
Implementation Document

for

Sahara

Version 1.0

Prepared by

Group :8

Pothuganti Nikhil
Divi Pothukuchi
Harshpreet Kaur
Achyuth Warrier
Ravi Arora
Aarav Oswal
R.Charan
Spandan Pati
Vedhanth
Balasubramanian
Ashik Stenny

Group Name: Ravi and Friends

230760	nikhilp23@iitk.ac.in
230378	divip23@iitk.ac.in
230464	harshpreet23@iitk.ac.in
230052	achyuthw23@iitk.ac.in
230846	raviorora23@iitk.ac.in
230012	aaravoswal23@iitk.ac.in
230819	rcharan23@iitk.ac.in
231031	spandanp23@iitk.ac.in
231135	vedhanthb23@iitk.ac.in
230224	ashiks23@iitk.ac.in

Course: CS253

Mentor TA: *Prof. Dr. Indranil Saha*

Date: 28/03/2025

CONTENTS.....	II
REVISIONS.....	II
1 IMPLEMENTATION DETAILS.....	4
2 CODEBASE.....	7
3 COMPLETENESS.....	34
APPENDIX A - GROUP LOG.....	58

Revisions			
Version	Primary Author(s)	Description of Version	Date Completed
1.0	Group 8	First Draft of Implementation Document	28/03/25

1 Implementation Details

Programming Languages and Frameworks

1) Frontend

We have used the following programming languages and frameworks for the frontend development of our application:

- **React**
 - React is a JavaScript library for building user interfaces. It allows us to create reusable UI components and manage the state of the application efficiently.
- **Next.js**
 - Next.js is a React framework that provides server-side rendering (SSR) and static site generation (SSG). It improves performance and SEO for web applications.
- **Material-UI (MUI)**
 - Material-UI is a popular React component library that implements Google's Material Design. It provides pre-built, customizable UI components.
- **Axios**
 - Axios is a promise-based HTTP client for making API requests. It simplifies the process of sending requests and handling responses.
- **XLSX**
 - XLSX is a library for reading and writing Excel files. It allows us to generate and download .xlsx files programmatically.
- **TypeScript**
 - TypeScript is a superset of JavaScript that adds static typing. It helps catch errors during development and improves code maintainability.
- **Icons (Material-UI Icons)**
 - Material-UI Icons provide a collection of ready-to-use icons that follow Material Design guidelines.
- **CSS-in-JS (Material-UI Styling)**
 - Material-UI provides a built-in styling solution using CSS-in-JS. It allows us to define styles directly in JavaScript.
- **React Hooks**
 - React Hooks like useState, useEffect, and useRef simplify state management and lifecycle methods in functional components.
- **React Router (Next.js Navigation)**
 - Next.js provides a built-in router for navigating between pages. It simplifies client-side routing.
- **LocalStorage**
 - LocalStorage is used to store data persistently in the browser.

2) Backend

For the backend of our web application, we have chosen FastAPI due to its numerous advantages.

- **FastAPI**
 - FastAPI is a modern, high-performance web framework for building APIs with Python.
 - It is designed for speed and efficiency, leveraging asynchronous programming for optimal request handling.
- **Starlette**
 - Starlette is a lightweight ASGI framework that serves as the foundation for FastAPI, enabling high-performance asynchronous web applications.
- **Pydantic**
 - Pydantic is used in FastAPI for data validation and serialization. It ensures type safety and helps catch errors early in the development process.
- **OpenAPI & Swagger**
 - FastAPI automatically generates interactive API documentation using OpenAPI and Swagger.
 - This allows developers to test endpoints and understand request-response formats easily.
- **Asynchronous Support**
 - FastAPI fully supports asynchronous programming, making it ideal for non-blocking I/O operations, real-time applications, and microservices.
- **Type Hints**
 - FastAPI utilizes modern Python type hints, improving code readability, maintainability, and editor support with features like auto-completion and static analysis.
- **Integration**
 - FastAPI integrates seamlessly with tools like SQLAlchemy, Redis, and WebSockets, making it a great choice for building scalable applications.
- **Community & Adoption**
 - FastAPI is open-source and has an active developer community. It is used by companies like Microsoft, Uber, and Netflix for high-performance backend services.
- **Cloud & Containerization**
 - FastAPI is well-suited for modern cloud environments and containerized deployments, enhancing scalability and production readiness.

3) Database

For the database of our application, we have chosen PostgreSQL due to its numerous advantages and advanced capabilities.

- **PostgreSQL**
 - PostgreSQL is a powerful, open-source relational database management system (RDBMS) known for its robustness, flexibility, and advanced feature set.
- **ACID Compliance**
 - PostgreSQL strictly follows ACID (Atomicity, Consistency, Isolation, Durability) principles, ensuring reliable transactions and data integrity even in concurrent environments.
- **MVCC (Multi-Version Concurrency Control)**
 - MVCC allows multiple transactions to occur simultaneously without locking issues, ensuring smooth and efficient query execution.
- **Indexing Techniques**
 - PostgreSQL supports multiple indexing methods, including B-tree, Hash, GiST, GIN, and BRIN indexes, optimizing query performance for various data access patterns.
- **Advanced Query Optimization**
 - PostgreSQL features a powerful query planner and optimizer, enabling efficient execution of complex queries, large datasets, and multi-table joins.
- **Scalability**
 - PostgreSQL supports both vertical and horizontal scaling, with features like streaming replication, logical replication, partitioning, and sharding for handling large data volumes.
- **Data Types & Storage**
 - PostgreSQL supports structured and semi-structured data, including JSON, XML, and arrays, making it adaptable for diverse data storage needs.
- **Community & Development**
 - PostgreSQL has an active global community that provides regular updates, security patches, and extensive documentation, ensuring long-term support and continuous improvements.
- **Enterprise Adoption**
 - PostgreSQL is used by companies like Apple, Microsoft, Instagram, and Spotify due to its reliability, scalability, and advanced capabilities for handling complex workloads.

2 Codebase

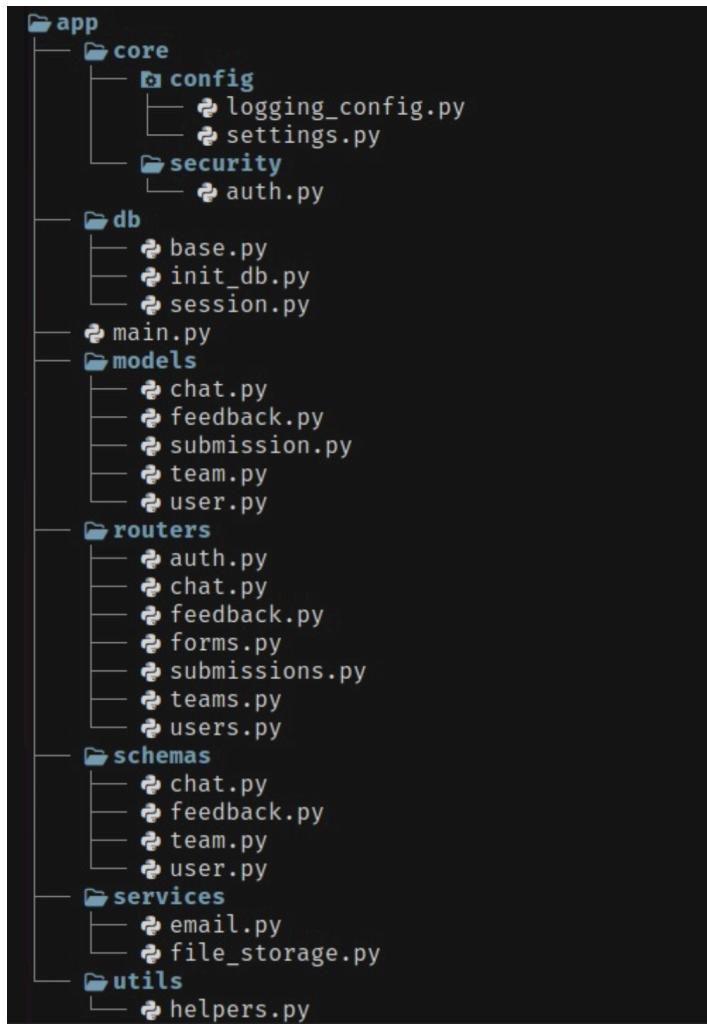
GitHub Repository: <https://github.com/EnvNaqSevraqf/sahara>

Code Structure:

The project is divided into two parts- the codebase and the databases based on the Model-View-Controller architecture format.

Codebase:

BackEnd



FrontEnd

```
</>src
  - app
    - dashboard
      - announcements
        - layout.tsx
        - page.tsx
      - assignments
        - creation
          - page.tsx
          - layout.tsx
          - page.tsx
          - professor.tsx
          - student.tsx
        - update
          - page.tsx
        - view
          - page.tsx
      - calendar
        - page.tsx
      - discussion
        - layout.tsx
        - page.tsx
      - feedback
        - admin.tsx
        - page.tsx
        - student.tsx
      - feedbackform
        - page.tsx
      - forms
        - [id]
          - formviewer.tsx
          - page.tsx
        - create_form
          - page.tsx
          - page.tsx
      - gradeables
        - [id]
          - page.tsx
        - create_gradable
          - page.tsx
          - page.tsx
        - layout.tsx
        - page.tsx
      - people
        - add
          - page.tsx
          - AddStudents.tsx
```

```
create_gradable
  └── page.tsx
  └── page.tsx
layout.tsx
page.tsx
people
  add
    └── page.tsx
  AddStudents.tsx
  AddTAs.tsx
  components
    ErrorBoundary.tsx
    Header.tsx
    LoadingState.tsx
    NavigationButton.tsx
  constants
    theme.ts
  hooks
    useDebounce.ts
  page.tsx
  PeopleTable.tsx
  skills
    page.tsx
student
  add
    page.tsx
    page.tsx
  StudentTable.tsx
tas
  add
    page.tsx
    page.tsx
  pairing
    page.tsx
  skills
    page.tsx
  TASKillsForm.tsx
  TATable.tsx
  TATEamPairing.tsx
  TeamCreationForm.tsx
  teams
    create
      page.tsx
    details
      page.tsx
      page.tsx
    TeamsCreationForm2.tsx
```

