extends to both students and administrators, ensuring a comprehensive ecosystem for learning and career readiness. Students can log in securely using Firebase Authentication and access various modules such as Skill Tracker, Resume Analysis, and Course Suggestion. With the help of the Gemini API, resumes are analyzed automatically to generate an ATS score and provide improvement tips, while the YouTube API fetches recommended learning videos based on individual skill gaps. For administrators, the system provides an Admin Dashboard that displays details of all registered students, including their skills, ATS scores, and job preferences. This enables better evaluation and guidance during the placement process. The use of Firebase Firestore and Realtime Database ensures accurate and real-time data management, while React.js and Tailwind CSS contribute to a responsive and user-friendly interface. Overall, the project aims to bridge the gap between academic learning and employability by offering a data-driven, AI-assisted platform that supports students in improving their skills and achieving successful placements.

.

CHAPTER - 2
SYSTEM STUDY

2.1 Existing System

Currently, most universities and colleges rely on traditional placement processes that are largely manual. Students prepare resumes and attend placement drives without structured guidance on the skills employers demand. Placement officers track student progress through spreadsheets or basic records, which often lack real-time insights and detailed analytics. While online learning platforms provide courses, they are not personalized to an individual's career path or aligned with specific placement goals. This fragmented approach results in students being underprepared and institutions facing lower placement success rates.

2.2 Proposed System

The proposed system, an AI-powered Skill and Placement Tracker, is designed to bridge the gap between academic learning and industry requirements. Unlike the existing manual and generalized methods, this system takes a personalized approach to improving student employability. By analyzing each student's resume, the AI identifies skill strengths and weaknesses, then suggests targeted learning paths to close these gaps. This ensures that students focus on acquiring the right skills that directly match employer expectations.

In addition to helping students, the system empowers placement officers with a centralized and intelligent dashboard. This dashboard provides real-time analytics on student readiness, highlighting overall progress as well as individual skill gaps across batches. With this insight, placement officers can plan training sessions, workshops, or

industry interactions more effectively. The data-driven approach reduces guesswork and allows institutions to strategically prepare students for upcoming placement drives.

Furthermore, the system ensures continuous monitoring and improvement. Instead of a one-time assessment, it tracks skill development at both individual and group levels, enabling ongoing evaluation of workforce readiness. By aligning academic outcomes with industry demands, the system not only improves placement success rates but also enhances the reputation of institutions. Ultimately, this proposed system provides a structured, technology-driven solution to ensure students are well-prepared for the competitive job market.

2.3 Feasibility Study

The project is technically feasible as it can be built using AI/ML tools, cloud platforms, and web technologies. It is also economically viable since it requires only moderate investment while offering high long-term benefits for institutions. Operationally, the system is user-friendly and easy to adopt with minimal training. Overall, the solution is practical, sustainable, and effective for improving student placements.

ECONOMIC FEASIBILITY:

The system is economically feasible because it does not require very high costs for development or maintenance. By using open-source AI/ML tools, cloud services, and web technologies, the project can be built with moderate investment. In return, it provides high value by improving student placement success rates, which strengthens the reputation of institutions. Thus, the benefits are greater than the costs, making it cost-effective.

TECHNICAL FEASIBILITY:

The project is technically feasible as it uses technologies that are already available and widely used. AI/ML models can analyze resumes and skill data, while cloud platforms and dashboards can manage and display information in real time. Since these technologies are reliable and scalable, the system can be developed and maintained without major technical risks.

BEHAVIORAL FEASIBILITY:

The system is behaviorally feasible because it is designed to be user-friendly for both students and placement officers. Students can easily upload resumes and follow personalized recommendations, while placement officers can use simple dashboards for tracking readiness. Since the system reduces effort and gives useful insights, both groups are likely to accept and adopt it willingly

CHAPTER - 3
SOFTWARE REQUIREMENT SPECIFICATION

3.1 Software Requirement Specification

The Software Requirement Specification (SRS) for the AI-Powered Skill & Placement Tracker defines the functional and non-functional requirements necessary to build a robust, scalable, and user-friendly platform. This document acts as a bridge between the system's objectives and its technical implementation, ensuring that developers, testers, and stakeholders share a common understanding of the project goals. It outlines how the system should behave, the technologies to be used, and the constraints within which it must operate.

Functionally, the system is required to provide four main modules: Student, Admin, Placement, and the AI & Recommendation Engine. The Student module must allow users to upload resumes, certifications, and personal details while receiving AI-driven recommendations for skill improvement. The Admin module is responsible for managing accounts, ensuring security, and maintaining system integrity. The Placement module supports placement officers by offering dashboards to track skill readiness and shortlist candidates efficiently. Finally, the AI & Recommendation Engine integrates APIs such as OpenAI, Gemini, and YouTube to analyze resumes, identify skill gaps, and suggest relevant courses, projects, or learning materials.

On the non-functional side, the system must ensure scalability, reliability, and data security. The use of React.js and Tailwind CSS for the front end ensures responsive design and ease of use across devices. The back end, built on Node.js and Express.js, must handle concurrent user requests efficiently, leveraging REST APIs for seamless communication. For data management, MongoDB is chosen for structured and semi-

structured data storage, while Firebase is used for secure file storage of resumes and certificates. This hybrid database approach ensures both flexibility and high performance.

Additionally, the system is expected to maintain high availability and fault tolerance, providing uninterrupted service to students, administrators, and placement officers. AI integration must be optimized to deliver accurate, real-time recommendations without performance bottlenecks. Visual Studio Code (VS Code) serves as the development environment due to its extensibility, debugging capabilities, and cross-platform support, ensuring smooth development workflows.

In conclusion, the SRS ensures that the AI-Powered Skill & Placement Tracker is not just a placement database but an intelligent, AI-driven platform. It captures both the functional needs—such as resume analysis, personalized learning plans, and placement dashboards—and the non-functional requirements like scalability, security, and usability. Together, these specifications provide a clear roadmap for developers while assuring stakeholders that the system will meet its objectives of enhancing student employability and improving placement success rates.

3.2 Software Requirement

- **Operating System: Windows 10 or Later**

  The system will be built on windows compatible environment. The application will be web based developed using React.js technology.

- **GUI –based Software**: Visual Studio Code (IDE) or later
- **Programming Language:** React.js, Tailwind CSS, Python (FastAPI)
- **Data Base & APIs**: Firebase , MongoDB, Gemini API, Youtube API

3.3 Hardware Requirements:

- **Processor:** Intel Pentium III or higher
- **HDD:** 80GB or more
- **RAM:** 1GB or more
- **Keyboard, Mouse, Monitor**

GUI Tools and its Importance:

Graphical User Interface (GUI) helps users interact with the computer system visually through icons, menus, buttons, and text, rather than relying on command-line inputs. A GUI ensures that users with minimal technical knowledge can still effectively use the system, making it user-friendly and accessible. In the AI-Powered Skill & Placement Tracker, GUI plays a vital role in creating a smooth and professional user experience for students, administrators, and placement officers.

Types of GUI in the Project:

Front-End: The front-end of the system is built using React.js, a JavaScript library developed by Facebook, allows for the creation of modular, reusable, and easily maintainable components. In the context of this project, components such as login forms, student dashboards, skill trackers, resume upload panels, and course suggestion cards are developed as independent modules. This modular approach simplifies debugging, enhances readability, and allows future updates without affecting the entire system.React's Virtual DOM ensures faster rendering and dynamic updates, making the interface highly responsive even when processing large datasets or real-time updates from Firebase. It also enables smooth navigation between different sections of the application — such as the student dashboard, admin dashboard, course recommendation pages, and resume analysis reports — without page reloads, ensuring a continuous user experience. The styling of the application is managed using Tailwind CSS, a utility-first CSS framework that provides flexibility and consistency in design. Unlike traditional CSS frameworks, Tailwind allows developers to directly apply styling classes in the HTML or JSX elements, which speeds up the design process and ensures that the layout remains responsive across devices of all screen sizes. The combination of Tailwind's grid system, responsive utilities, and color customization allows the application to maintain a professional, modern look while remaining lightweight and fast. The front-end logic also handles the integration with external APIs such as the Gemini API for resume analysis and the YouTube API for fetching course recommendation videos. When a student uploads their resume, the front-end sends a request to the backend API endpoint, displays the results returned by the AI model, and updates the interface dynamically with the obtained ATS score and keyword analysis. Similarly, the Course Suggestion section uses skill gap data to fetch relevant video recommendations, helping students enhance their knowledge effectively. Overall, the front-end plays a critical role in ensuring that users both students and administrators can easily navigate through different functionalities. It combines aesthetic appeal with functional precision, offering a balanced interface that supports both usability and accessibility.

