**High-Level Design**

**Database Collections-**

1. **Users**
   1. \_id
   2. Handle (unique)
   3. Password
   4. Email (unique)
   5. Dob
   6. fullName
   7. type
   8. Problems solved
   9. Problems attempted
2. **Tasks**
   1. \_id
   2. Name
   3. Statement
   4. Constraints
   5. Format
   6. Testcases
   7. tag
3. **Submissions**
   1. \_id
   2. Task\_id (foreign key)
   3. User\_id (foreign key)
   4. Code
   5. language
   6. Verdict
   7. Exec. Time
   8. Memory
   9. Submission time

**Frontend:-**

1. **Webpages**
   1. Sign Up page
      1. It would be a form with fields as in user collection and would have a submit button
   2. Login Page
      1. It would be a form with two fields email and password. It would have submit and forgot password buttons.
   3. Homepage
      1. Navbar at top
      2. List of Problem cards
      3. Footer at bottom
   4. Task Page
      1. Navbar at top
      2. Problem Title, Problem statement, Constraints, Sample inputs and outputs on the second div
      3. Footer at bottom
   5. Submission/Editor Page
      1. Navbar at top
      2. Div containing options for selecting task id and language
      3. Option for uploading files/ typing in the inbuilt code editor
      4. Footer at bottom
   6. Verdict Page
      1. Navbar at top
      2. Source code (read-only)
      3. Div containing information of the submission as enlisted in the submissions collection.
      4. Footer
   7. Profile Page
      1. Navbar at top
      2. Div containing Name, email, photo, dob, handle name( read-only), problems solved(read-only) and problems attempted(read-only).
      3. Footer
2. **Components**
   1. Navbar
      1. Logo at left
      2. Handle at right with link to profile page
      3. Logout button at rightmost
   2. Footer
      1. Text
      2. “Mail to” link
   3. Problem Card
      1. Title of the Problem
      2. Tags
      3. Success Rate
      4. Clicking anywhere on the card would redirect to the problem page of that problem

**Backend:-**

1. **Routes**
   1. User routes
      1. /register (POST)
      2. /login (POST)
      3. /profile (GET)
   2. Problem routes
      1. /homepage (GET)
      2. /task (GET)
      3. /verdict (GET)
   3. Secured routes
      1. /submit (POST)
      2. /setting (PUT)
   4. Admin routes
      1. /task (GET, POST, PUT, DELETE)
2. JWT will be used for authentication of users for secured routes and admin routes

**DOCKER:-**

1. Need to make docker images and containers for the OJ.
2. Code Sandboxing to be implemented.