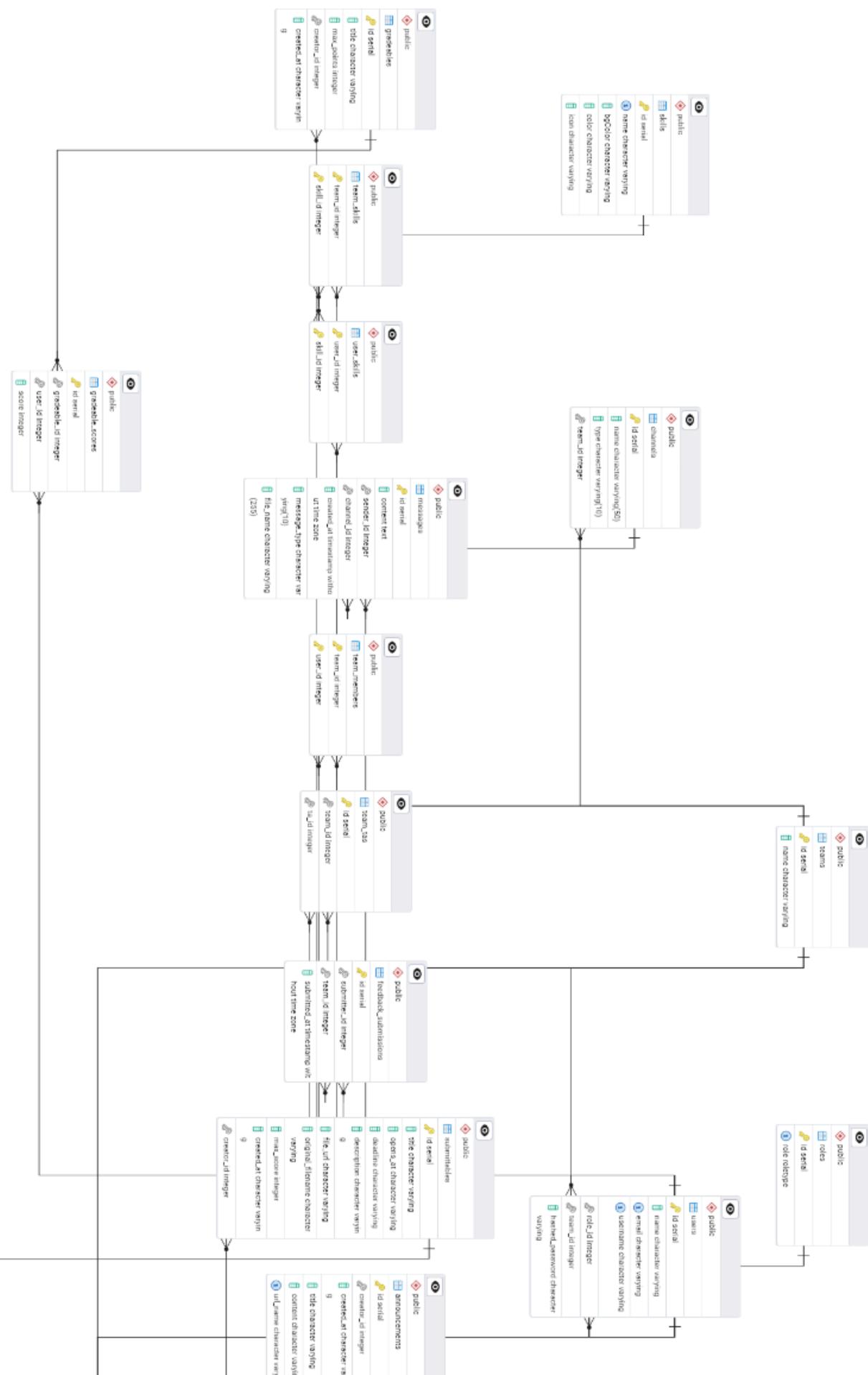
```
teams
  create
    page.tsx
  details
    page.tsx
    page.tsx
  TeamsCreationForm2.tsx
  TeamsDetails.tsx
  types
    index.ts
profile
  page.tsx
quiz
  [id]
    page.tsx
    quizViewer.tsx
  create_quiz
    page.tsx
    page.tsx
  submission
    creation
      page.tsx
      layout.tsx
      page.tsx
      professor.tsx
      student.tsx
    update
      page.tsx
    view
      page.tsx
  team
    page.tsx
    utils.ts
favicon.ico
globals.css
layout.tsx
login
  auth.jsx
  ForgotPassword.tsx
  layout.tsx
  login.tsx
  page.tsx
main
  calendar
    page.tsx
```

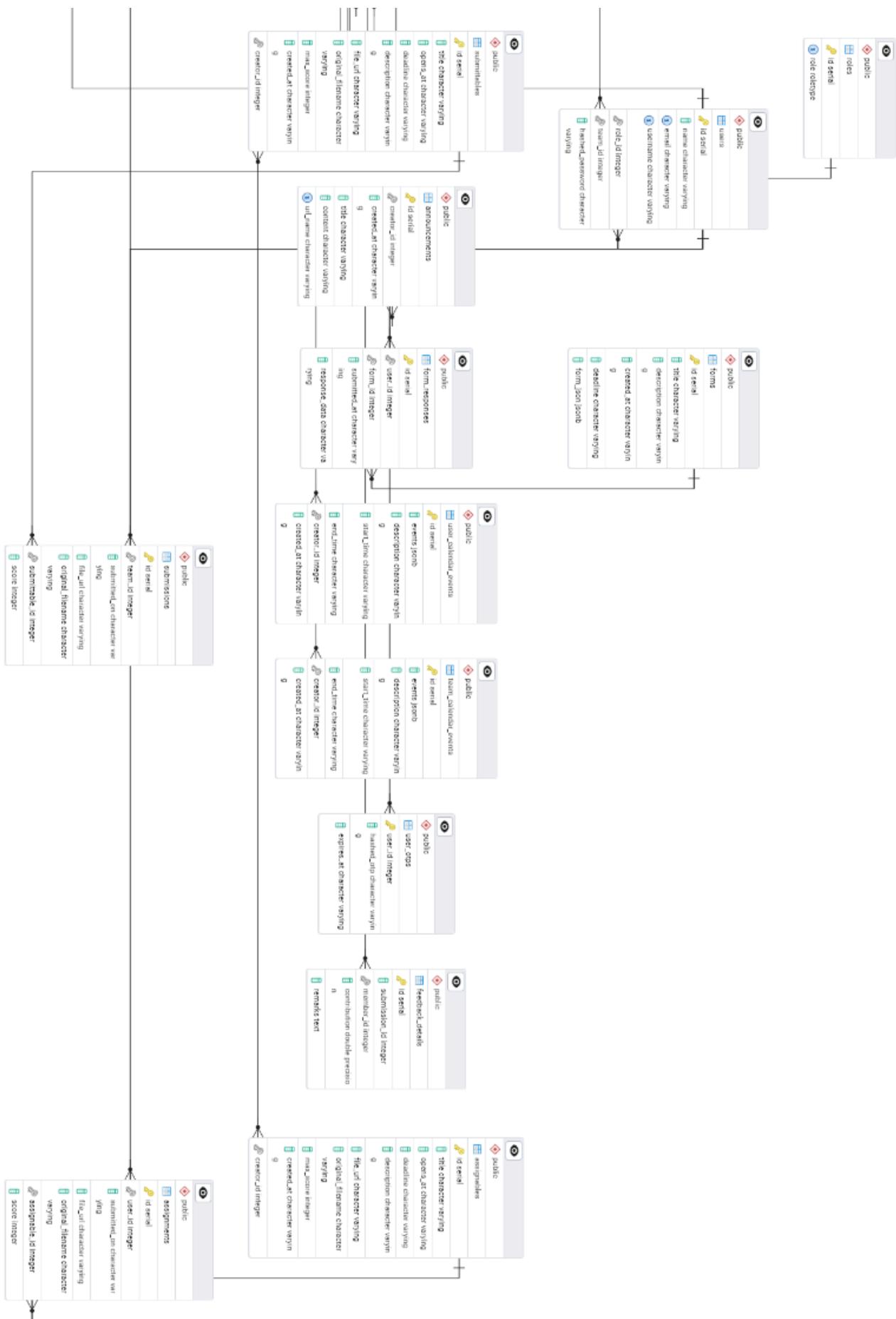
```
~ /Project/
  └── main
    ├── calendar
    │   ├── page.tsx
    │   ├── layout.tsx
    │   └── page.tsx
    └── page.tsx
  └── components
    ├── AuthWrapper.tsx
    ├── Header.tsx
    ├── logout.tsx
    ├── MenuButton.tsx
    ├── MenuContent.tsx
    ├── NavbarBreadcrumbs.tsx
    ├── Sidebar.tsx
    ├── TiptapEditor.tsx
    ├── Topbar.tsx
    └── upload.tsx
  └── config
    └── index.ts
  └── contexts
    └── AuthContext.tsx
  └── theme.tsx
  └── types
    ├── axios.d.ts
    ├── mui-icons.d.ts
    ├── tiptap.d.ts
    └── toolpad.d.ts
  └── utils
    └── roles.ts
```

Database

The database section contains various databases and collections of the form-

```
|---- announcements
|---- channels
|---- feedback_details
|---- feedback_submissions
|---- form_responses
|---- forms
|---- global_calendar_events
|---- gradeable_scores
|---- gradeables
|---- messages
|---- quiz_responses
|---- quizzes
|---- roles
|---- skills
|---- submission_files
|---- submissions
|---- submittable_reference_files
|---- submittables
|---- team_calendar_events
|---- team_members
|---- team_skills
|---- team_tas
|---- teams
|---- user_calendar_events
|---- user_skills
|---- user_otps
|---- users
```





API Endpoint Specifications

Authentication

1. Login Endpoint

- **URL:** /login
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** application/x-www-form-urlencoded
- **Required Parameters:**
 - `username` (string): User's login username
 - `password` (string): User's login password
 - `scope` (string, optional): Default: "*"
 - `client_id` (string, optional): Client identifier
 - `client_secret` (string, optional): Client secret

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successful login
 - **Response Body:** Not specified in the documentation
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input validation failed

2. Reset Password Endpoint

- **URL:** /reset-password
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** application/json
- **Required Parameters:**
 - `new_password` (string): New password to be set
 - `confirm_password` (string): Confirmation of new password

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Password reset successful
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input validation failed

3. Create Prof Endpoint

- **URL:** /create-prof
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** application/json
- **Required Parameters:**
 - **name** (string): Full name of the user
 - **email**(string):Email id of the user

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Professor profile created successfully
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input validation failed

File Upload

4. Upload Students

- **URL:** /upload-students
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** multipart/form-data
- **Required Parameters:**
 - **file** (string, binary): File containing student data

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Students file uploaded successfully
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input validation failed

5. Upload TAs

- **URL:** /upload-tas
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** multipart/form-data
- **Required Parameters:**
 - **file** (string, binary): File containing student data

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** TAs file uploaded successfully
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input validation failed

Announcements

6. Create Announcements

- **URL:** /announcements
- **Method:** POST
- **Request Body Schema:** multipart/form-data
- **Required Parameters:**
 - title (string)
 - description (string)
 - file (optional)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully created an announcement
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input validation failed

7. Get Announcements

- **URL:** /announcements
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:** null