Back-End: The back-end GUI support is managed using Node.js, a JavaScript runtime built on Chrome's V8 engine, enables asynchronous and event-driven programming. This means that multiple user requests can be handled simultaneously without blocking the system, ensuring that the application performs efficiently under high loads. The use of Node.js makes the server lightweight, scalable, and highly suitable for real-time data applications such as this one, where multiple users may be uploading resumes or checking skill scores at the same time. Express.js, a minimalist web framework for Node.js, is used to create RESTful APIs that manage communication between the front-end and back-end systems. For example, when a student uploads a resume through the interface, the Express.js server handles the file request, interacts with the Gemini API to analyze the document, and returns the ATS score and relevant feedback to the React front-end. Similarly, it communicates with Firebase to fetch or update student records, skill data, and progress reports in real time. To ensure data privacy and secure access, the back-end uses Firebase Authentication for user login and registration, verifying email-password combinations and managing session tokens. It also interacts with Cloud Firestore and Realtime Database to store and retrieve structured data. Firestore handles static or user-specific data such as registration details, skill records, and resume analysis reports, while the Realtime Database supports dynamic data synchronization for features like live skill updates or admin tracking. Additionally, the back-end incorporates error handling and validation layers to ensure data accuracy and system stability. When interacting with external APIs, it verifies the responses and prevents issues such as malformed data or unauthorized access. The integration of CORS (Cross-Origin Resource Sharing) ensures that requests from the front-end (React.js) are securely allowed by the Node.js server, maintaining data integrity across all operations. This design ensures that the system's performance remains consistent, reliable, and scalable, even as the number of users grows. The use of asynchronous API calls and efficient routing mechanisms allows quick response times and minimal latency, providing a smooth interaction between the user interface and the server. In summary, the back-end acts as the central processing unit of the system. It connects the application to Firebase, manages API requests, and ensures that all operations — from resume analysis to course recommendation — are executed securely and efficiently. Together with the front-end, it

forms a robust full-stack architecture capable of delivering intelligent insights and personalized placement support to every user.

Importance of GUI Applications:

1. GUI makes the system **user-friendly**, allowing students, placement officers, and admins to interact easily without technical barriers.
2. It abstracts the **internal workings** of the system, so users only focus on the results and insights rather than the complexities of AI and database processes.
3. GUI applications **reduce user effort** by performing maximum tasks (like skill analysis, recommendations, and dashboards) while requiring minimal input, such as uploading a resume or selecting preferences.
4. A well-designed GUI enhances **productivity and engagement**, leading to wider adoption of the system and improved placement readiness outcomes.

Contents of Front-end:

The front-end of the AI-Powered Skill and Placement Tracker is designed with a secure login and authentication system that uses role-based access for students, placement officers, and administrators. Each user role has a personalized experience, ensuring that features are tailored to their needs. Students, for example, can manage their profiles, update academic details, and upload resumes for AI analysis, while administrators and officers access dashboards suited to monitoring and management.

Students benefit from an interactive dashboard that displays AI-analyzed skill gaps, personalized course recommendations, certification paths, and even project ideas aligned with industry demands. A resume upload and analysis interface provides real-time feedback, making it easier for students to identify strengths and weaknesses. Placement officers, on the other hand, are given tools to monitor skill readiness at both individual and group levels, generate analytics, and track overall progress to improve placement outcomes.

Administrators are equipped with controls for user management, system monitoring, and insight generation through data analytics. To further support learning and preparation, the

front-end includes a skill development section that integrates curated tutorials, recommended resources, and reminders. Notifications and alerts keep users updated on placement activities, deadlines, and recommendations. Built with React.js and Tailwind CSS, the interface ensures responsive design, smooth navigation, and modern visuals, offering a professional yet user-friendly experience across devices.

About React.js

React.js is an open-source JavaScript library developed by Jordan Walke, a software engineer at Facebook (now Meta), and was first released in 2013.

The React.js Buzzwords:

The key considerations were summed up by the React.js team in the Following list of buzzwords:

Component-Based Architecture:

React.js follows a component-based architecture, which means applications are built from small, independent, and reusable components. Each component, such as a header, footer, or login form, manages its own logic and rendering. This modular design makes development faster, easier to test, and more maintainable, since components can be reused across multiple parts of the application.

JSX (JavaScript XML):

JSX is a special syntax extension in React that allows developers to write HTML-like code inside JavaScript. Instead of separating markup and logic, JSX combines them, making the code more readable and easier to understand. For example, developers can directly write <h1>Hello World</h1> inside JavaScript, and React will compile it into JavaScript function calls.

Virtual DOM:

The Virtual DOM is one of React's most powerful features. It acts as a lightweight copy of the actual DOM and updates only the parts of the user interface that change, instead of re-rendering the entire page. This selective updating improves performance, making React apps fast and efficient, especially for applications that deal with frequent UI changes.

One-Way Data Binding:

React uses **one-way data binding**, meaning data flows in a single direction, from parent components to child components via props. This makes applications more predictable and easier to debug, since developers always know where the data is coming from and how it affects the UI.

Declarative Programming:

React promotes declarative programming, where developers describe what the UI should look like for a given state, and React automatically updates it when the state changes. This contrasts with imperative programming, where developers must manually define every step to update the UI. Declarative code is simpler, cleaner, and easier to maintain.

React Hooks:

Hooks are functions introduced in React 16.8 that allow developers to use state and lifecycle features in functional components. Common hooks include useState for managing data, useEffect for handling side effects, and useContext for working with global state. Hooks simplify coding and reduce the need for class components.

State and Props:

In React, state represents data that can change over time within a component, while props are inputs passed from one component to another. State makes components dynamic and interactive, while props allow data sharing between components. Together, they form the backbone of how React manages data and UI.

Context API:

The Context API is a built-in feature in React that allows sharing global data across components without the need for "prop drilling" (passing props manually through multiple levels). It is commonly used for managing themes, authentication, or user preferences across the app.

Redux / State Management:

Redux is an external library often used with React for large-scale state management. While React handles local component state well, Redux centralizes the application's state in a single store, making it easier to manage and debug in bigger projects.

Single Page Application (SPA):

React is often used to build Single Page Applications (SPAs), where the content updates dynamically without reloading the page. This provides users with a smoother, app-like experience, as navigation feels instant and continuous.

Visual Studio Code IDE

Visual Studio Code (VS Code) is a free, open-source, cross-platform code editor developed by Microsoft. Unlike traditional IDEs, VS Code is lightweight but highly extensible, allowing developers to add features such as debugging, Git integration, code linting, and language-specific tools via extensions. It supports multiple programming

languages including JavaScript, Python, Java, C++, and frameworks such as React.js, Node.js, and more. VS Code is widely used for both front-end and back-end development, making it ideal for projects like the AI-Powered Skill and Placement Tracker. VS Code offers many features for application development, such as:

Integrated Development Environment (IDE): VS Code provides an integrated development environment with essential tools for software development. These include a code editor with syntax highlighting and auto-completion, debugger for identifying and fixing issues, terminal for executing commands, and source control integration for version management. The editor also allows visualization of project files and directories, making navigation simple and efficient. Extensions further enhance the IDE, allowing developers to customize the environment for their specific project requirements.

Rapid Application Development (RAD): VS Code supports rapid application development through its lightweight interface, powerful extensions, and built-in IntelliSense feature. Developers can quickly write, test, and debug code, reducing development time. For front-end projects like React.js dashboards, VS Code's live server extensions allow real-time preview of changes, helping developers build applications faster and more efficiently.

Event-Driven programming language: VS Code facilitates event-driven programming by allowing developers to write, test, and debug event-based code efficiently. For instance, in a React.js application, user interactions like button clicks, form submissions, or API calls can be handled through event handlers in a streamlined coding environment. Integrated debugging and console outputs make tracking events and testing responses easier. The three main parts of event-driven programming are:

The event: An event is an object that gets generated when user does something such as mouse click, dragging, pressing a key on the keyboard etc.

Source of the event: The component where the event has occurred is the source of the event. For example, if a user click on a submit button then the source of this event is the submit button.

The event listener: An event listener is attached to a component and contains the methods/functions that will be executed in response to an event. For example, if a user clicks on a button, then the button's event listener (actionPerformed) will execute some code in response to this event.

Starting VS Code: To start VS Code, download and install it from the official website, then launch the application. Developers can open a project folder, access terminals, add extensions, and start coding immediately. The user-friendly interface displays the Explorer for project files, a Code Editor in the center, Activity Bar for navigation, and a Status Bar providing real-time information about the project. With its versatility, VS Code has become a preferred tool for modern web and software development.

The first screen of VS Code will display in figure 3.1 below.

