**ASSESSMENT PLATFORM**

| **Name** | SARANRAJ R |
| --- | --- |
| **Roll no** | 7376222FD147 |
| **Seat no** | 185 |
| **Project id** | 01 |
| **Problem Statement** | Assessment platform that should contain mcq, short answer and match type of question and compilers. |

**TECHNICAL COMPONENT:**

| **Component** | **Tech Stack** |
| --- | --- |
| Frontend | React (JavaScript Library) |
| Backend | Spring Boot (Java Framework) |
| Database | MySQL Database |
| API | RESTFul API |

**IMPLEMENTATION TIMELINE:**

| **Phase** | **Deadline** | **Status** | **Notes** |
| --- | --- | --- | --- |
| Stage 1 | 06/06/2024 | Approved | Planning and  requirement |
| Stage 2 |  | In Progress | Design UI/UX and Prototyping |
| Stage 3 |  | Not Started | DB Designing |
| Stage 4 |  | Not Started | Backend  Implementation |
| Stage 5 |  | Not Started | Testing &  Implementation |
| Stage 6 |  | Not Started | Deployment |

**PURPOSE:**

The Programming Skill (PS) Platform is designed to provide students with a comprehensive interface for learning and assessment in programming and other subjects. It offers a structured approach to course selection, registration, ongoing course management, and assessment with various question types including MCQs, programming questions, and match questions.

**SCOPE:**

The PS Platform allows students to:

* Select and view available courses.
* Register for eligible courses.
* Manage ongoing courses with features such as booking slots for video sessions.
* Undergo assessments with different question formats and realtime question fetching based on course type.

**BUSINESS CONTEXT:**

The PS Platform aims to enhance students learning experience by providing a streamlined, userfriendly interface for course management and assessment. Stakeholders include students and educators within the academic institution.

**CONSIDERATIONS:**

**User Experience:**

Prioritize an intuitive and userfriendly interface for easy navigation and use.

**Security Measures:**

Implement a secure login system allowing access only for users with valid college email IDs.

**Time Management:**

Provide features for students to book slots for course sessions at predefined times.

**Programming Support:**

Offer support for programming assessments with an online compiler for languages such as C, C++, Python, and Java.

**DEPENDENCIES:**

* Integration with the college's email server for user authentication and access control.
* Reliable hosting infrastructure to support platform performance and availability.
* Compatibility with modern web browsers.

**USERS:**

**Student:**

Requires access to course selection, registration, ongoing course management, and diverse question types during assessments.

**Educator:**

Needs tools to manage courses and create assessments with various question formats.

**Administrator:**

Manages system settings, user accounts, and ensures smooth platform operations.

**USER STORIES:**

**As a student,**

* To select and register for courses.
* To manage ongoing courses and book slots for video sessions.
* To undergo assessments with diverse question types within the specified time frame.

**As an educator,**

To manage courses and create assessments with various question formats.

**As an administrator,**

To monitor platform usage, manage user accounts, and ensure the security and accuracy of the platform.

**FUNCTIONAL REQUIREMENTS:**

**Secure Login:**

Authentication system allowing access only for users with valid college email IDs.

**Course Management:**

Features to select available courses, register for eligible courses, and manage ongoing courses.

**Slot Booking:**

Allow students to book slots for course sessions at predefined times.

**Question Types:**

Support for MCQs, programming questions, and match questions.

