BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade

Sathyamangalam - 638401 Erode District, Tamil Nadu, India

Student Name: ATHISUNDARARAJ S

Seat No: 122

Project ID: 2

Project title: Training Performance Tracker

Components - Tech Stack Used:

Components	Tech Stack
Backend	Html, CSS, JavaScript
Frontend	Python, Django
Database	MySQL

Implementation Timeline:

Project Phases	Deadline	Status
Planning and Requirement Gathering	03/05/2024	Under review
Design and Prototyping		Not started
DB Designing and implementation		Not started
Backend Development		In progress
Integration and Testing		Not started
Deployment		Not started

Project Phases in Detail:

1. Planning and Requirement Gathering:

Overview: In this phase, we have to identify stakeholders, gather requirements, and define the project scope, objectives, and constraints. The primary goal is to understand the project's purpose, user needs, and business requirements.

Key Activities:

- Identify stakeholders and their roles.
- Gather requirements through interviews, surveys, and workshops.
- Define project objectives, scope, and constraints.
- Create a project plan, including timelines, resources, and deliverables.

Deliverables:

- Stakeholder analysis.
- Requirement documentation.
- Project plan.

2. Design and Prototyping:

Overview: This phase focuses on designing the system architecture, user interface, and interactions based on the gathered requirements. Prototyping helps visualize the system and gather feedback early in the development process.

Key Activities:

- Design system architecture and database schema.
- Create wireframes and mockups for user interfaces.
- Develop prototypes to validate design concepts and gather feedback.
- Refine designs based on stakeholder feedback.

Deliverables:

- System architecture diagrams.
- Database schema design.
- User interface wireframes and mockups.
- Prototypes with user feedback.

3. DB Designing and Implementation:

Overview: In this phase, the database structure is designed based on the requirements and system architecture. The database is implemented according to the design, ensuring data integrity, security, and performance.

Key Activities:

- Design the database schema based on the project requirements.
- Choose appropriate database technologies and tools.
- Implement the database structure, including tables, relationships, and constraints.
- Populate the database with initial data if necessary.

Deliverables:

- Database schema design documentation.
- Database implementation.

4. Backend Development:

Overview: The backend development phase involves building the server-side logic, APIs, and business logic required to support the system's functionality. This phase ensures that the system can handle data processing, user authentication, and other backend operations.

Key Activities:

- Develop server-side components using chosen programming languages and frameworks.
- Implement APIs for data retrieval, manipulation, and authentication.
- Integrate with external services if required.
- Implement business logic to support system functionality.

Deliverables:

- Backend codebase.
- API documentation.
- Authentication mechanisms.

5. Integration and Testing:

Overview: Integration and testing involve combining individual components of the system, verifying their interactions, and ensuring that the system functions as expected. This phase includes various testing activities to identify and fix defects.

Key Activities:

- Integrate frontend and backend components.
- Perform unit testing, integration testing, and system testing.
- Conduct functional, performance, and security testing.
- Identify and fix defects through iterative testing cycles.

Deliverables:

- Integrated system.
- Test reports and documentation.
- Bug fixes and enhancements.

6. Deployment:

Overview: Deployment involves deploying the system to production environments, making it accessible to users. This phase ensures a smooth transition from development to live operation.

Key Activities:

- Prepare the production environment, including servers, databases, and other infrastructure.
- Deploy the application code, database schema, and any necessary configurations.
- Perform smoke testing to verify that the system works in the production environment.
- Monitor system performance and address any issues that arise during deployment.

Deliverables:

- Deployed application in the production environment.
- Monitoring and logging configurations.
- Deployment documentation.

Problem Statement:

In educational institutions, managing student training data efficiently while providing insights into talent scenarios is essential for facilitating personalized learning and academic growth. However, the absence of a centralized system for storing and analyzing student data poses several challenges:

1. **Fragmented Data Management:** Currently, student training data, including attendance records and assessments, are often scattered across various platforms or stored in disparate formats, leading to inefficiencies in data access and utilization.
2. **Lack of Insight into Talent Development:** Without a unified system for talent analysis, educators and administrators struggle to gain insights into students' skill development and progress, hindering their ability to provide targeted support and guidance.
3. **Inefficient Analysis Processes:** Manual analysis of student data is time-consuming and prone to errors, making it challenging for educators to identify trends, patterns, and areas for improvement effectively.
4. **Limited Student Engagement:** Students lack a comprehensive view of their own progress and talent development, resulting in reduced engagement and motivation towards their learning goals.