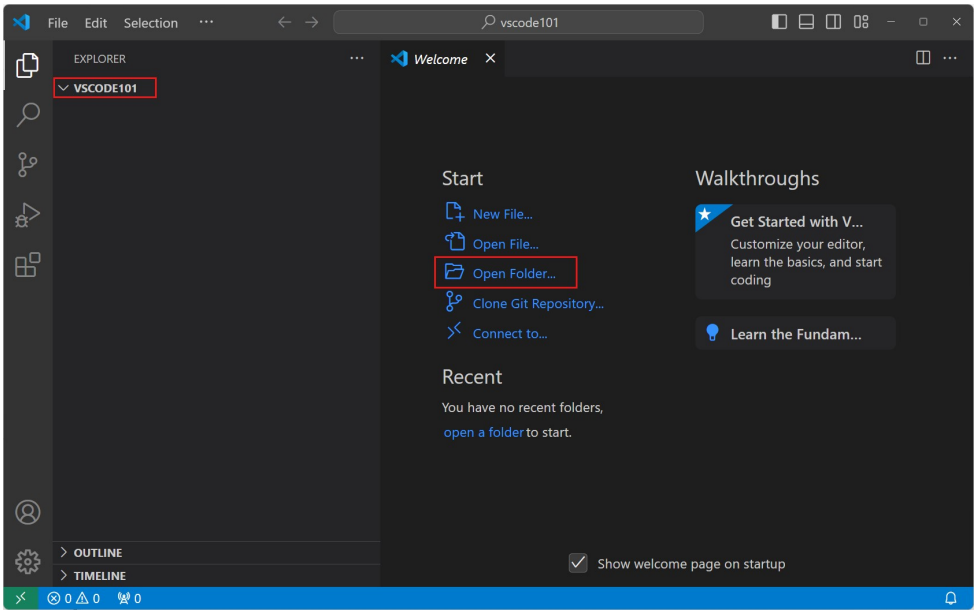


Figure 3.1: The first screen of VS Code IDE

Components of Visual Studio Code IDE:

Some popular components of VS Code IDE are:

1. **Explorer:** Shows all project files and folders for easy navigation.
2. **Editor Window:** The main area where you write and edit code.
3. **Terminal:** Built-in command-line tool for running code and commands.
4. **Extensions:** Add extra Features like debuggers, linters, or framework support.
5. **Activity Bar:** Provides quick access to Explorer, Search, Source Control, and more.
6. **Status Bar:** Displays real-time info like errors, Git branch, and file details.

Creating a Project in VS Code

To create a new GUI application project, we require following steps:
1. Open **VS Code**, click **File → New Folder**, and name it (e.g., "SkillTracker").
2. Go to **File → Open Folder** and select the new folder.
3. Create files inside (like index.html, app.js, style.css) or initialize a framework project with commands in the **Terminal** (e.g., npx create-react-app myapp).
4. Start coding—your project is ready!

BACK-END

Backend: It is implemented using Node.js, Express.js and Python, ensuring robust and scalable server-side operations. Node.js handles multiple user requests efficiently with its asynchronous, non-blocking architecture, while Express.js provides a flexible framework for building REST APIs or Fast APIs. These APIs enable smooth communication between the front-end and back-end, such as processing resume uploads, analyzing them with AI services, and sending results back to the interface. This architecture ensures high performance, security, and reliability across the application.

Data Base is a collection of tables and table is a collection of records in a tabular form i.e. in row and columns format.

Ex: **Table: Student**

Student_id	Number
First_Name	Text
Last_Name	Text
Job_Role	Text
E-mail	Text
Password_hash	Number
Contact_No	Number
Address	Text

In the above table, a row represents relationship b/w different set of values, so a table is also called a Relation. A row in the table is called tuple or record & a column in the table is called attribute or field.

NoSQL (Document-Oriented Database):- Cloud Firestore is a NoSQL, document-oriented database. Its structure is based on a hierarchy of collections and documents.

FIREBASE: Firebase is a cloud-based platform by Google that provides real-time databases, authentication, and secure file storage. It allows developers to store and sync data instantly across devices. With its scalable and serverless setup, you don't need to manage backend servers. Using Firebase ensures reliable, fast, and easy integration for storing student resumes, certificates, and skill data in your project.

i. It helps us to create any type & size of database.
ii. It helps us to manipulate the table data i.e. we can insert, delete or modify any specific information from any specific table.

iii. It helps us to query the table for retrieving (displaying) any specific information from any specific database.

iv. It helps us to protect or share the table data with multiple users.

v. It helps us to maintain Data Dictionary & system catalog.

Firebase Firestore

Firestore is used as the primary database for storing and managing user-related data such as student details, skill records, ATS scores, and course information. It provides a flexible, cloud-based NoSQL data structure that allows efficient reading, writing, and real-time updates. Firestore ensures that all information stored remains synchronized across both the student and admin dashboards without manual refresh.

Firebase Realtime Database

The Realtime Database is implemented for operations that require live data synchronization. It allows instant updates whenever changes are made in user activities, such as course recommendations or skill updates. This ensures that any modification by a student or admin is immediately reflected across the application, providing a seamless and dynamic user experience.

Firebase Authentication

Firebase Authentication is used to handle secure user login and registration processes. It ensures that only verified users can access the system by validating their email and password credentials. This feature provides an additional layer of security, preventing unauthorized access to sensitive student and admin data while maintaining a smooth sign-in experience for legitimate users.

CHAPTER - 4
PROGRAM DESIGN

The **AI-Powered Skill & Placement Tracker** is designed to manage students, admins, placement officers, and AI-driven skill analysis. The system helps students improve their employability by uploading resumes, certifications, and projects, while placement officers and admins monitor readiness and track placement data. The AI engine analyzes uploaded data, detects skill gaps, and provides personalized recommendations for courses, certifications, and project ideas.

The project contains total 12 Frames as

1. Login	(Login Frame)
2. Dashboard	(Main Frame)
3. Student Profile	(Profile Frame)
4. Resume Upload	(Upload Frame)
5. Skill Analysis	(Analysis Frame)
6. Course Recommendations	(Recommendation Frame)
7. Placement Dashboard	(Placement Frame)
8. Admin Management	(Admin Frame)
9. Student Progress Tracking	(Tracking Frame)
10. AI & Project Suggestions	(Project Frame)
11. Notifications	(Notification Frame)
12. Reports	(Reports Frame)

4.1 Database Design of the Project

Table 4.1: Student

Student_id	Number
First_Name	Text
Last_Name	Text
Job_role	Text
E-mail	Text
Password	Number

Table 4.2: Admin

Admin_id	Number
E-mail	Text
Password	Number
Last_login	Datetime

4.2 DATA FLOW DIAGRAM

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination. The data flow diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels. This processes of representing different details level is called “leveling” or “partitioning” by some data flow diagram advocates. Like a road map, there is no starting point or stop

point, any time or timing, or steps to get somewhere. A road map shows all existing or planned roads because the road is needed.

Details that is not shown on the different levels of the data flow diagram such as volumes, timing, frequency, etc. is shown on supplementary diagrams or in the data dictionary. For example, data store contents may be shown in the data dictionary. Data Flow Diagram (DFD) uses a number of symbols to represent the systems. Data Flow Diagram also known as 'Bubble Chart' is used to clarify system requirements and identifying the major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the level of details.

The following diagram 4.1 illustrates the notations and symbols used to construct the DFD:

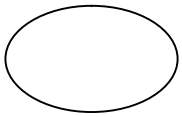
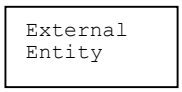
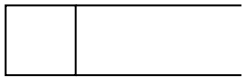
	A circle or bubble represents a process transform Incoming Data flow(s) into outgoing flows.
	A producer or consumer of information that resides outside the bounds to be modeled.
	The arrowhead indicates the direction of flow.
	The table in which information will be stored ultimately.

Figure 4.1: Illustrates the notations and symbols used to construct the DFD

The figure 4.2 Below shows the Context Level DFD for Airline Reservation Process.