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved all announcements
 - **Response Body:**
 - id: integer (Id) - required
 - creator_id: integer (Creator Id) - required
 - created_at: string (Created At) - required
 - title: string (Title) - required
 - content: object (Content)
 - url_name: Url Name (string or null)

8. Delete Announcements

- **URL:** /announcements/{id}
- **Method:** DELETE
- **Request Body Schema:** null
- **Required Parameters:** id (integer)

Responses:

- **204 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully deleted an announcement
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

9. Update Announcements

- **URL:** /announcements/{id}
- **Method:** PUT
- **Request Body Schema:** null
- **Required Parameters:**
 - id (integer) (Path Parameter)
 - title (string) (Query Parameter)
 - description (string) (Query Parameter)
 - file(file)(optional)

Responses:

- **202 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully updated an announcement
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

10. Show Announcement

- **URL:** /announcements/{id}
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:**
 - id (integer) (required)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved an announcement
 - **Response Body:**
 - id: integer (Id) - required

- `creator_id`: integer (Creator Id) - required
- `created_at`: string (Created At) - required
- `title`: string (Title) - required
- `content`: object (Content)
- `url_name`: Url Name (string or null)
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

Calendar

11. Updated Calendar

- **URL:** /calendar/update
- **Method:** POST
- **Request Body Schema:** null
- **Required Parameters:** events (array of events) (required)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully updated calendar
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

12. Get Calendar

- **URL:** /calendar/
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:** null

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved calendar
 - **Response Body:** null

People

13. Get (all) People

- **URL:** /people/
- **Method:** GET
- **Request Body Schema:** null

- **Required Parameters:** `null`

Responses:

- **200 Successful Response:**

- **Content Type:** `application/json`
- **Description:** Successfully retrieved all people
- **Response Body:** `null`

14. Get all Teams

- **URL:** `/team/`
- **Method:** GET
- **Request Body Schema:** `null`
- **Required Parameters:** `null`

Responses:

- **200 Successful Response:**

- **Content Type:** `application/json`
- **Description:** Successfully retrieved all teams
- **Response Body:** `null`

Quizzes & Forms

15. Create Quiz

- **URL:** `/quiz/create`
- **Method:** POST
- **Request Body Schema:** `null`
- **Required Parameters:** `null`

Responses:

- **200 Successful Response:**

- **Content Type:** `application/json`
- **Description:** Successfully created quiz
- **Response Body:** `null`

16. Get Forms

- **URL:** `/forms`
- **Method:** GET
- **Request Body Schema:** `null`
- **Required Parameters:** `null`

Responses:

- **200 Successful Response:**

- **Content Type:** `application/json`
- **Description:** Successfully retrieved all form
- **Response Body:** `null`

17. Create Form

- **URL:** /forms/create
- **Method:** POST
- **Request Body Schema:** null
- **Required Parameters:** null

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully created a form
 - **Response Body:** null

18. API Create Form

- **URL:** /api/forms/create
- **Method:** POST
- **Request Body Schema:** application/json
- **Required Parameters:**
 - title (string)
 - form_json(Object)
 - deadline(string)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully created form
 - **Response Body:** null

19. API Submit Response

- **URL:** /api/forms/submit
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:** null

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully submitted form
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

20. API Get Form

- **URL:** /api/forms/{form_id}
- **Method:** GET

- **Request Body Schema:** null
- **Required Parameters:**
 - form_id (int)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved form
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

21. API Check Deadline

- **URL:** /api/forms/{form_id}/check-deadline
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:**
 - form_id(int)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved deadline for form
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

22. API Get Forms

- **URL:** /api/get_forms
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:**
 - user_id(integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved form
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

23. API Get User Response

- **URL:** /api/forms/{form_id}/user/{user_id}
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:**
 - form_id(integer)
 - user_id(integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved user response
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

Skills Form (For TA)

24. Get All Skills

- **URL:** /api/skills
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:** null

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved all skills
 - **Response Body:** null

25. Create Skills

- **URL:** /api/skills/create
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - name (string)
 - bgColor (string)
 - color (string)
 - icon (string)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully created skill

- **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

26. Get User Skills

- **URL:** /api/users/{user_id}/skills
- **Method:** GET
- **Request Body Schema:** null
- **Required Parameters:**
 - user_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved User Skills
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed
 - **Response Body:** null

27. Assign Skills to Users

- **URL:** /api/users/assign-skills
- **Method:** PUT
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - user_id (integer)
 - skill_ids (array of integers)

Responses:

- **202 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully assigned skills to users
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

28. Delete Skills

- **URL:** /api/skills/{skill_id}
- **Method:** DELETE
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - skill_id (integer)

Responses:

- **202 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully deleted skill
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

29. Get Team Skills

- **URL:** /api/teams/{team_id}/skills
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - team_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved team skills
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

30. Assign Skills to Teams

- **URL:** /api/teams/assign-skills
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - team_id (integer)
 - skill_ids(array of integers)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully assigned skills to teams
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

31. Get Current TA-Team Assignments

- **URL:** /match
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer

- **Request Body Schema:** `null`
- **Required Parameters:** `None`

Responses:

- **200 Successful Response:**
 - **Content Type:** `application/json`
 - **Description:** Returns current team-TA assignments with team details and skills
 - **Response Body:** `null`
- **422 Validation Error:**
 - **Content Type:** `application/json`
 - **Description:** Input Validation failed

32. Create New TA-Team Assignments

- **URL:** `/match/{n}`
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** `null`
- **Required Parameters:** `n` (integer): Number of TAs to assign per team (path parameter)

Responses:

- **200 Successful Response:**
 - **Content Type:** `application/json`
 - **Description:** Creates new team-TA assignments based on skill match and other constraints
 - **Response Body:** `null`
- **400 Bad Request:**
 - **Content Type:** `application/json`
 - **Description:** Constraint on n, when `n<=0`
- **422 Validation Error:**
 - **Content Type:** `application/json`
 - **Description:** Input Validation failed

33. Update Team's TA Assignments

- **URL:** `/teams/{team_id}/update-tas`
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** JSON
- **Required Parameters:**
 - `team_id` (integer): ID of the Team to update TAs of (path parameter)
 - `ta_ids` (array): List of TA IDs to assign to the team (Request body)

Responses:

- **200 Successful Response:**
 - **Content Type:** `application/json`
 - **Description:** Updates TA assignments for a specific team

- **Response Body:** null
- **400 Bad Request:**
 - **Content Type:** application/json
 - **Description:** When selected users aren't TAs or do not exist
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

Gradeables

34. Get Gradeable Table

- **URL:** /gradeables/
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved gradable table
 - **Response Body:** null

35. Get Grorable by ID

- **URL:** /gradeables/{gradeable_id}
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - gradeable_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved grorable by id
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

36. Get Grorable Submissions

- **URL:** /gradeables/{gradeable_id}/scores
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - gradeable_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved gradable submissions
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

37. Create Gradeables

- **URL:** /gradeables/create
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** application/json
- **Required Parameters:**
 - **title** (string)
 - **due_date**(string)
 - **max_points**(integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully created a gradable
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

Project Submissions

38. Submit Files

- **URL:** /submittables/{submittable_id}/submit
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** multipart/form-data
- **Required Parameters:**
 - **submittable_id** (integer)
 - **files**

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully submitted files
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

39. Download Submission

- **URL:** /submissions/{submission_id}/download
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - submission_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully downloaded submission
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

40. Get All Submittables

- **URL:** /submittables/
- **Method:** PUT
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:** null

Responses:

- **202 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieves all submittables
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

41. Download Reference Files

- **URL:** /submittables/{submittable_id}/reference-files/{ref_file_id}/download
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - submittable_id (integer)
 - ref_file_id(integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully downloaded the reference file for a submittable
 - **Response Body:** null
- **422 Validation Error:**

- **Content Type:** application/json
- **Description:** Input Validation failed

42. Create Submittable

- **URL:** /submittables/create
- **Method:** POST
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** multipart/form-data
- **Required Parameters:**
 - submittable (object)
 - files

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully created a submittable
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

43. Get Submittable

- **URL:** /submittables/{submittable_id}
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - submittable_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved submittable
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

44. Delete Submittable

- **URL:** /submittables/{submittable_id}
- **Method:** DELETE
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - submittable_id (integer)

Responses:

- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully deleted submittable
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

45. Get all the submissions for a Submittable

- **URL:** /submittables/{submittable_id}/submissions
- **Method:** GET
- **Authorization:** OAuth2PasswordBearer
- **Request Body Schema:** null
- **Required Parameters:**
 - submittable_id (integer)

Responses:

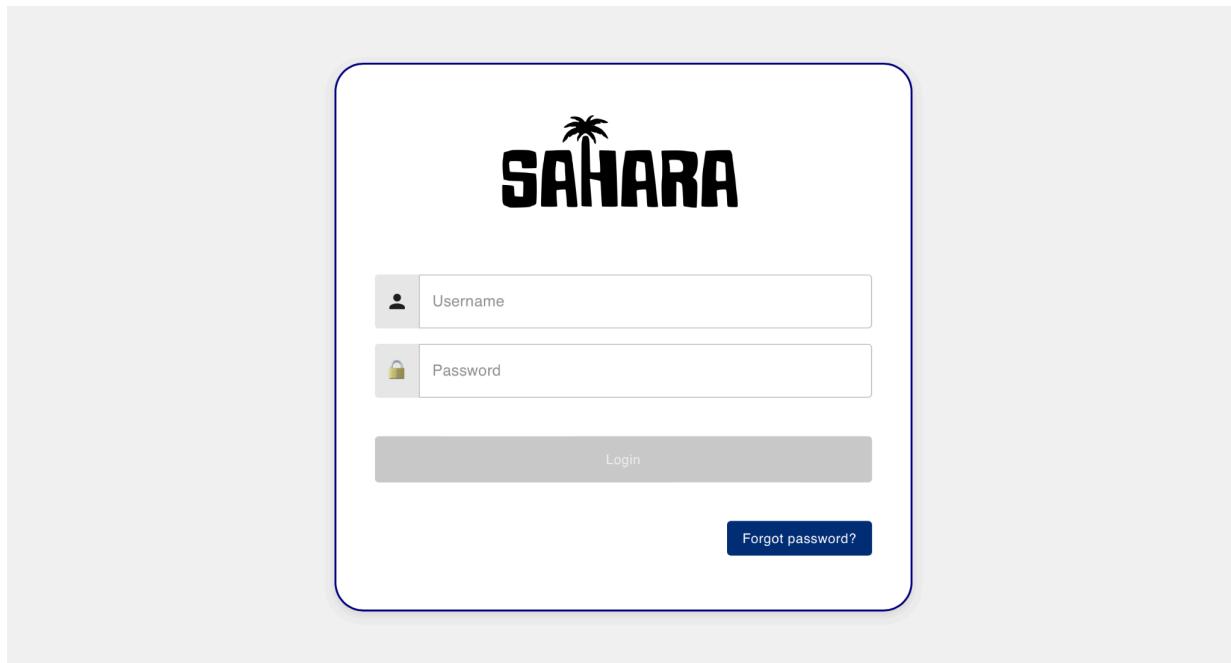
- **200 Successful Response:**
 - **Content Type:** application/json
 - **Description:** Successfully retrieved all submissions
 - **Response Body:** null
- **422 Validation Error:**
 - **Content Type:** application/json
 - **Description:** Input Validation failed