Project Flow:

1. Purpose:

The purpose of this project is to develop a comprehensive Student Data Management and Analysis System to streamline data management, provide personalized insights, and improve decision-making for educational institutions.

2. Scope:

User Authentication:

- Implement a secure authentication system for students and administrators using their work mail.

Student Dashboard:

- Allow students to upload attendance, assessment, and other relevant data.
- Provide personalized analysis of student skills and performance.

Admin Dashboard:

- Enable administrators to access and manage student data.
- Provide tools for overall performance analysis and decision-making.

Skill Analysis:

- Develop a machine learning model to analyze student data and identify strengths and weaknesses in various skills.

Skill Ranking:

- Implement another machine learning model to generate rankings for each skill based on the analysis.
- Allow administrators to view top-performing students in specific skills.
- Data Storage and Retrieval:
- Integrate with a SQL database for storing and retrieving student data.

3. Business Context:

In educational institutions, effective management of student data and analysis of their performance are essential for improving teaching quality, student outcomes, and institutional effectiveness. By developing a Student Data Management and Analysis System, we aim to provide a centralized platform for administrators and students to access, analyze, and act upon student data efficiently. This system will enhance decision-making processes, improve student engagement, and contribute to overall academic success.

4. Considerations:

- **Security:** Ensure the system's security by implementing robust authentication measures and data encryption techniques to protect sensitive student information.
- **Scalability:** Design the system to accommodate the growing number of users and data volume over time without compromising performance.
- **User Experience:** Prioritize user experience by creating intuitive interfaces for both students and administrators, making data upload, analysis, and management tasks straightforward and accessible.
- **Regulatory Compliance:** Ensure compliance with relevant data protection regulations, such as GDPR or FERPA, to safeguard student privacy and confidentiality.

5. Dependencies:

- **Technology Stack:** The project depends on technologies such as HTML, CSS, JavaScript for frontend development, Python and Django for backend development, SQL for database management, and machine learning libraries for skill analysis and ranking.
- **Data Availability:** The availability and quality of student data are crucial dependencies for training machine learning models and generating meaningful insights.
- **Stakeholder Involvement:** Collaboration and feedback from stakeholders, including students, teachers, and administrators, are essential for defining requirements and validating the system's functionality.

User Personas

Student: Needs an up-to-date Dashboard to effectively upload attendance, assessment, and other relevant data.

Admin Staff: Manages system operations and Database, resolves conflicts, and approves student details, check analysis and view rankings.

User Stories:

- As a student, I want to be able to log in securely using my work mail, so that my personal information remains protected.
- As a student, I want to upload my attendance and assessment data easily, so that I can keep track of my academic progress.
- As a student, I want to view personalized analysis of my skills and performance, so that I can identify areas for improvement.
- As an administrator, I want to have full access to student data, so that I can monitor overall performance and make data-driven decisions.
- As an administrator, I want to be able to restrict or delete data uploaded by students if necessary, to maintain data integrity.
- As an administrator, I want to analyze student performance by skill, so that I can identify top-performing students in specific areas.

Functional Requirements:

- **User Authentication:**

  The system shall provide secure user authentication using work mail for both students and administrators.

- **Data Upload:**

  Students shall be able to upload attendance, assessment, and other relevant data.

- **Administrators:** should have the ability to manage and review uploaded data.

- **Personalized Analysis:**
  1. The system shall generate personalized analysis of student skills and performance based on uploaded data.
  2. This analysis shall be accessible to students and administrators.