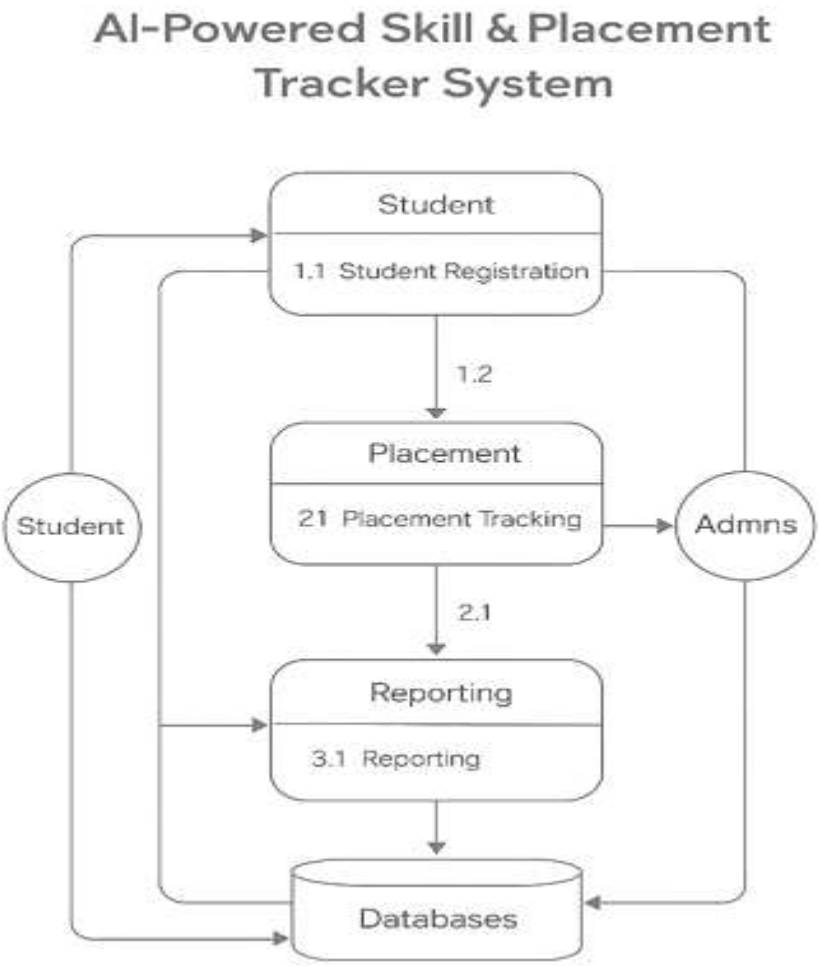


Figure 4.2: The Context Level DFD for AI-Powered Skill & Placement Tracker

4.3 E - R DIAGRAM

The Entity-Relationship (E/R) data model represents different entities, attributes and their relationship graphically. The E/R data model was introduced by P.P. Chen.

Components of E/R model:

The Entity relationship model contains following three components: Entity, Attribute and Relationship

1. Entity: An entity is a real-world data item having some characteristics and behaviour called properties. For Example student is an entity having properties Rollno, Name and Marks. Similarly, Employee is also an entity having properties Empno, Ename & Salary.

An **entity type (set)** is a set of entities having similar type i.e, entites having common properties are represented by an Entity Set. For example group of students having properties Rollno, Name and Marks are represented by an Entity set Student. Similarly, the group of employees having properties Empno, Ename and Salary are represented by an entity set Employee.

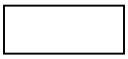
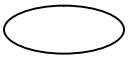
An **entity instance** is an instance of entity set i.e, an entity instance is a specific individual, thing or object. For example "Smith" is an instance of entity type Student, Similarly 10000 is an instance of entity type Employee.

2. Attributes: An entity is represented by set of properties. These properties are called attributes. For example student entity contains following properties Rollno, Name and Marks. Hence attributes of student entity set are Roll, Name and Marks. Similarly, Employee entity contains properties Empno, Ename and Salary, Therefore employee entity has attributes Empno, ename and salary.

3. Relationship: A relationship is an association among several entities. For example there is relationship between Vendor and Item as Vendor supply Item. Similarly there is relationship between teacher and student as teacher teaches the student.

E/R data model symbols:

As E/R data model is the pictorial representation of different entities, attributes and their relationship, therefore it provides different graphical symbols for representing different components.

Component	Symbol	Name
Entity		Rectangle
Attribute		Ellipsis/Oval
Relationship		Diamond

E-R model for Regestration System is shown in figure 4.3 below:

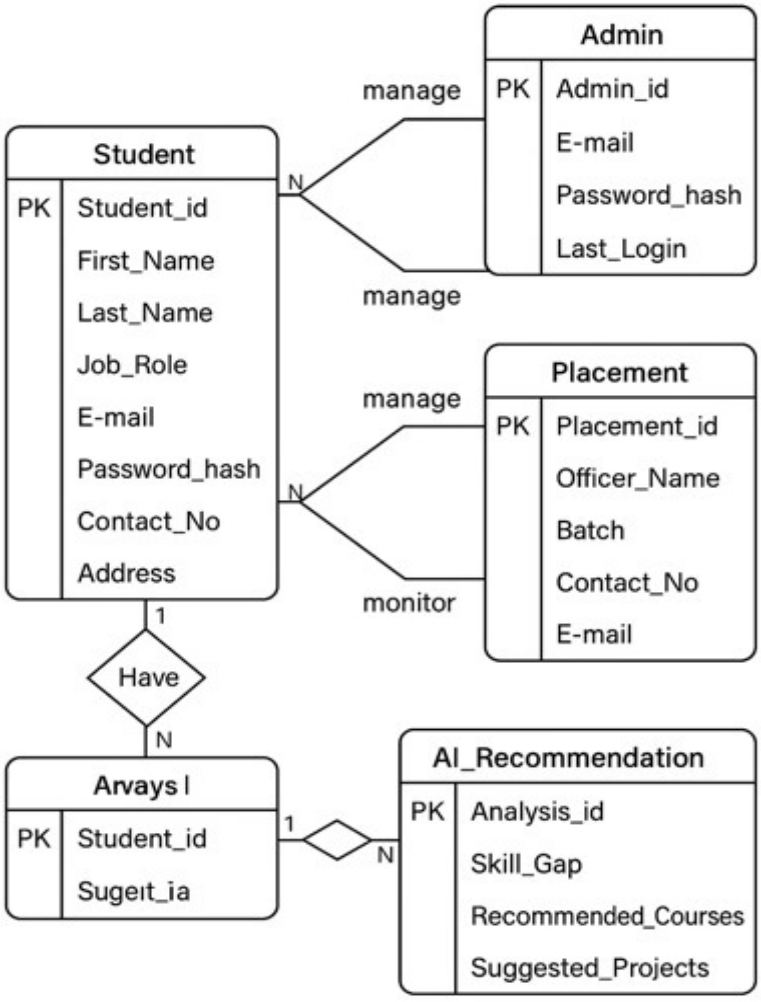
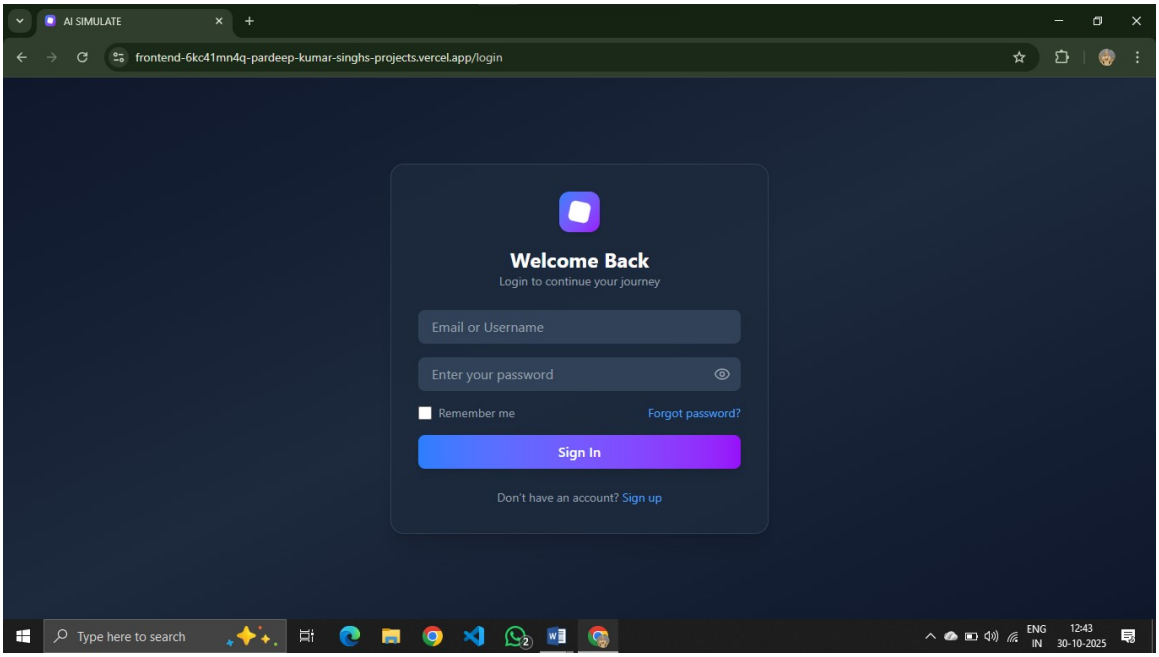