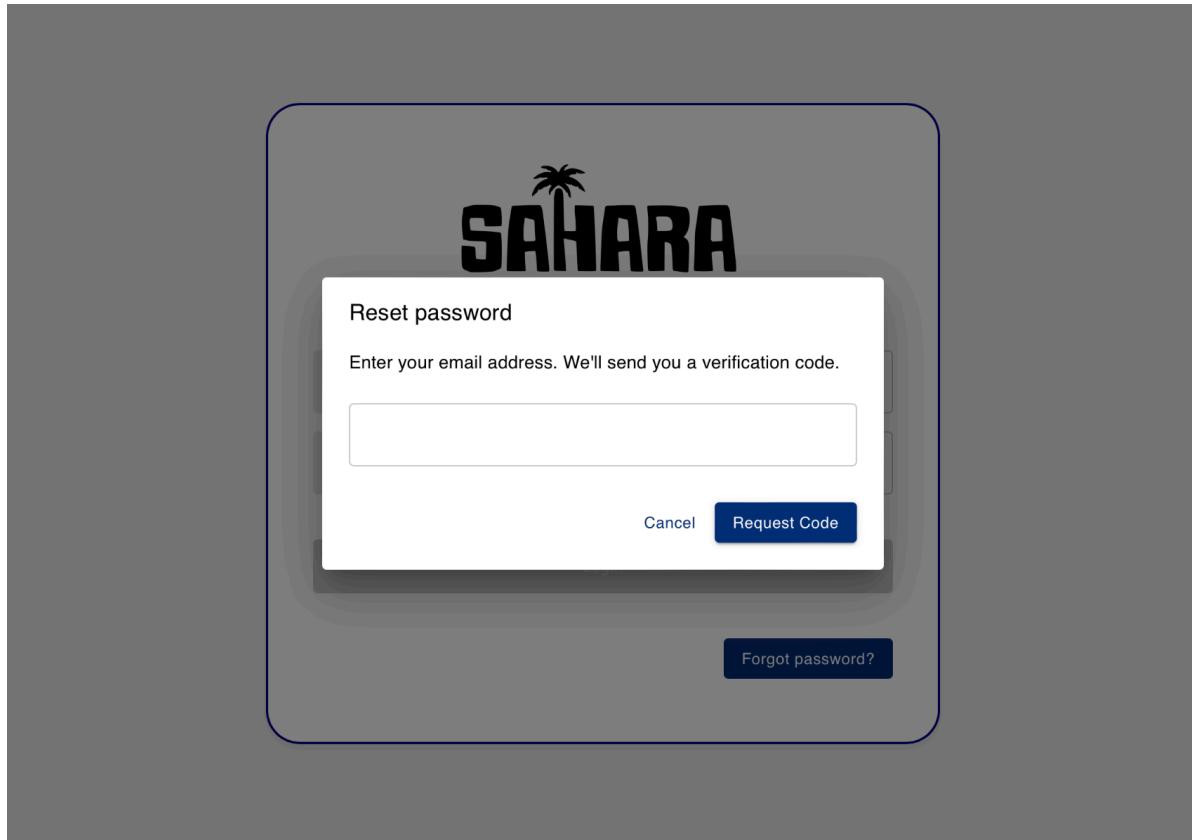
3 Completeness

Implemented Features

1. Authentication-

- Login and reset the password (after generating and verifying the **OTP**)





Your Password Reset Code - Sahara

From: Sahara Team
To: harshpreet23@iitk.ac.in
Date: Today 14:51

[Summary](#) [Headers](#) [Plain text](#)

Password Reset Verification

Hello,

You've requested to reset your password for your Sahara account.

Please use the following verification code to complete the process:

9 4 5 2 3 3

This code is valid for 10 minutes and can only be used once.

If you didn't request a password reset, please ignore this email or contact support if you have concerns.

This is an automated message, please do not reply directly to this email.

© 2025 Sahara Team

- The professor can add a list of TAs and Students to the course

The screenshot shows the SAHARA software interface. At the top right are user icons for 'R' and 'LOGOUT'. On the left is a dark blue sidebar menu with the following items: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People (with a dropdown arrow), Discussions, Feedback Form, and Calendar. The main content area has a title 'Add Students' and a large input field labeled 'Attach .csv file' with a 'UPLOAD' button below it.

The screenshot shows the SAHARA software interface. At the top right are user icons for 'LOGOUT'. On the left is a dark blue sidebar menu with the following items: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People (with a dropdown arrow), Discussions, Feedback Form, and Calendar. The main content area has a title 'Add TAs' and a large input field labeled 'Attach .csv file' with a 'UPLOAD' button below it.

- The professor can add other professors to the course

2. Privacy of Users-

- We have used **jwt tokens** and the **HS256** hashing algorithm to store the hashed passwords.
- The hashing algorithm we are using for the passwords is **bcrypt**.
- We have also generated a secret key to ensure the security of the user.
- Access to each page is determined by the role that is part of the jwt token generated.
- The generated token expires after 60 minutes.

3. Features Implemented in the Professor's Section

- Dashboard:

- Can view all the tabs, and can navigate to them.

The screenshot shows the SAHARA dashboard interface. On the left is a dark sidebar menu with the following items: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People, Discussions, Feedback Form, and Calendar. The main content area has a light background. At the top, it says "Good Evening," followed by a sub-header "Manage course activities, grade submissions, and track student progress." Below this is a section titled "Recent Announcements" with a "VIEW ALL" link. It lists four recent announcements with their details and a right-pointing arrow for more information: "Mid Semester Examination Today," "Quiz Grades Released Today," "Welcome to CS253! Today," and "Shell Scripting Announcement Today."

- Submissions:

- Create/Edit/Delete/View/Grade submissions

The screenshot shows the SAHARA software interface. On the left is a dark sidebar with various navigation options: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People, Discussions, Feedback Form, and Calendar. At the top right is a 'LOGOUT' button. The main content area has a header 'Course Home / Create Submission' and a title 'Create New Submittable'. Below the title is a sub-header 'Create a new submittable for your students to complete.' The form fields include:

- Name*: Implementation
- Maximum Score*: 100
- Enter the maximum possible score for this submittable
- Opens At*: Mar 04, 2025 12:00 AM
- Due Date*: Mar 28, 2025 12:00 AM
- Description*
- Reference File (Optional): ImplementationTemplate-CS253_917215b1-91d9-42e6-8295-afad2b2d9b9b.docx (with a 'CHANGE REFERENCE FILE' button)
- CREATE SUBMITTABLE** button

- Assignments:

- Create/Edit/Delete/View/Grade assignments

The screenshot shows the SAHARA software interface. On the left is a dark sidebar with various navigation options: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People, Discussions, Feedback Form, and Calendar. At the top right is a 'LOGOUT' button. The main content area has a header 'Course Home / Assignments' and a title 'Ongoing/Upcoming Documents'. Below the title is a 'Create Assignment' button. The list of assignments includes:

- Assignment a**: Due on: Mar 31, 2025, 04:00
Opens at: Mar 28, 2025, 00:35
- Assignment Bash Assignment**: Due on: Mar 29, 2025, 00:00
Opens at: Mar 16, 2025, 00:00

Details for the 'Bash Assignment' assignment:

- Description**
- System Instructions**: You are given an extra 10 minutes after due time to submit this assignment. However, please note that any submissions made after the due time are marked as late submissions.
- Results**: Results will be available after Mar 29, 2025, 00:00
- Reference Files**: shell_scripting_assignment_5d793725-ff31-47dc-8627-296b67d524c8.pdf
- Action buttons: View All Submissions, Update Assignment, Delete Assignment



Course Home / Create Assignment

Create New Assignment

Create a new assignment for your students to complete.

Name *

Maximum Score * Enter the maximum possible score for this assignment

Opens At * Due Date *

Description *

Reference File (Optional)



Course Home / Update Assignment

Update Assignment

Update the details of your assignment.

Title *

Deadline * Opens At (Optional)

Description *

The screenshots illustrate the SAHARA Learning Management System (LMS) interface. The top screenshot shows the 'Assignments' page for a specific assignment, displaying a single submission. The bottom screenshot shows the 'Ongoing/Upcoming Documents' page, listing two assignments with their respective due dates.

Top Screenshot: Assignments Page

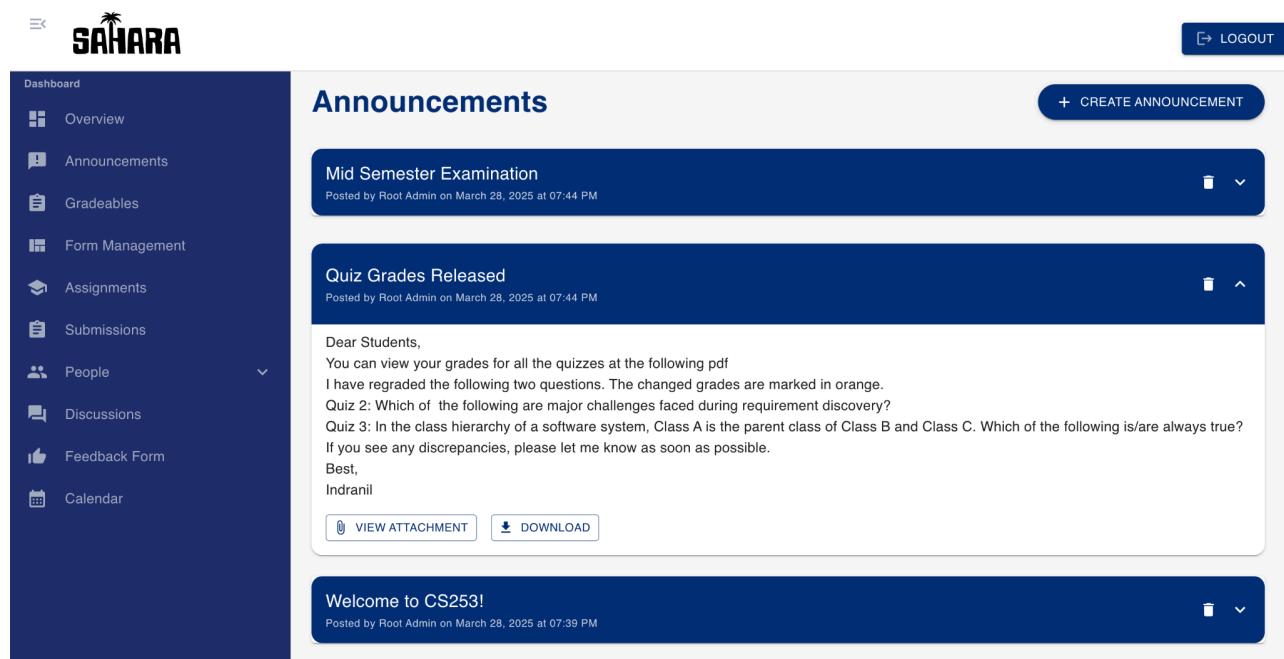
- Title:** C++ Assignment
- Submission Details:**
 - User ID: [redacted]
 - Submitted: 28/03/2025, 19:44:06
 - Score: Not graded
- Actions:** Download, Grade, Delete

