



SRS - Online Assessment System

Development Road map

Aa Requirements	Priority	Expected Due date
<u>Login / Registration</u>	P0	
<u>Create Class</u>	P0	
<u>Create problem</u>	P0	
<u>setup compiler API</u>	P0	
<u>Create Lab / Contest</u>	P0	
<u>Comment on problem / blog / class</u>	p1	
<u>Email API setup [Preferably SMTP Gmail]</u>	p1	

<u>Aa Requirements</u>	Priority	Expected Due date
<u>Code Editor , mark down editor for comments</u>	p1	
<u>Submit code / assignment</u>	p1	
<u>Lab Grading</u>	p1	
<u>Plagiarism Check</u>	p2	
<u>Peer code review</u>	p2	
<u>Live class</u>	p2	

Description :

The goal is to develop a web application that provides the platform for faculties & students to manage their C programming assignments and many more including coding contests, doubt solving, peer code reviews. Faculties can even share their approach to a problem in a form of a blog or pre-recorded video lectures.

Total **3** type of users:

1. **Admin**
2. **Faculty**
3. **Students**

Functional Requirements

Common Requirements

- **Log-in to the system**

Input: User credentials

Output: if the credentials are correct then the user will be redirected to the dashboard

Exceptional Flow: If the entered credentials are incorrect then the user will see the error message and will be asked to enter the credentials again.

- **Edit Profile**

Input: User selection

Output: Prefilled user details will be displayed.

- **Update profile**

Input: updated user details

Output: Success message displaying profile has been updated.

- **Forgot Password**

Input: User email

Output: Success or failure message

Process: If the user is registered to the system then an email with password reset link will be sent to the registered email id or appropriate message will be displayed.

Admin

- **Create Faculty Account**

Input: Faculty details

Output: Success message displaying user has been created.

- **Create Student Account**

Input: Student details

Output: Success message displaying user has been created.

- **View Stats**

Input: User selection

Output: Statistical data of the system will be displayed.
i.e total no of students / faculties etc.

Faculty

- **Create Class**

Description: faculty can create classroom.

Input: Details of classroom.

Output: Unique classroom code will be generated by the system.

Process: unique code will be generated by the system.

- **Create Lab**

Description: Lab can be created in the selected classroom by faculty.

Input: Lab details. ex. Title, description, deadline etc.

Output: Lab will be created and the user will be redirected to the newly created lab.

- **Create Contest**

Description: Contest can be created in the selected classroom by faculty.

Input: Contest details. ex. Title, description, Time duration etc.

Output: Contest-created and redirected to the Lab page with a success message.

- **View Lab**

Description: Faculty can view the Lab details and it's Questions (Programs).

Input: User selection

Output: Lab details

- **View Contest**

Description: Faculty can view the Contest details and it's Programs.

Input: User selection

Output: Contest details

- **Edit Lab**

Input: User selection

Output: Prefilled Lab details will be displayed.

- **Update lab details**

Input: updated lab details

Output: Success message displaying lab details has been updated & Redirected.

- **Edit contest**

Input: User selection

Output: Prefilled contest details will be displayed.

- **Update contest details**

Input: updated contest details

Output: Success message displaying contest details has been updated & Redirected.

- **Delete Lab**

Input: User selection

Output: Success message displaying lab has been deleted & Redirected.

- **Delete Contest**

Input: User selection

Output: Success message displaying Contest has been deleted & Redirected

- **View assignment submissions**

Input: User selection

Output: List of assignments by individual students will be displayed.

- **View code and Suggest improvement [comments]**

Description: The faculty can suggest a few changes to any students' submission through comments. Comments are private to the particular user and faculty.

Input: A brief about possible Improvements in the code

Output: On pressing submit button comment will be displayed on the assignment page.

- **Create problems per lab / contest**

Input: Problem details ex. Problem definition, Test cases, running time, difficulty level, etc.

Output: A problem will be created and the user will be redirected to the respective lab/contest.

- **View problems per lab / contest**

Input: User selection

Output: Faculty will see the Problem definition and details.

- **Edit problems per lab / contest**

Input: User selection

Output: Prefilled problem details will be displayed.

- **Update problem details**

Input: updated problem details

Output: Success message displaying Problem data has been updated & Redirected.

- **Delete problems per lab / contest**

Input: User selection

Output: Success message displaying Problem has been deleted & Redirected.

- **Comment on problem**

Description: The student can comment on any problem regarding his doubt.

Input: User comment

Output: On pressing the submit button comment will be displayed on the problem page.

- **Lab grading [1 -10]**

Description: Faculty can grade the assignments submitted by the students on a scale of 1 to 10.

Input: Lab grade

Output: Success message

- **Create blog to provide solution**

Input: Blog details

Output: Success message displaying blog created and the user will be redirected to the newly created blog.

- **View Statistics**

Description: number of the contest, number of contest attended by the user, etc. will be displayed in form of Charts

Input: User selection

Output: Statistics of data in form of charts

- **Create Live Class**

Input: User selection

Output: Virtual class will be created and the link will be shared with the class

- **Post a Comment in class**

Description: Faculty and Students both can share some information with the class. Like sharing resources or discussing few common doubts.

Input: User comment, attachment (if any)

Output: List of user comments in a particular class

- **Post a Comment on Blog**

Description: Faculty and Students both can comment on blog

Input: User comment, attachment (if any)

Output: List of user comments in a particular class

Students

- **Join class**

Description: Student can join the class by unique class code

Input: Class code

Output: Redirected to classroom

- **submit assignments**

Description: Student can submit the assignment of problem in the Lab/ Contest

Input: User selection

Output: Success message

- **Attend contest**

Input: User selection

Output: Redirected to the contest page

- **online editor / compiler**

Input: Code to be compiled with the language selection

Output: Code output

- **View Statistics**

Description: Students can view their statistical data like score & grade of every contest/lab in the form of charts and many more.

Input: User selection

Output: Statistical data in the form of charts.

- **peer code review**

Description: After completion of lab or any contest, the student can view other classmates' code

Input: User selection

Output: Selected code

- **Join Live Class**

Input: User selection

Output: Redirect to live class

References :

Online free APIs to compile C code

<https://github.com/Cloud-Compiler-API/Cloud-Compiler-API>

<https://www.npmjs.com/package/compile-run>

<https://codechef.ideone.com/api>

https://cran.r-project.org/web/packages/lintr/vignettes/creating_linters.html