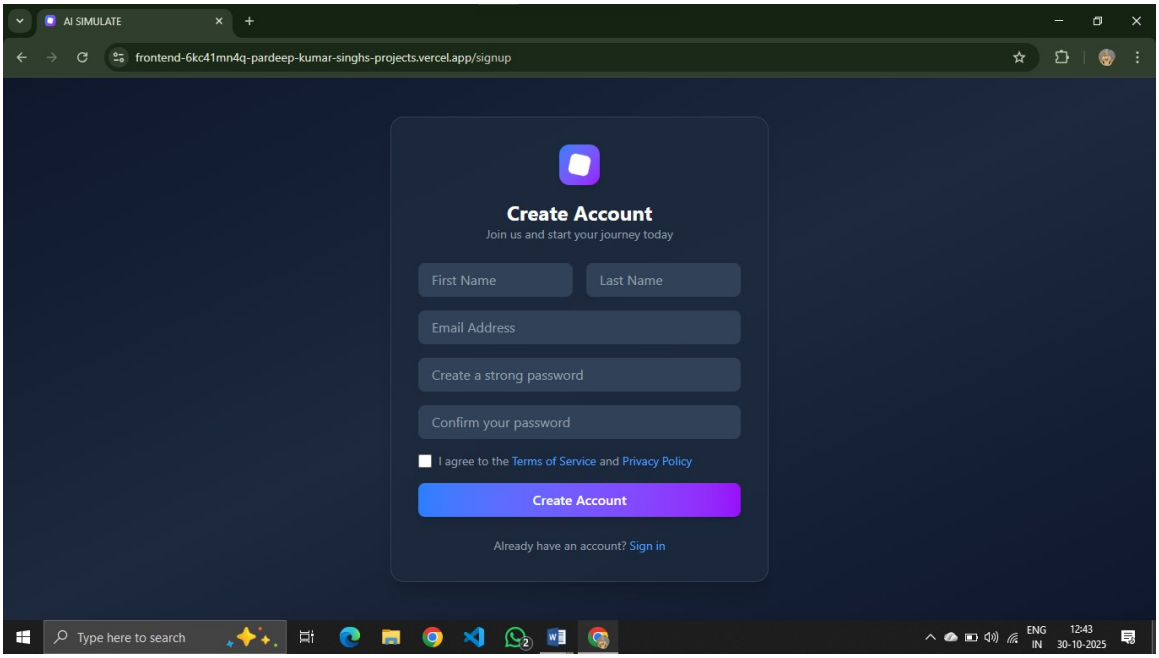
Figure 4.3: E-R model for AI-Powered Skill & Placement Tracker