Bottom Screenshot: Ongoing/Upcoming Documents Page

- Assignment 1:** C++ Assignment
 - Open at: Mar 11, 2025, 00:00
 - Due on: Mar 30, 2025, 00:00
 - Description
 - System Instructions: You are given an extra 10 minutes after due time to submit this assignment. However, please note that any submissions made after the due time are marked as late submissions.
 - Results: Results will be available after Mar 30, 2025, 00:00
 - Actions: View All Submissions, Update Assignment, Delete Assignment
- Assignment 2:** Bash Scripting Assignment
 - Open at: Mar 15, 2025, 00:00
 - Due on: Mar 29, 2025, 00:00

● Announcement:

- Can create and delete announcements
- Can attach files to each announcement
- Can use rich text features such as bullet points, numbering and text formatting in description of each announcement



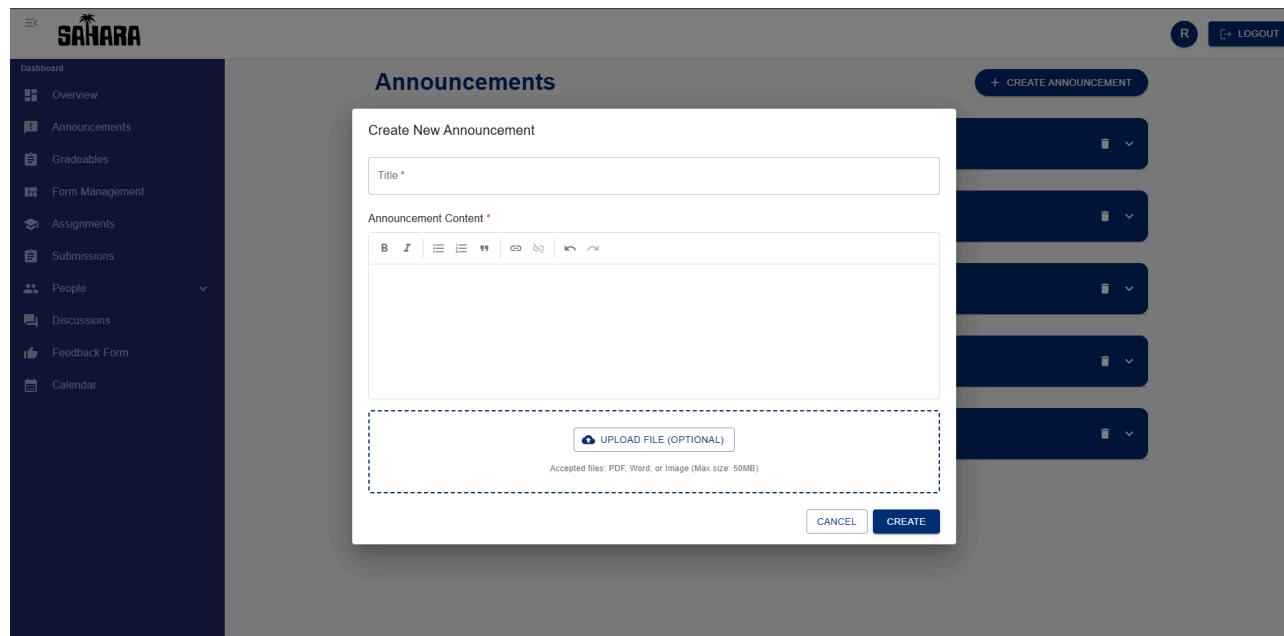
The screenshot shows the SAHARA software interface for managing announcements. The left sidebar contains navigation links for Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People, Discussions, Feedback Form, and Calendar. The main content area is titled "Announcements". It displays three announcements:

- Mid Semester Examination** (Posted by Root Admin on March 28, 2025 at 07:44 PM)

Dear Students,
You can view your grades for all the quizzes at the following pdf
I have regraded the following two questions. The changed grades are marked in orange.
Quiz 2: Which of the following are major challenges faced during requirement discovery?
Quiz 3: In the class hierarchy of a software system, Class A is the parent class of Class B and Class C. Which of the following is/are always true?
If you see any discrepancies, please let me know as soon as possible.
Best,
Indranil

[VIEW ATTACHMENT](#) [DOWNLOAD](#)
- Quiz Grades Released** (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Welcome to CS253!** (Posted by Root Admin on March 28, 2025 at 07:39 PM)

A blue button "+ CREATE ANNOUNCEMENT" is located in the top right corner of the announcements section.



The screenshot shows the "Create New Announcement" form overlaid on the announcements page. The form has a title "Create New Announcement" and fields for "Title *", "Announcement Content *", and an optional "UPLOAD FILE (OPTIONAL)" section. The "Announcement Content" field includes a rich text editor toolbar. The "CREATE" button is located at the bottom right of the form.

The screenshot shows the SAHARA software interface. On the left is a dark sidebar with various navigation options: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People, Discussions, Feedback Form, and Calendar. The main content area is titled 'Announcements'. It lists several announcements:

- Mid Semester Examination (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Quiz Grades Released (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Welcome to SAHARA (Posted by Root Admin on March 28, 2025 at 07:44 PM) - This announcement is currently selected, as indicated by a white background.
- Library Management System (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Shell Scripting Announcement (Posted by Root Admin on March 28, 2025 at 07:27 PM)

A modal dialog box is overlaid on the screen, centered over the selected announcement. The dialog is titled 'Delete Announcement' and contains the message: 'Are you sure you want to delete this announcement? This action cannot be undone.' At the bottom of the dialog are two buttons: 'CANCEL' (in grey) and 'DELETE' (in red).

- **People:**

- Upload **CSV** file of students (to add students)
- Upload **CSV** file of TAs (to add TAs)
- Upload **CSV** file of teams
- Get the skills from TAs

The screenshot shows the SAHARA software interface. On the left is a dark sidebar with various navigation options: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People, Discussions, Feedback Form, and Calendar. The main content area is titled 'Teaching Assistants'.

At the top of the main area, there is a 'Filter by Skills:' section with a 'Filter Mode' dropdown set to 'OR'. Below this are several skill tags: Java, MongoDB, Node.js, Python, React, and Spring Boot.

The main content area is titled 'TA List' and contains a table with the following data:

S. No.	Name	Email	Skills
1	Hemang Mohanlal Khatri	hemangmkhatri@cse.iitk.ac.in	React, Node.js, Python
2	Jeswaanth Gogula	jeswaanth@cse.iitk.ac.in	Java, Spring Boot, MongoDB

On the right side of the main content area, there are three buttons:

- ADD TAS
- TA SKILLS FORM
- TA-TEAM PAIRING

The screenshot shows the SAHARA software interface. On the left is a dark blue sidebar with various navigation options: Dashboard, Overview, Announcements, Gradeables, Form Management, Assignments, Submissions, People (selected), Discussions, Feedback Form, and Calendar. At the top right is a 'LOGOUT' button. The main content area has a title 'TEAMS' DETAILS' and a sub-section 'TEAMS DETAILS'. It displays a message 'Team Creation Status : Complete' with a due date 'Due : 05-02-2025' and a status 'Teams Formed : 17/17'. There are two search bars: 'Search Team' and 'Search Name, Roll No'. Below these are two tables: 'Formed Teams' and 'Beta-Test Pairs'. The 'Formed Teams' table shows a single team entry: Team Number 19, Team Name 'Team 19', and Members AYUSH GAUTAM, CHATLA SOWMYA SRI, SPARSH GUPTA, UJJWAL KUMAR, PRANSHU MANI TRIPATHI, ANANY DEV CHOUDHARY, SIDHARTH A S, LOKESH MEHRA, SIDDHARTH PATHAK, ABHISHEK KUMAR, ANAND CHUTANI. The 'Beta-Test Pairs' table has columns 'Team Number Pairs' and 'Team Names'. At the bottom, there is a link 'Download as Excel File'.

The screenshot shows the 'Teams Creation' page. The left sidebar is identical to the previous one. The main area has a title 'Teams Creation'. A file input field shows 'randoms.csv' selected, with a note 'Successfully selected file: randoms.csv'. Below this is a message: 'The CSV file should be in the format: team name - member1 - member2 - ... - member10'. At the bottom is a 'SUBMIT' button.