- **Data Management:**

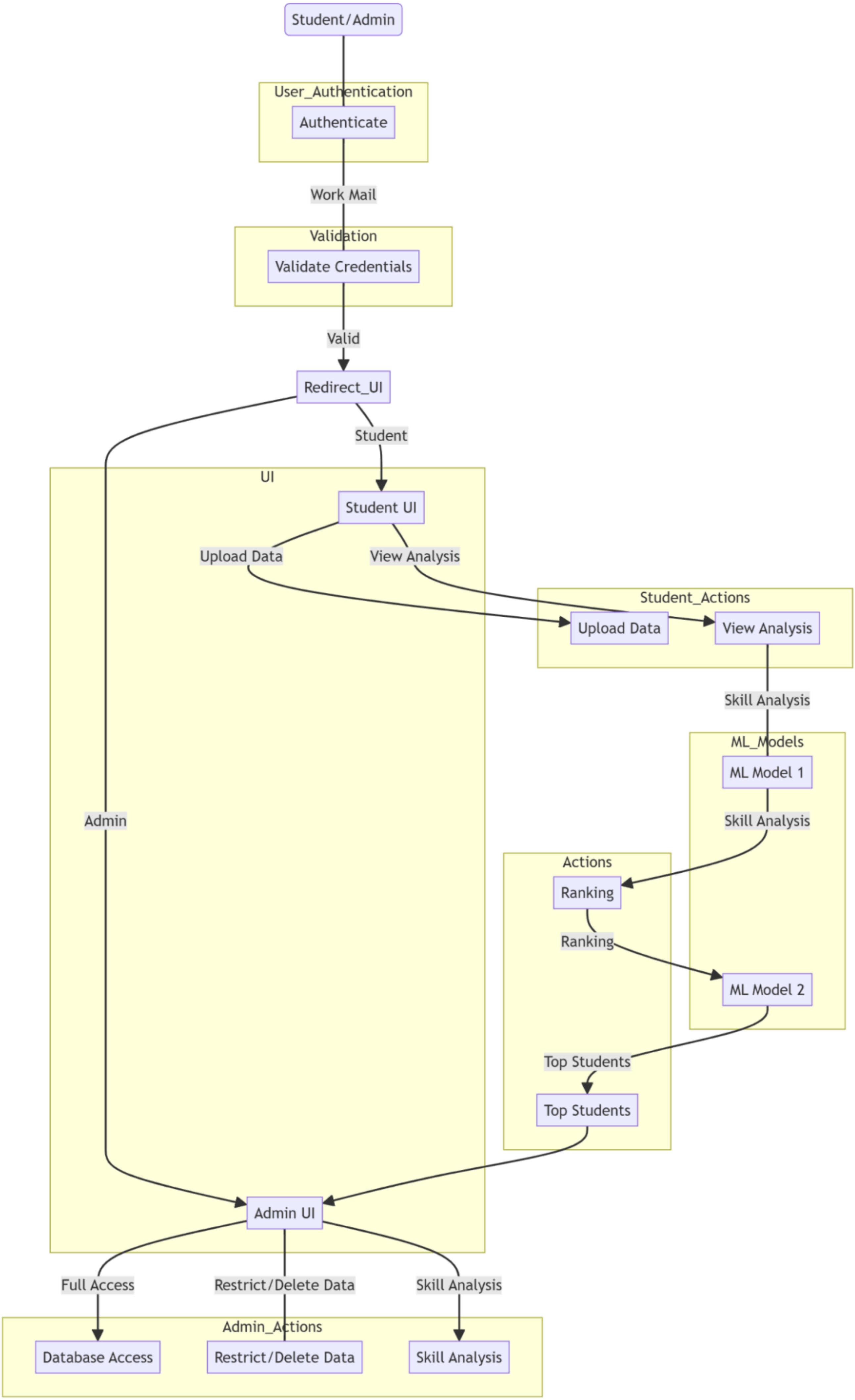
  Administrators shall have full access to student data and be able to perform actions such as restricting users or deleting data.

- **Skill Analysis:**

  The system shall use machine learning to analyze student data and identify strengths and weaknesses in various skills.

- **Skill Ranking:**
  1. Another machine learning model shall generate rankings for each skill based on the analysis.
  2. Administrators shall have access to view the top-performing students in specific skills

FLOW CHART:


Student/Admin
User_Authentication
Authenticate
Work Mail
Validation
Validate Credentials
Valid
Redirect_UI
Student
UI
Student UI
Upload Data
View Analysis
Student_Actions
Upload Data
View Analysis
Skill Analysis
ML_Models
ML Model 1
Skill Analysis
Admin
Actions
Ranking
Ranking
ML Model 2
Top Students
Top Students
Admin UI
Full Access
Restrict/Delete Data
Skill Analysis
Admin_Actions
Database Access
Restrict/Delete Data
Skill Analysis