4.4 Screen Shots

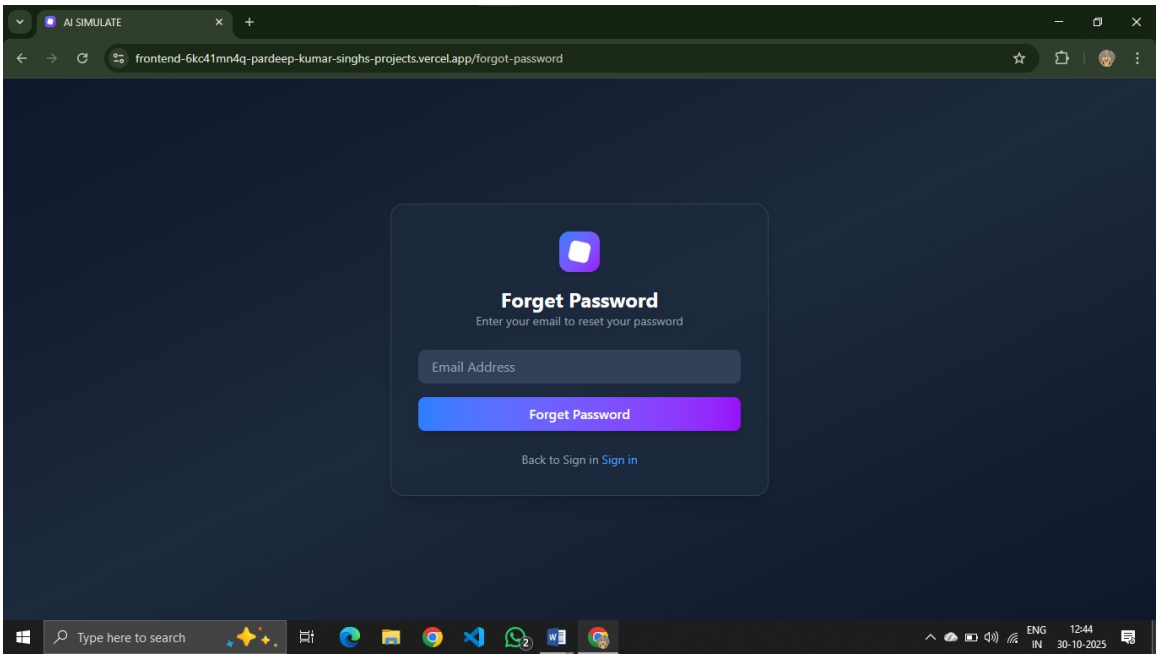
Login Frame



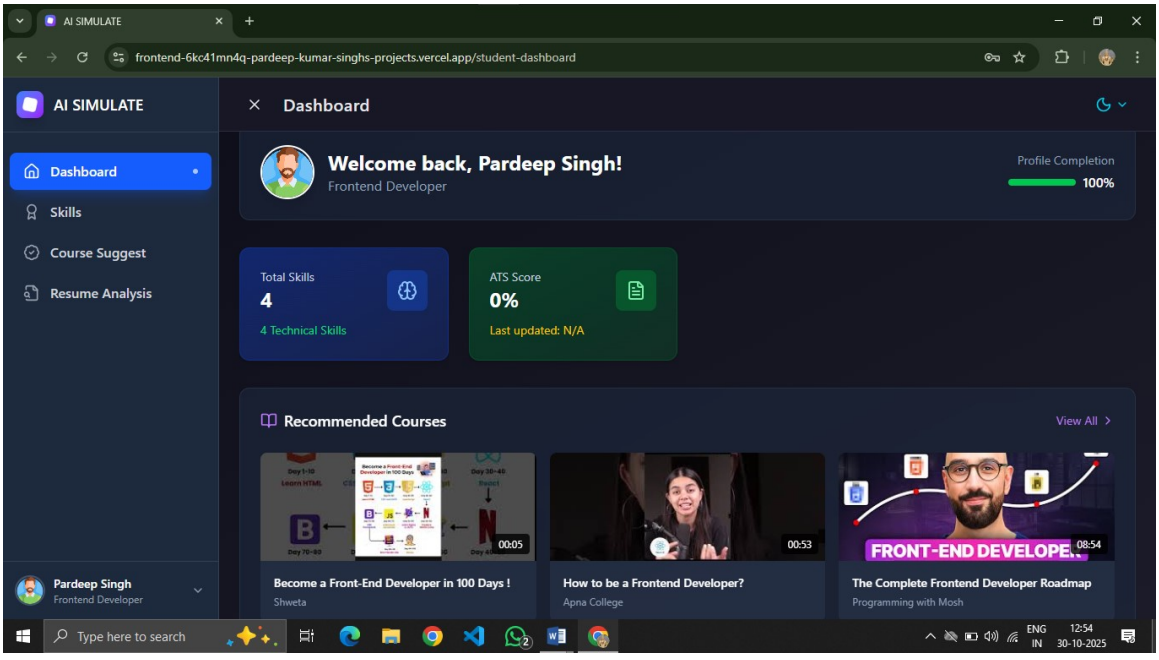
Signup Frame



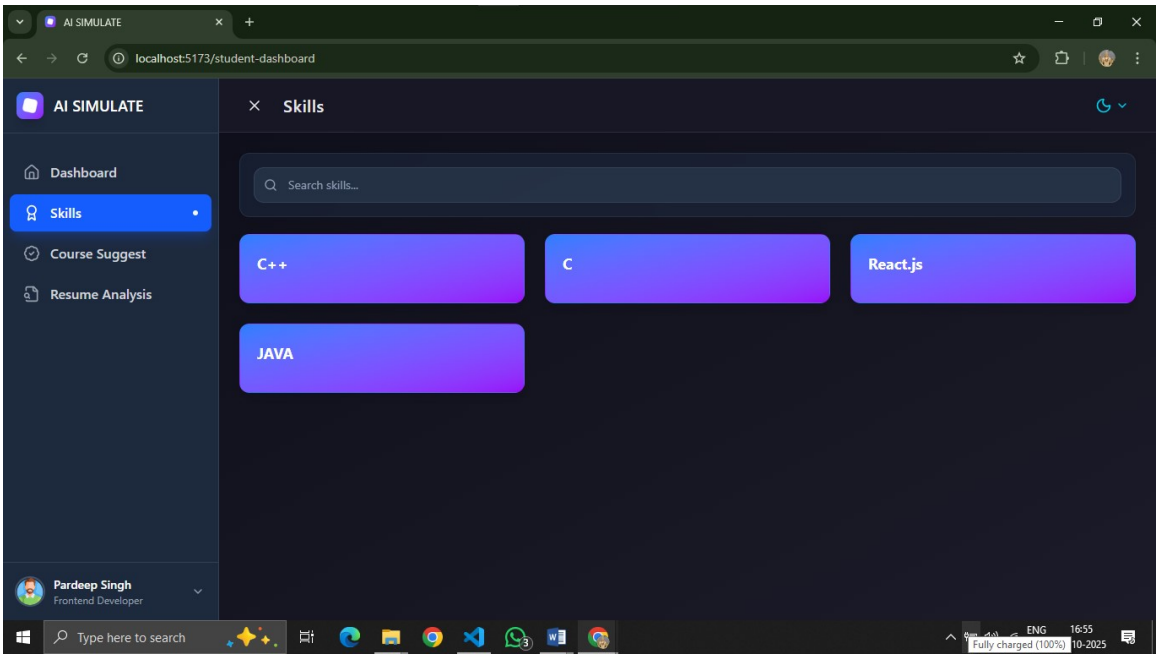
Forget Frame



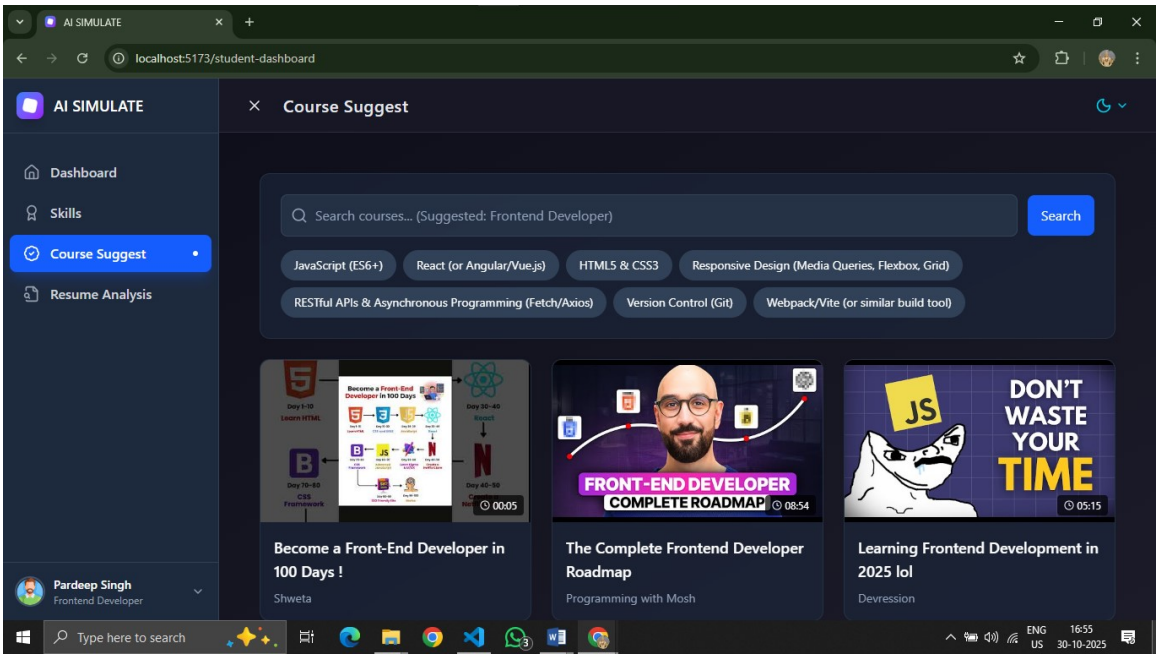
Student Dashboard Frame



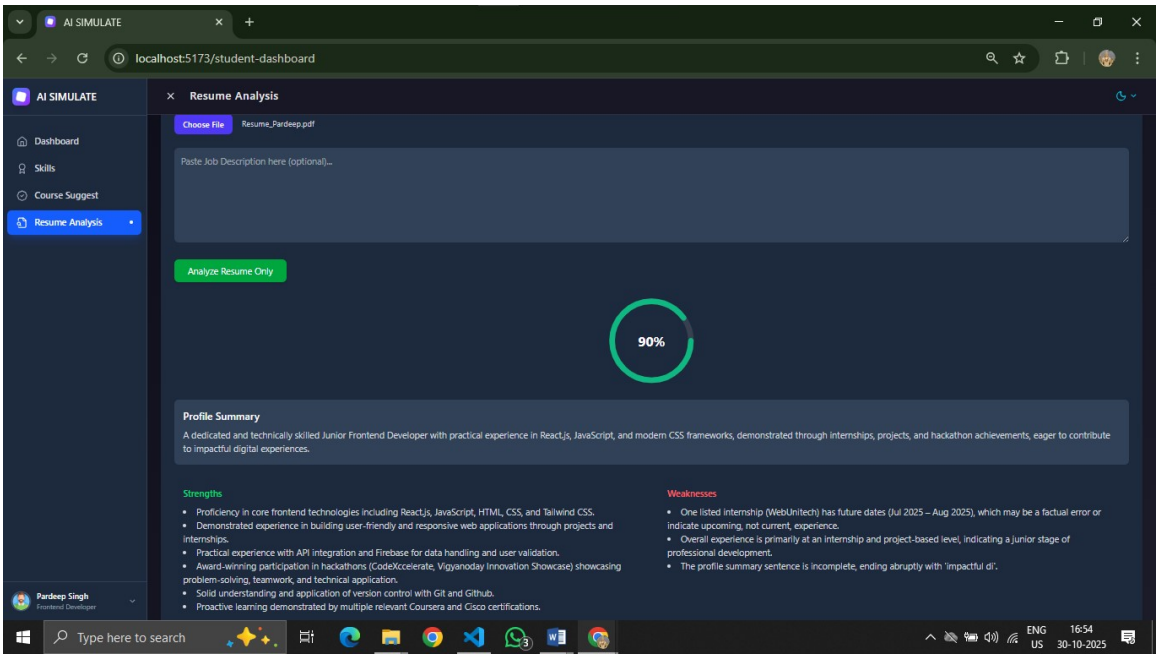
Skills Frame



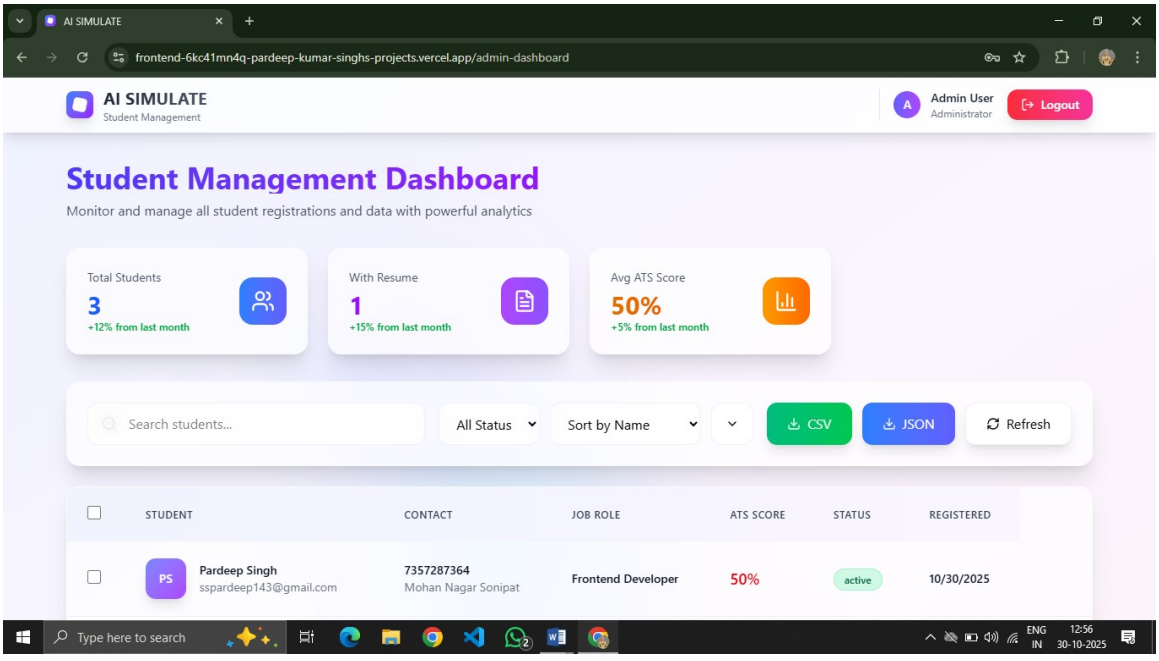
Course Suggest Frame



Resume Analysis Frame



Admin Dashboard Frame



CHAPTER - 5

System Development & Implementation

The System Development and Implementation phase is one of the most crucial stages in software engineering, where the planned design and architecture are transformed into a working application. For the AI-Powered Skills & Placement Tracker, this phase involved developing both front-end and back-end components, integrating Firebase services, implementing APIs, and finally hosting the complete project on a cloud platform for real-time use.

