Problem : TO create HLD DOC for online judge platform :

**Database :**

1)Problem : {

Pid : int,

title : String,

Description : String,

SubmissionCount : int ,

Difficulty Level : String,

Marks : int

Tags : array of string (for filtering )

}

2) Submission : {

subId : Int,

pid : Int (Foreign key ),

user\_id : Int (Foreign key)

code : String ,

verdict : String ,

timeDate : Date

}

3) User :{

User\_id : Int,

MailId : String,

Phone : Int,

Password : String,

Total score : int

Pic : image

}

4) Test case:{

Pid: int(Foreign key)

Inputs : array of String,

Outputs : array of String

}

**2)Front End Screens**

Screen 1 :

Problem list and user profile( name and dp) (if not logged in will show login sing up box in place of user profile)

Screen 2 :

Individual Problem descriptive

Coding window

Input bar (for custom test case) and run and submit button

Screen 3 : user profile dashboard

User info

Dashboard : acceptance % , total submission, rank

List of problems recently attempted

* **Problem List :**

Front end : create a simple list of problems in React that displays the title and difficulty level of that problem , and links that to individual problem page

Backend : design an Api that fetches the problem list from the database and returns the same to front end for displaying

* **User profile :**

Frontend : shows the users username and image if logged in , or else prompts to signup or login

Backend : design an api that gets the username and his image as he logs in to the platform.

* **Individual problem :**

Frondend : display the title, description ,submission count, difficulty level, tags

Backend : make a get call to db to fetch the details of the specific problem

* **Coding window :**

Frontend : input box for the problem with [run+(input test cases) ] and submit buttons .

Backend :

run button - > run the code with the user input and show the output in verdict defined space .

submit - > make a get request to access the test cases from db for that problem ,execute the user code on the test cases . compare the obtained result with the test case output if every test case was successful then make post call to save code with verdict as passed and his user total score by making post call . If not success then with post call save that code with failed verdict .

* **User dashboard :**

Frontend : show the user info , his total score , his rank , calculate his acceptance % and his top 10 recently attempted problems .

Backend : with get call to user db , display user info and get call to submission db to show is recently attempted 10 questions (failed ones ).

**Execution :**

Will be using custom isolation technique for safety and security , will be running our code from container hosted on AWS.