- Pair any valid number of TAs to each team automatically based on skill match
- Update any particular TA Team pairing
- Can view the **Team Details**
- Can update the TAs allotted to a specific team manually

SAHARA

Dashboard

- Overview
- Announcements
- Gradeables
- Form Management
- Assignments
- Submissions
- People
- Discussions
- Feedback Form
- Calendar

Teams

Number of TAs per team: ALLOCATE TAs

Team ID	Team Name	Skills	Allotted TA	Actions
19	Team 19	React, Node.js, Java	ta6, ta10, ta8	
18	Team 18	Java, MongoDB, React	ta1, ta2, ta4	
5	Team 5	Java, MongoDB, React	ta5, ta3, ta7	
20	Team 20	Python, Java, Node.js	ta6, ta10, ta9	
15	Team 15	React, Java, Spring Boot	ta8, ta1, ta2	
16	Team 16	MongoDB, Java, Spring Boot	ta4, ta5, ta3	
17	Team 17	React, MongoDB, Node.js	ta6, ta7, ta9	
13	Team 13	Java, Spring Boot, React	ta10, ta8, ta1	
6	Team 6	React, Spring Boot, Node.js	ta2, ta4, ta5	
12	Team 12	Python, React, MongoDB	ta3, ta7, ta9	
14	Team 14	Node.js, Python, React	ta6, ta10, ta8	
7	Team 7	Java, React, Spring Boot	ta1, ta2, ta4	

SAHARA

Dashboard

- Overview
- Announcements
- Gradeables
- Form Management
- Assignments
- Submissions
- People
- Discussions
- Feedback Form
- Calendar

Teams

Number of TAs per team: ALLOCATE TAs

Team ID	Team Name	Skills	Allotted TA	Actions
19	Team 19	React, Node.js, Java	ta6, ta10, ta8	
18	Team 18		ta1, ta2, ta4	
5	Team 5		ta5, ta3, ta7	
20	Team 20		ta6, ta10, ta9	
15	Team 15		ta8, ta1, ta2	
16	Team 16		ta4, ta5, ta3	
17	Team 17		ta6, ta7, ta9	
13	Team 13		ta10, ta8, ta1	
6	Team 6	React, Spring Boot, Node.js	ta2, ta4, ta5	

Edit TA Assignments

Assigned TAs: ta6, ta10, ta8

- ta6
- ta10
- ta8
- ta1
- ta2
- ta4
- ta5
- ta3
- ta7
- ta9

ID	Name	Email	Role
220004	AADI SINGH	aadis22@iitk.ac.in	Student
220010	AADYA DHIR	aadyad22@iitk.ac.in	Student
230011	AARADHYA ROHI	aaradhya23@iitk.ac.in	Student
230012	AARAV OSWAL	aaravoswal23@iitk.ac.in	Student
230015	AARSH JAIN	aarshjain23@iitk.ac.in	Student
230017	AADISH MADHIMDA CHATE	aadishchate23@iitk.ac.in	Student

Team ID	Team Name	Skills	Assigned TAs	Actions
1	Team 1	Python, Spring Boot, React	(ta2, ta4, ta5)	
2	Team 2	Python, Java, MongoDB	(ta5, ta3, ta7)	
3	Team 3	Python, Spring Boot, MongoDB	(ta8, ta1, ta2)	
4	Team 4	Node.js, MongoDB, React	(ta10, ta8, ta1)	
5	Team 5	Java, MongoDB, React	(ta5, ta3, ta7)	
6	Team 6	React, Spring Boot, Node.js	(ta2, ta4, ta5)	
7	Team 7	Java, React, Spring Boot	(ta1, ta2, ta4)	
8	Ravi and Friends	Node.js, Java	(ta3, ta7, ta9)	

- **Feedback:**

- The Professor can view each member's (from any team) feedback submission on the team project contribution
- Can export the team submissions as individual csv files or as an excel file with multiple sheets corresponding to multiple teams

Team Feedback Details - Ravi and Friends

Feedback Matrix

Submitter	AARAV OSWAL	ACHYUTH HARIKRISHNA WARRIER	ASHIK STENNY	DIVI POTLUKUCHI	HARSHPREET KAUR	POTHUGANTI NIKHIL	R.CHARAN	RAVI ARORA	SPANDAN PATI	VEDHANTH BALASUBRAMANIAN
ASHIK STENNY 28/03/2025	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
ACHYUTH HARIKRISHNA WARRIER 28/03/2025	5%	5%	20%	10%	10%	10%	10%	10%	10%	10%
Mean	7.5%	7.5%	15.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%
Variance	6.3%	6.3%	25.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Detailed Submissions

Submission by ASHIK STENNY
Submitted on: 28/03/2025, 14:24:37

Submission by ACHYUTH HARIKRISHNA WARRIER
Submitted on: 28/03/2025, 14:26:22

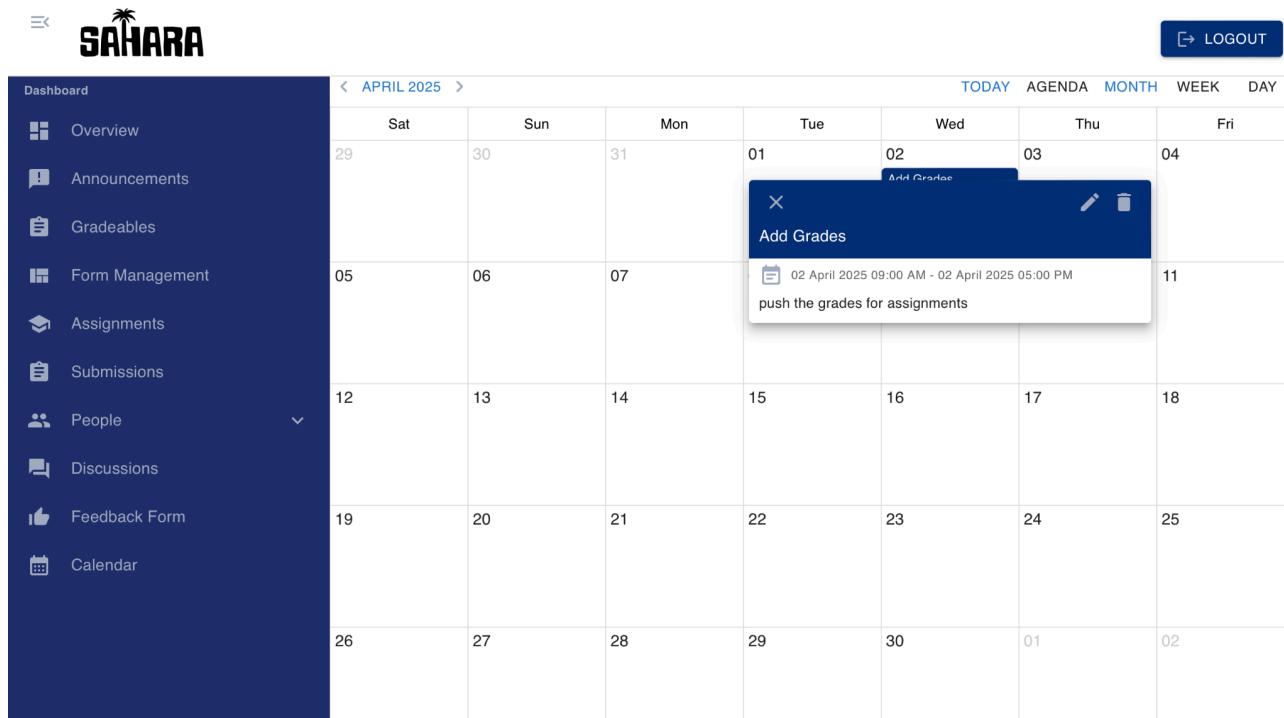
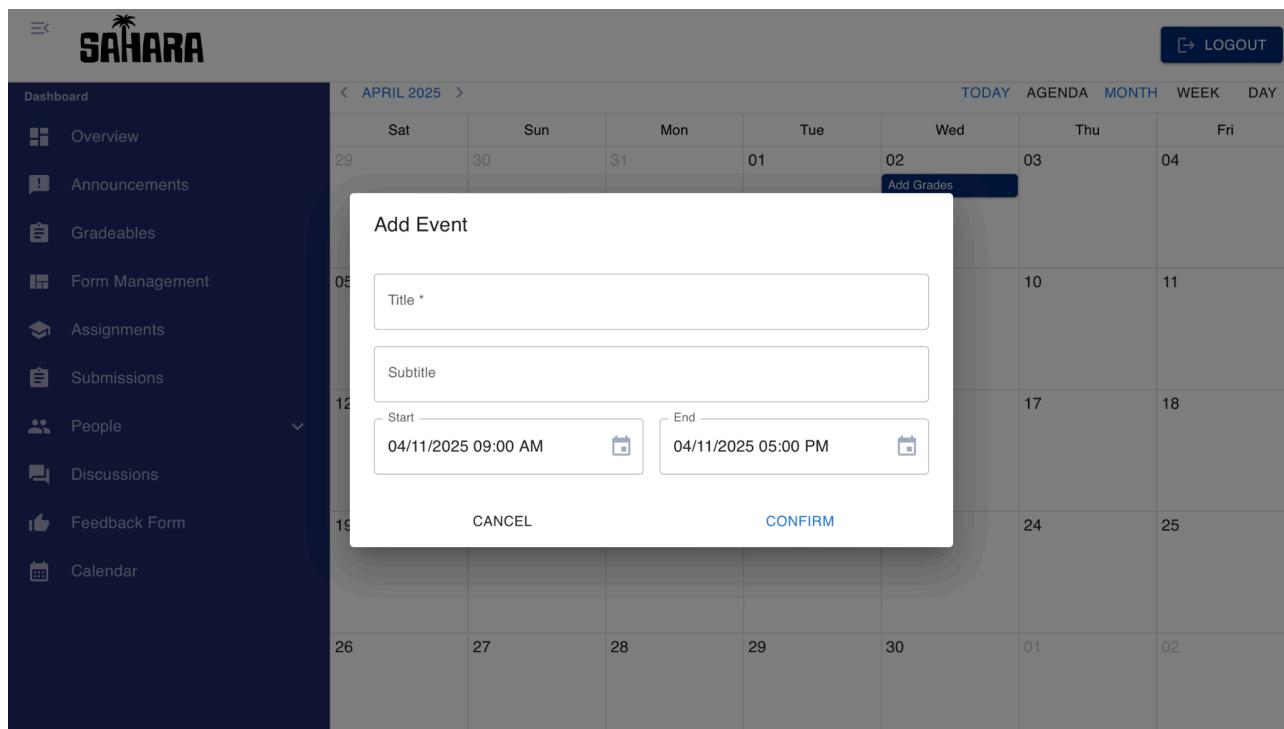
CLOSE

- **Calendar:**

- The Professor can add and view his/her personal events.
- The events are editable as well.