5.1 Development Environment

The project is developed using Visual Studio Code (VS Code) as the primary code editor because of its lightweight design, flexibility, and support for multiple frameworks. The front-end is developed using React.js and styled with Tailwind CSS to ensure a responsive and modern interface. The back-end operations, including API communication, are managed using Python (FastAPI).

For data storage and authentication, Firebase is used due to its ease of integration, real-time synchronization, and scalability. The Gemini API is utilized for analyzing resumes and generating an ATS (Applicant Tracking System) score, while the YouTube API is integrated to suggest learning resources and courses to students.

5.2 Implementation Procedure

The implementation started by setting up the Firebase project, creating Firestore collections for storing user data, and configuring Authentication for secure access. Then, the React.js front-end was connected to Firebase using SDKs for real-time data retrieval.

Modules such as User Registration, Skill Tracker, Resume Analysis, and Course Suggestion were implemented sequentially. Each module was tested individually before final integration. The Admin Dashboard was later implemented to allow administrators to

monitor student progress, view registered users, and assess performance based on ATS scores and skill data.

The final step included API integration, where Gemini API processed resumes and YouTube API fetched learning videos dynamically. Once the system was tested locally, it was deployed using Vercel for the front-end and Render for the back-end.

5.3 Integration with Firebase

Firebase plays a key role in this project by managing authentication, data storage, and real-time updates.

- Firebase Authentication: Used for secure login and registration of students and admins.
- Firestore Database: Stores structured data such as user profiles, skill records, and resume analysis results.
- Realtime Database: Manages dynamic data such as activity tracking, course updates, and dashboard metrics.

Firebase ensures that every update made by the user is instantly reflected across the application, providing a seamless user experience.

5.4 API Integration (Gemini & YouTube)

To make the system intelligent, external APIs were integrated:

- Gemini API: Processes uploaded resumes and returns an ATS score along with improvement suggestions. This helps students understand how well their resume aligns with industry standards.
- YouTube API: Fetches video tutorials and learning resources based on each student's weak skill areas identified by the AI analysis.

These integrations make the system AI-driven and more interactive, helping users continuously enhance their skills.

CHAPTER - 6
TESTING

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Testing is a very important but an expensive activity. The purpose of testing can be quality assurance, verification and validation or reliability estimation.

Level of Testing:

Testing can be performed in following three levels.

i. Unit testing
ii. Integration testing
iii. System testing

Unit Testing

Each program component is tested on its own, isolated from the other components in the system. Such testing, known as module testing, component testing or unit testing, verifies that the component functions properly with types of inputs expected from studying the component design.

Integrated Testing

When all collecting components have been unit tested, the next step is ensuring the interfaces among the components are defined and handled properly. Integration testing is the process of verifying that the system components work together as described in the system and program design specifications.

System Testing

System testing focuses on a complete, integrated system to evaluate compliance with specified requirements. Tests are made on characteristics that are only present when the entire system is run.

6.1 TEST CASE DESIGN

The design of tests for software and other engineered products can be as challenging as the initial design of the product itself. Yet, for reasons that we have already discussed, software engineers often treat testing as an afterthought, developing test cases that may "feel right" but have little assurance of being complete. Recalling the objectives of testing, we must design tests that have the highest likelihood of finding the most errors with a minimum amount of time and effort.

A rich variety of test case design methods have evolved for software. These methods provide the developer with a systematic approach to testing. More important, methods provide a mechanism that can help to ensure the completeness of tests and provide the highest likelihood for uncovering errors in software.

Any engineered product (and most other things) can be tested in one of two ways:

(1) Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function;

(2) knowing the internal workings of a product, tests can be conducted to ensure that "all gears mesh," that is, internal operations are performed according to specifications and all internal components have been adequately exercised.

Test case design for "AI Powered Skills & Placement Tracker":

We can use various test cases for checking the correctness of our project – Airways System. These test cases are listed below:

1. Checking user authentication: Ensures only valid users can log in and unauthorized users are denied access.

2. Checking User Registration: Verifies new users can register successfully and duplicate accounts are restricted.

3. Checking Firebase Data Storage: Confirms all user data (skills, resume, ATS score) is correctly saved and retrieved from Firebase.

4. Checking Resume Analysis using Gemeine API: Validates resume upload and analysis, ensuring accurate ATS score and keyword suggestions.

5. Checking YouTube API Video Fetch: Checks that relevant videos load correctly based on user skill gaps or selected topics.

6.2 Types of Testing

The first test approach is called black-box testing and the second, white-box testing. When computer software is considered, *black-box testing* alludes to tests that are conducted at the software interface. Although they are designed to uncover errors, black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information (e.g., a database) is maintained.

A black-box test examines some fundamental aspect of a system with little regard for the internal logical structure of the software.

White-box testing of software is predicated on close examination of procedural detail. Logical paths through the software are tested by providing test cases that exercise specific sets of conditions and/or loops. The "status of the program" may be examined at various points to determine if the expected or asserted status corresponds to the actual status.

WHITE-BOX TESTING: White-box testing, sometimes called *glass-box testing,* is a test case design method that uses the control structure of the procedural design to derive test cases. Using white-box testing methods, the software engineer can derive test cases that (1) guarantee that all independent paths within a module have been exercised at least once, (2) exercise all logical decisions on their true and false sides, (3) execute all loops at their boundaries and within their operational bounds, and (4) exercise internal data structures to ensure their validity.

BLACK-BOX TESTING: *Black-box testing,* also called *behavioral testing,* focuses on the functional requirements of the software. That is, black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black-box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

Black-box testing attempts to find errors in the following categories: (1) incorrect or missing functions, (2) interface errors, (3) errors in data structures or external data base access, (4) behavior or performance errors, and (5) initialization and termination errors.

Unlike white-box testing, which is performed early in the testing process, black-box testing tends to be applied during later stages of testing. Because black-box testing purposely disregards control structure, attention is focused on the information domain.

CHAPTER - 7
SYSTEM IMPLEMENTATION

After receiving user acceptance for the newly developed system, the implementation phase begins. This is the stage where the designed and tested system is deployed into a real working environment. The main objective of this phase is to ensure that the system functions smoothly and meets user expectations. The major steps involved are:

- Deployment of Backend and Frontend
- Database Setup and Configuration
- User Training
- System Documentation

The required hardware, software, and dependencies are set up to make the system fully operational. The backend (Python) and frontend (React) are deployed on the respective servers or cloud platforms. The database is configured with proper authentication, security, and backup procedures to ensure data safety and reliability.

During this phase, the complete system is made available to users. Once the system is deployed successfully, user training sessions are conducted. These sessions cover:

- How to log in and access the system
- How to input, update, and manage data
- How to view and analyze system-generated results
- How to generate reports and interpret outputs

After users are trained and confident with the new system, the organization transitions from the old method to the new automated system. This transition is known as **Changeover**, and the following strategies can be adopted:

1. **Direct Changeover:** The old system is completely replaced by the new one. This approach is quick but risky and requires complete testing and proper training before implementation.

2. **Parallel run**: Both the old and new systems are operated simultaneously for a specific period. The same data is processed in both systems to compare accuracy and reliability.

▪ Manual results can be compared with the results of the computerized system.

▪ The operational work is doubled.

▪ Failure of the computerized system at the early stage does not affect the working of the organization, because the manual system continues to work, as it used to do.

3. **Pilot run**: The new system is tested in a limited area or with sample data before full implementation. The results are analyzed and compared with the old system to ensure accuracy. This method is less risky and cost-effective, allowing early detection and correction of errors before the full rollout.

CHAPTER - 8
DOCUMENTATION

The documentation of the system is also one of the most important activities in the system development life cycle. This ensures the continuity of the system. Generally following two types of documentations are prepared for any system.