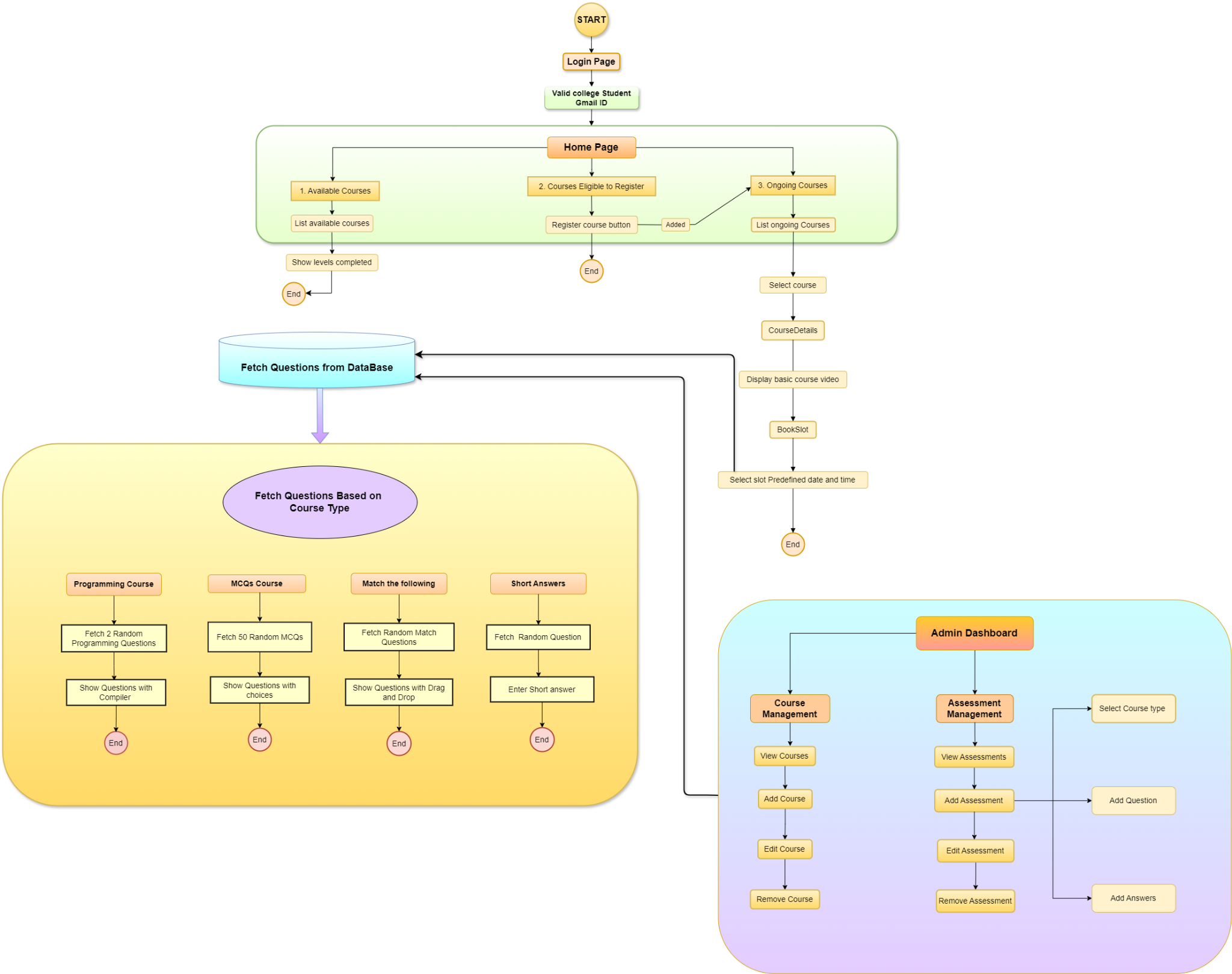
**Online Compiler:**

Integration of an online compiler for C, C++, Python, and Java languages for programming assessments.

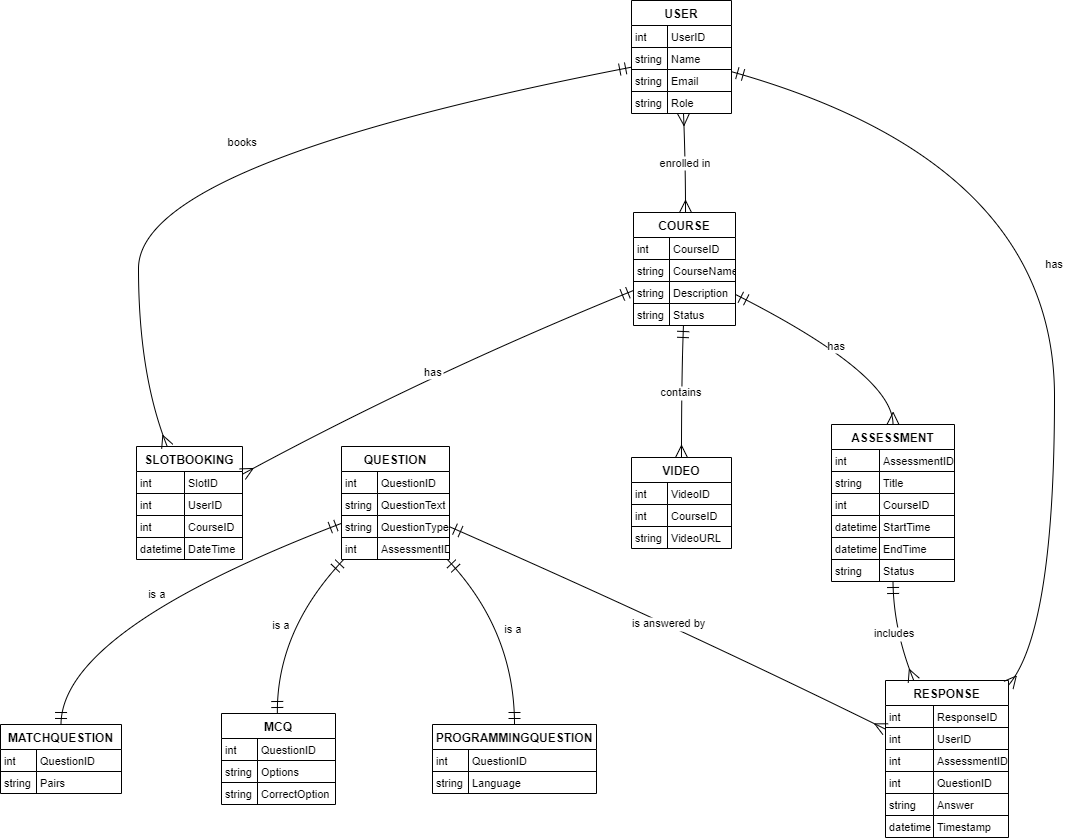
**Assessment Creation:**

Tools for educators to create and manage assessments efficiently.

**5. FLOWCHART :**

****

**Entity Relationship Diagram:**

****