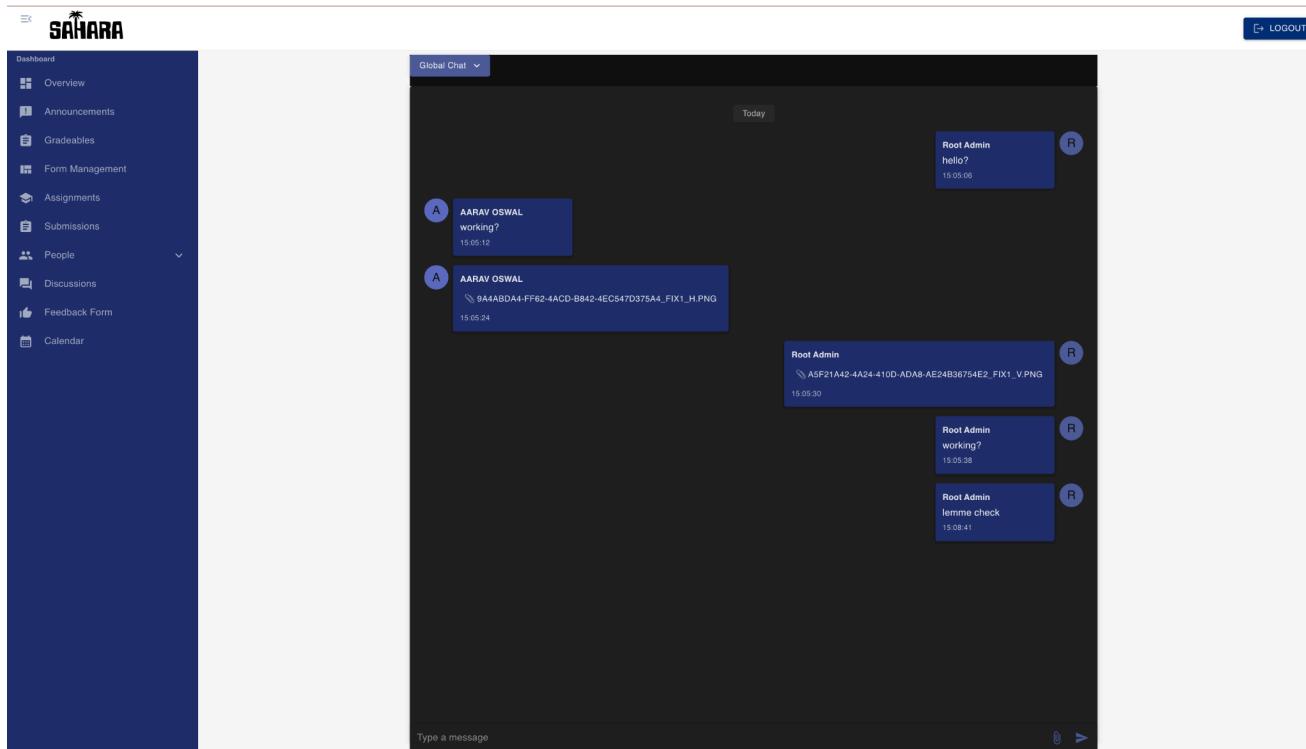
MARCH 2025

Sat	Sun	Mon	Tue	Wed	Thu	Fri
01	02	03	04	05	06	07
08	09	10 Meeting with TAs	11	12	13	14
15	16	17	18	19	20 Team 8 Meet	21
22	23	24	25	26	27	28
29	30	31	01	02 Add Grades	03	04



- **Discussion forum:**

- Can participate in global chat, which includes all the registered users



- **Gradeables:**

- Manage and view gradeables

Name	Maximum Points
Quiz 1	6
Quiz 2	6
Midsem	100

- Create and upload scores for gradables

The screenshot shows the SAHARA software interface. On the left is a dark blue sidebar with a navigation menu:

- Dashboard
- Overview
- Announcements
- Gradeables
- Form Management
- Assignments
- Submissions
- People
- Discussions
- Feedback Form
- Calendar

The main content area has a header "Course Home / Gradeables". Below it is a blue banner with the text "Create Gradeable and Upload Scores" and a sub-instruction "Create a new gradeable by providing a title, maximum points, and uploading a CSV file with student scores." The form contains fields for "Test/Assignment Title" (with placeholder "Quiz 1") and "Maximum Points" (set to 100). There is also a section for "Upload Scores" with a "Browse" button and a note about CSV file size. At the bottom is a blue "Create Gradeable" button.

- View Individual Scores as well as Statistics

The screenshot shows the SAHARA software interface for Quiz 2. The sidebar is identical to the previous screenshot. The main content area has a blue banner for "Quiz 2" with "204 student submissions" and a "Mean Score: 5.20".

Student Results

Student ID	Name	Score
220004	AADI SINGH	5
220010	AADYA DHIR	5
230011	AARADHYA ROHI	4
230012	AARAV OSWAL	6
230015	AARSH JAIN	6
230017	AARUSH NARENDRA GHATE	4
220020	AAYUSH GAUTAM	6

Score Distribution

A bar chart showing the number of students versus their scores. The x-axis represents scores from 0 to 6, and the y-axis represents the number of students from 0 to 120. The distribution is skewed right, with most students scoring between 5 and 6.

Score Statistics

Mean Score 5.20	Median Score 6.00	Mode Score 6.00	Variance 1.71
--------------------	----------------------	--------------------	------------------

4. Features Implemented in the TA's Section-

- **Dashboard:**
- **Submissions:**
 - Create/Edit/Delete/View Grade
- **Announcements:**
 - View all the announcements
 - Can view attached file in a new browser tab
 - Can download the attached file to local machine

The screenshot shows the SAHARA software interface. At the top, there is a navigation bar with icons for Home, Dashboard, Overview, Announcements, Team, Documentation, Assignments, Forms, Discussions, Feedback Form, and Calendar. On the far right of the top bar are 'R' and 'LOGOUT' buttons. Below the navigation bar is a header with the 'SAHARA' logo and a search bar. The main content area is titled 'Announcements'. It displays five announcement cards:

- Mid Semester Examination** (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Quiz Grades Released** (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Welcome to CS253!** (Posted by Root Admin on March 28, 2025 at 07:39 PM)

Attached is the FCH for this course. Best of luck!

[VIEW ATTACHMENT](#) [DOWNLOAD](#)
- Library Management System** (Posted by Root Admin on March 28, 2025 at 07:39 PM)
- Shell Scripting Announcement** (Posted by Root Admin on March 28, 2025 at 07:27 PM)

- **Assignments:**
 - Create/Edit/Delete/View/Grade assignments
- **Submissions:**
 - Create/Edit/Delete/View/Grade submissions
- **Feedback:**
 - The TA can view each member's (from any team) feedback submission on the team project contribution
- **Calendar:**
 - TAs can add and view his/her personal events.
 - The events are editable as well.
- **Discussion Form:**
 - Can participate in global chat which includes all the registered users.
 - Can participate in chat with the team assigned.
 - A TA can have more than 1 team assigned, and hence can have multiple chats.
- **Skill Form**
 - Any TA can choose their skills that they are comfortable with and update it

5. Features Implemented in the Students' Section-

- Profile:

The screenshot shows the profile page of the SAHARA application. At the top right, there is a user icon with the letter 'A' and a 'LOGOUT' button. The main header displays the user's name, 'ACHYUTH HARIKRISHNA WARRIER', and their handle, '@achyuthw23'. On the left, a sidebar menu lists various features: Dashboard, Overview, Announcements, Team, Documentation, Assignments, Forms, Discussions, Feedback Form, and Calendar. The profile information section contains fields for Email (achyuthw23@iitk.ac.in), Role (student), and Team (Ravi and Friends). Account details show the User ID 230052.

- Dashboard:

The screenshot shows the SAHARA application dashboard. At the top right, there is a 'LOGOUT' button. The main area features a greeting, 'Good Evening,' and a message encouraging users to stay organized and manage their work. Below this, a 'Recent Announcements' section lists five items with 'VIEW ALL' links. The announcements are: 'Mid Semester Examination' (Today), 'Quiz Grades Released' (Today), 'Welcome to CS253!' (Today), 'Library Management System' (Today), and 'Shell Scripting Announcement' (Today).

- **Announcements:**

- View announcements
- Can view the attached file in a new browser tab
- Can download the attached file

- **Team:**

- View team members

The screenshot shows the SAHARA software interface. On the left is a dark sidebar with navigation links: Dashboard, Overview, Announcements, Team, Documentation, Assignments, Forms, Discussions, Feedback Form, and Calendar. The main content area is titled "Team Members" and displays a team named "Ravi and Friends" with 10 members. A section titled "Team Skills" shows checkboxes for Node.js, Java, and MongoDB. Below this is a table listing team members with their names and emails.

Name	Email
AARAV OSWAL	aaravoswal23@iitk.ac.in
ACHYUTH HARIKRISHNA WARRIER	achyuthw23@iitk.ac.in
ASHIK STENNY	ashiks23@iitk.ac.in
DIVI POTUKUCHI	divip23@iitk.ac.in
HARSHPREET KAUR	harshpreet23@iitk.ac.in
POTHUGANTI NIKHIL	nikhilp23@iitk.ac.in
R.CHARAN	rcharan23@iitk.ac.in
RAVI ARORA	raviora23@iitk.ac.in

- Add or remove skills for teams for pairing

This screenshot shows the same SAHARA interface as above, but with a modal dialog box titled "Manage Team Skills" overlaid on the "Team Members" page. The dialog box contains a list of skills: Node.js, Java, and MongoDB, with "Node.js" and "Java" currently selected. A message in the dialog says, "Select the skills that best represent your team's expertise. These skills will help TAs better understand your team's capabilities." At the bottom of the dialog are "CANCEL" and "UPDATE SKILLS" buttons.

The screenshot shows the SAHARA Learning Management System (LMS) home page. The header features the SAHARA logo and navigation links for 'Dashboard', 'Logout', and 'Announcements'. The sidebar on the left contains links for 'Overview', 'Announcements', 'Team', 'Documentation', 'Assignments', 'Forms', 'Discussions', 'Feedback Form', and 'Calendar'. The main content area is titled 'Announcements' and lists several recent posts:

- Mid Semester Examination** (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Quiz Grades Released** (Posted by Root Admin on March 28, 2025 at 07:44 PM)
- Welcome to CS251!** (Posted by Root Admin on March 28, 2025 at 07:39 PM)
 - Attached is the FCH for this course. Best of luck!
 - [VIEW ATTACHMENT](#)
 - [DOWNLOAD](#)
- Library Management System** (Posted by Root Admin on March 28, 2025 at 07:39 PM)
- Shell Scripting Announcement** (Posted by Root Admin on March 28, 2025 at 07:27 PM)

- **Assignments:**
 - View/attempt assignments

The screenshot shows the SAHARA LMS assignment page. The header includes the SAHARA logo and navigation links for 'Logout'. The sidebar on the left is identical to the home page. The main content area shows the following information:

Course Home / Submissions

Project Progress
0% of 2 documents submitted ~ 2 documents remaining

Ongoing/Upcoming Documents

Assignment	Description	System Instructions	Results
C++ Assignment	Description You are given an extra 10 minutes after due time to submit this assignment. However, please note that any submissions made after the due time are marked as late submissions.	System Instructions You are given an extra 10 minutes after due time to submit this assignment. However, please note that any submissions made after the due time are marked as late submissions.	Results Results will be available after Mar 30, 2025, 00:00

[SUBMIT DOCUMENT](#)

The image consists of two vertically stacked screenshots of a software application named "SAHARA".

Screenshot 1: Assignment Submission

This screenshot shows a file selection dialog box over a web page. The dialog box is titled "Open" and shows a list of files in the "Downloads" folder. The files listed are:

Name	Date modified	Type
file	27-03-2025 04:42	Text Document
file (1)	27-03-2025 20:51	Text Document
download	27-03-2025 20:58	Adobe Acrobat D...
github-recovery-codes	20-03-2025 16:40	Text Document
Copy of Copy of ELS25-26	19-03-2025 00:06	Adobe Acrobat D...
Copy of Copy of ELS25-26 (1)	19-03-2025 00:41	Adobe Acrobat D...