- User or Operator Documentation
- System Documentation

8.1 Operational Document

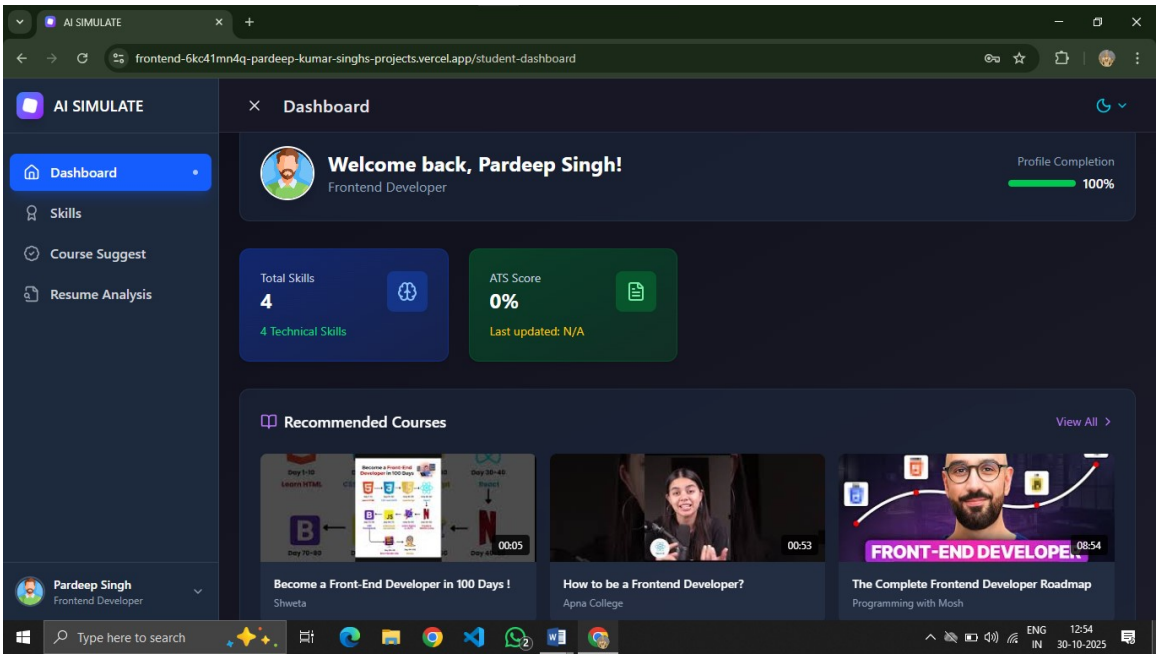
The system documentation contains the details of system design, programs, their coding, system flow, data dictionary, process description, etc. This helps to understand the system and permit changes to be made in the existing system to satisfy new user needs.

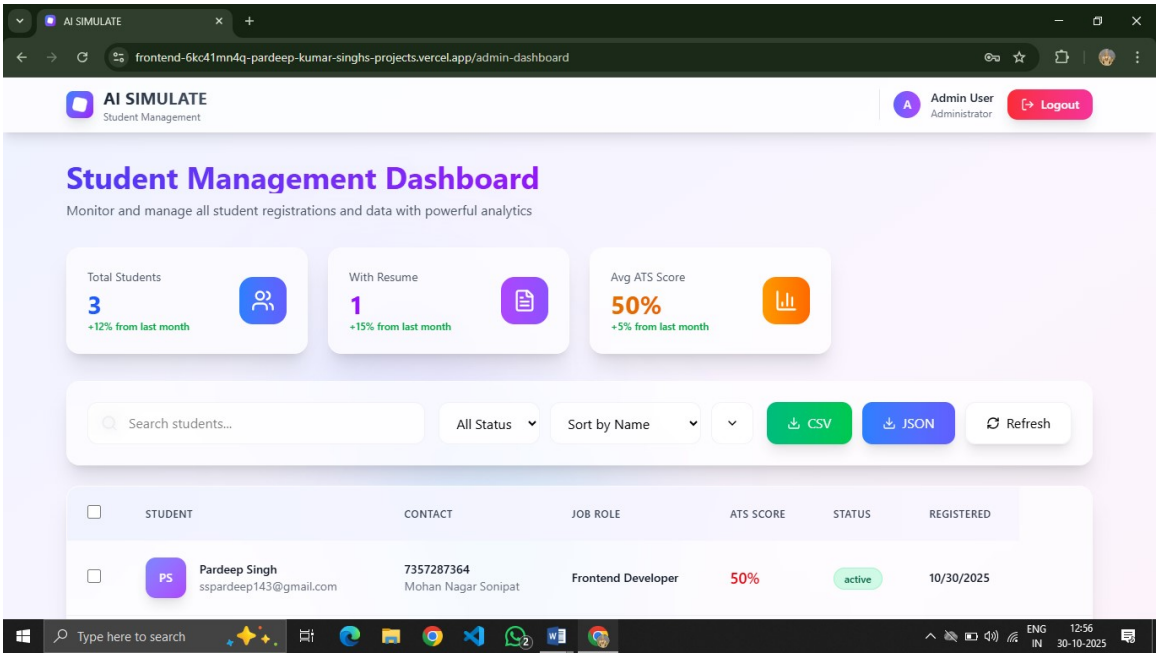
8.2 User Manual

The user documentation is a complete description of the system from the user's point of view detailing how to use or operate the system. It also includes the major error messages likely to be encountered by the user.

User Manual for AI Powered Skills & Placement Tracker:

We can start our AI Powered Skills & Placement Tracker project by entering the correct user email and password in the login screen. After successful authentication, the user is redirected to the main dashboard of the project, as shown below:


AI SIMULATE
frontend-6kc41mn4q-pardeep-kumar-singhs-projects.vercel.app/student-dashboard
AI SIMULATE
Dashboard
Skills
Course Suggest
Resume Analysis
Dashboard
Welcome back, Pardeep Singh!
Frontend Developer
Profile Completion
100%
Total Skills
4
4 Technical Skills
ATS Score
0%
Last updated: N/A
Recommended Courses
View All
00:05
Become a Front-End Developer in 100 Days !
Shweta
00:53
How to be a Frontend Developer?
Apna College
FRONT-END DEVELOPE..
08:54
The Complete Frontend Developer Roadmap
Programming with Mosh
Pardeep Singh
Frontend Developer
Type here to search
ENG
IN
12:54
30-10-2025


AI SIMULATE
frontend-6kc41mn4q-pardeep-kumar-singhs-projects.vercel.app/admin-dashboard
AI SIMULATE
Student Management
Admin User
Administrator
Logout
Student Management Dashboard
Monitor and manage all student registrations and data with powerful analytics
Total Students
3
+12% from last month
With Resume
1
+15% from last month
Avg ATS Score
50%
+5% from last month
Search students...
All Status
Sort by Name
CSV
JSON
Refresh
STUDENT
CONTACT
JOB ROLE
ATS SCORE
STATUS
REGISTERED
PS
Pardeep Singh
sspardeep143@gmail.com
7357287364
Mohan Nagar Sonipat
Frontend Developer
50%
active
10/30/2025
Type here to search
ENG
IN
12:56
30-10-2025

In this project, two separate dashboards are designed Student Dashboard and Admin Dashboard — each serving different purposes and functionalities.

From the Student Dashboard, the user can access multiple sections such as Home, Skills, Course Suggestion, and Resume Analysis.
The Home section provides an overview of the student's progress and key placement readiness indicators.

The Skills section allows students to add, update, and track their technical and soft skills.

The Course Suggestion section displays AI-generated learning recommendations, fetched using the YouTube API, based on the student's skill gaps or resume analysis results.

The Resume Analysis section enables students to upload their resumes, which are analyzed through the Gemeine API to generate ATS scores, identify missing keywords, and offer improvement tips for better placement prospects.

The Admin Dashboard is designed to monitor and manage student-related information. It includes detailed data on student registrations, job roles, ATS scores, and overall skill performance. The admin can view each student's progress, verify their submitted resumes, and evaluate readiness levels for placement opportunities.

This dual-dashboard structure ensures smooth interaction between students and administrators, enabling effective monitoring, guidance, and data-driven placement preparation.

CHAPTER - 9
CONCLUSION & FUTURE SCOPE

The AI-Powered Skill Development & Placement Tracker is a comprehensive solution designed to bridge the gap between student skill development and industry requirements. By leveraging artificial intelligence, the system provides personalized skill assessments, recommends relevant learning resources, and predicts placement readiness.

For students, it serves as a personal career guide, highlighting skill gaps and providing actionable recommendations. For institutions and placement officers, it offers data-driven insights, enabling efficient monitoring, informed decision-making, and enhanced placement outcomes.

The modular architecture, combined with advanced AI and API integrations, ensures scalability, flexibility, and continuous improvement. Ultimately, this system empowers students to enhance their employability while assisting colleges in achieving higher placement success rates, thus creating a smarter, more efficient, and impactful placement ecosystem.

CHAPTER - 10

BIBLIOGRAPHY & REFERENCES

1. React.js https://react.dev/reference/react

2. FastAPI. https://fastapi.tiangolo.com/reference/

3. Gemini API. https://aistudio.google.com/

4. OpenAI. (2023). OpenAI API Documentation. Retrieved from https://platform.openai.com/docs

5. Google Developers. (2023). YouTube Data API Documentation. Retrieved from https://developers.google.com/youtube/v3

6. Tailwind Labs. (2023). Tailwind CSS Documentation. Retrieved from https://tailwindcss.com/docs