The web page below the dialog box displays assignment details for "Bash Scripting Assignment". It shows the assignment opens at Mar 15, 2025, 00:00 and is due on Mar 29, 2025, 00:00. A "SUBMIT DOCUMENT" button is visible.

Screenshot 2: Project Progress

This screenshot shows a progress summary for a project. It indicates "50%" completion and "1 of 2 documents submitted". Below this, it lists an "Ongoing/Upcoming Documents" for "C++ Assignment", which opens at Mar 11, 2025, 00:00 and is due on Mar 30, 2025, 00:00. It includes sections for "Description", "System Instructions", "Uploaded Answer Files" (with a file named "02-02-25, 836 PM Microsoft Lens.pdf"), and "Results". A "DELETE SUBMISSION" button is located at the bottom right of the document card.

- **Submissions:**
 - View/make submissions

The screenshot shows the SAHARA software interface. At the top, there's a navigation bar with a logo, a search bar containing 'SAHARA', and a 'LOGOUT' button. The main area has a dark blue sidebar on the left with links: Dashboard, Overview, Announcements, Forms, Discussions, Feedback Form, and Calendar. The main content area has a header 'Project Progress' with a 100% completion circle and a message '1 of 1 documents submitted'. Below this is a section titled 'Ongoing/Upcoming Documents' with a sub-section for 'Implementation Document'. It shows the assignment details: 'Assignment: Implementation Document', 'Opens at: Mar 12, 2025, 00:00', 'Due on: Mar 31, 2025, 23:59', and a description of the requirements. It also lists 'Question Files' (ImplementationTemplate-CS253.docx), 'Uploaded Answer Files' (Implementation_Document (1).pdf), and 'Results' (available after Mar 31, 2025, 23:55).

- **Feedback:**

- View and submit the feedback form which the prof releases at the end of the course.

The screenshot shows the SAHARA software interface. At the top, there's a navigation bar with a logo, a search bar containing 'SAHARA', and a 'LOGOUT' button. The main area has a dark blue sidebar on the left with links: Dashboard, Overview, Announcements, Forms, Discussions, Feedback Form, and Calendar. The main content area is titled 'Team Contribution Feedback - Ravi and Friends'. It shows a table of team members and their contribution percentages. The table has columns for 'Team Member', 'Contribution Percentage (%)', and 'Remarks'. The team members listed are VEDHANT BALASUBRAMANIAN, DIVI POTHUKUCHI (You), ASHIK STENNY, HARSHPREET KAUR, R.CHARAN, AARAV OSWAL, SPANDAN PATI, ACHYUTH HARIKRISHNA WARRIOR, POTHUGANTI NIHIL, and RAVI ARORA. Each row includes a '0.00' contribution value and a 'Add remarks' button. At the bottom of the table, it says 'Total 100.00%' and has a 'SUBMIT FEEDBACK' button.

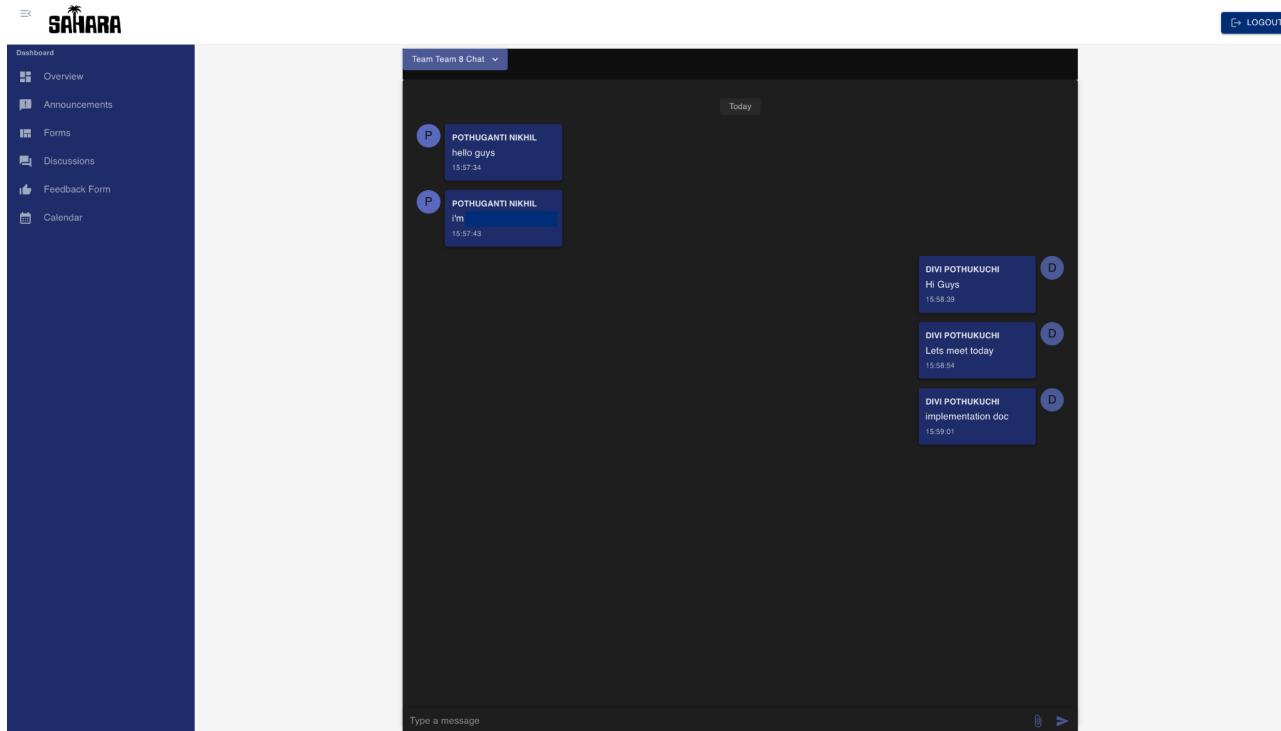
- **Calendar:**

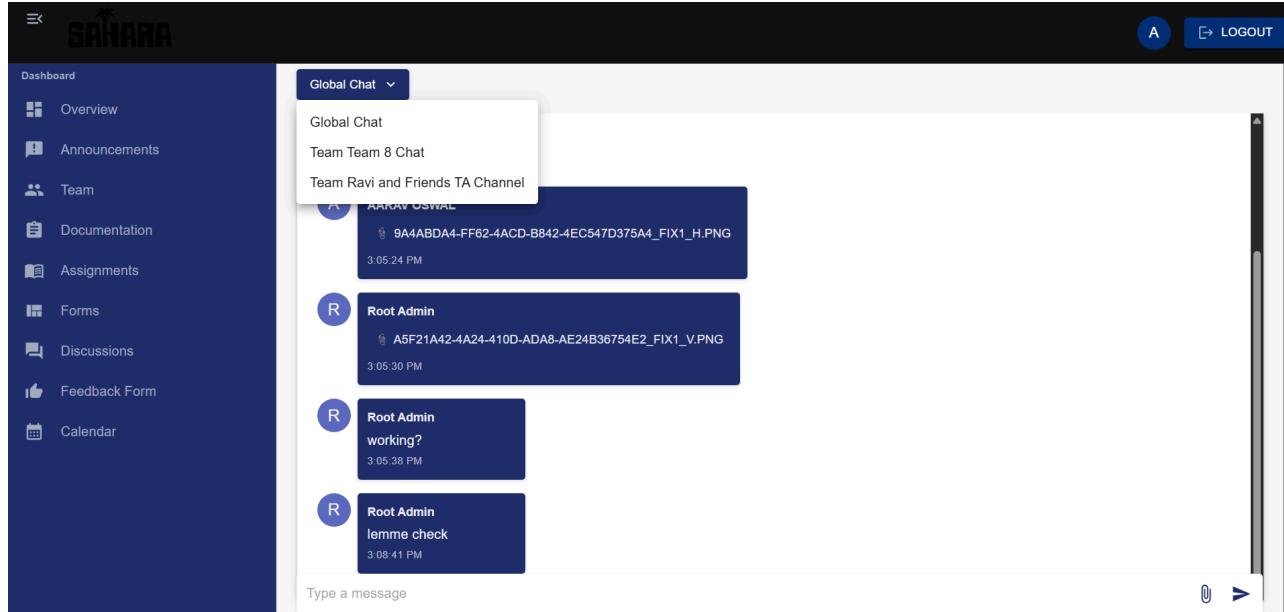
- Students can view all the events present in the calendar.

- Students can add personal events on their calendar.
- The events are editable as well.

- **Discussion Form:**

- Can participate in global chat which includes all the registered users
- Can participate in chat with the team and TA assigned to them.
- Can participate in chat with just the team members





Future Development Plans

1. Team Creation via Invite System

- Implement a feature that allows users to create teams using an invite-based system.
- Allow the professor to manually edit teams.

2. AI-Assisted Grading

- Integrate AI-powered grading to streamline the evaluation of gradable tasks.

3. Message Tagging and Anonymous Texting in Forums

- Enable the ability to tag messages.
- Introduce an option for anonymous texting within forums.
- Enabling censoring text messages in forums.

4. Activity Notifications and Deadline Reminders

- Implement automatic notifications/emails for key activities, including:
- Creation of gradable task
- Grading of assignments
- TA assignments
- Beta team assignments
- Deadline reminders
- General fixing of bugs and improvement of software through rigorous testing

5. Beta Testing Functionalities

- Pair teams for Beta Testing
- Create forms of the Beta Testing pairs to facilitate communication

6. Quiz Functionality

- Should be able to create quizzes - Questions with respective expected answers
- Students should be able to attend quiz within deadline and their responses should be validated and respective scores should be given

7. Global Events

- Professors will be able to add global events in the calendar.
- All the users will have global events in their calendar.
- The global events can be added and edited by the professor only.

Reserved User_ID's for testing

The following is a list of credentials so that the special dashboards for the different types of users can be tested

Username	Password	Role
prof123	prof123	prof
student123	student123	student
ta123	ta123	TA

Appendix A - Group Log

Date	Timings	Duration	Minutes
3rd March	6-9 PM	3 hours	Discussed the components to be present and the division of work on these components
15th March	5-7.30 PM	2.5 hours	Discussed the backend part of these components
18th March	10 PM- 2 AM	4 hours	Discussed the combination of these components and started work on the front end
20th March	10 PM- 2 AM	4 hours	Continued work on the front end
23rd March	10 PM- 2 AM	4 hours	Started work on integrating front-end with back-end
25th March	8 PM- 2 AM	6 hours	Continued integrating
26th March	6PM - 8 AM	14 hours	Fixed minor bugs in the code

27th March	7 PM- Midnight	5 hours	Made the UI consistent and visually appealing
28th March	Midnight - Midnight	24 hours	Worked on the Implementation Document and bettered UI