Tracking Student Progress in Software Projects

Software Engineering Project (BSCS3001)

Milestone 4

API Endpoints

Submitted By

Saarthak Saran (21f3001154)

Prachi Tiwari (21f2001019)

Deepanshu Kumar Pathak (21f3001217)

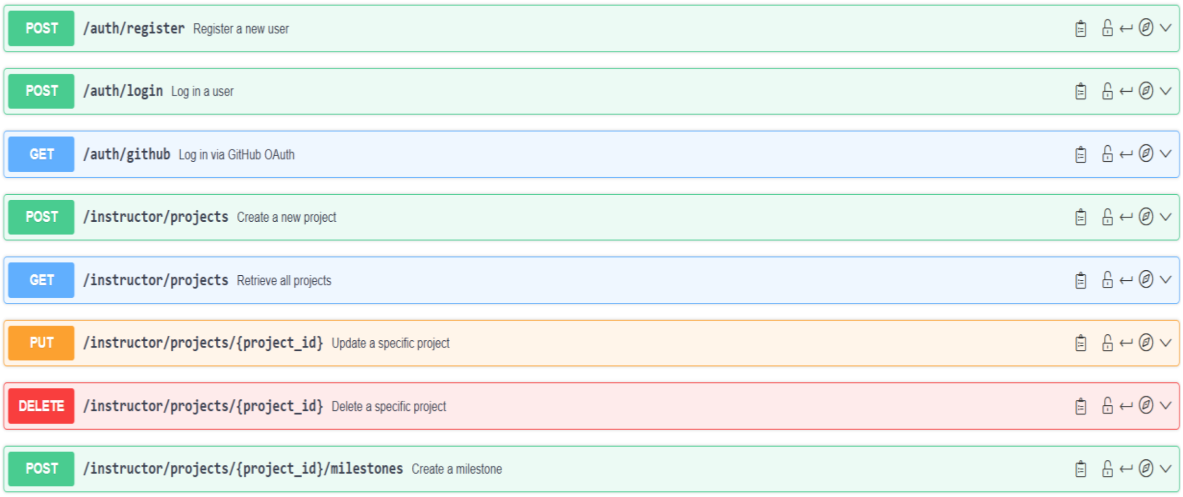
Abhijeet Kumar Garg (21f1003267)

Himanshu Kumar (21f3001746)

Aditya Singh (22f1000873)

API Documentation

Method	Endpoint	Description
POST	/auth/register	Registers a new user as a student or instructor.
POST	/auth/login	Logs in a user and provides a JWT token.
GET	/auth/github	Logs in a user via GitHub OAuth.
POST	/instructor/projects	Creates a new project.
GET	/instructor/projects	Retrieves all projects created by the instructor.
PUT	/instructor/projects/{project_id}	Updates a specific project.
DELETE	/instructor/projects/{project_id}	Deletes a specific project.
POST	/instructor/projects/{project_id}/milestones	Creates a milestone for a project.
PUT	/instructor/projects/{project_id}/milestones/{milestone_id}	Updates a specific milestone.
DELETE	/instructor/ projects/{project_id}/milestones/{milestone_id}	Deletes a specific milestone.
POST	/instructor/ projects/{project_id}/milestones/{milestone_id}/feedback	Adds manual feedback to a milestone.
GET	/instructor/ projects/{project_id}/milestones/{milestone_id}/ai-feedback	Generates AI feedback for a milestone.
GET	/instructor/teams/{team_id}/github/commits	Retrieves GitHub commit history for a team's repository.
GET	/student/github/commits	Retrieves GitHub commit history for the student's team.
GET	/student/projects	Retrieves all projects assigned to the student's team.
GET	/student/milestones	Retrieves milestones assigned to the student's team.
POST	/student/milestones/{milestone_id}/submit	Submits a milestone with a submission URL.
GET	/student/feedback	Retrieves feedback (manual and AI) for milestones.



POST /auth/register Register a new user

POST /auth/login Log in a user

GET /auth/github Log in via GitHub OAuth

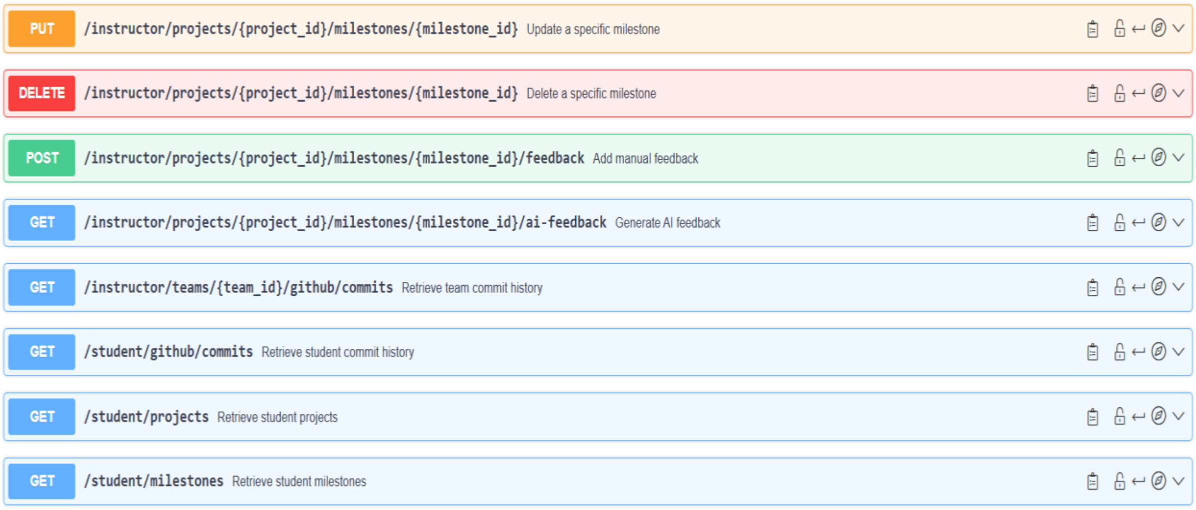
POST /instructor/projects Create a new project

GET /instructor/projects Retrieve all projects

PUT /instructor/projects/{project_id} Update a specific project

DELETE /instructor/projects/{project_id} Delete a specific project

POST /instructor/projects/{project_id}/milestones Create a milestone



PUT /instructor/projects/{project_id}/milestones/{milestone_id} Update a specific milestone

DELETE /instructor/projects/{project_id}/milestones/{milestone_id} Delete a specific milestone

POST /instructor/projects/{project_id}/milestones/{milestone_id}/feedback Add manual feedback

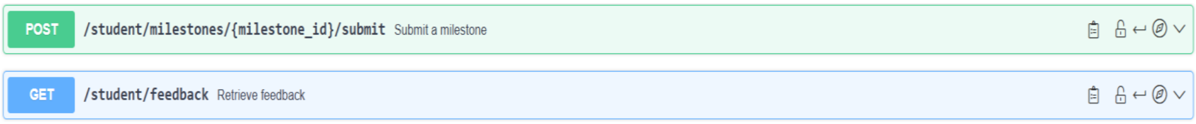
GET /instructor/projects/{project_id}/milestones/{milestone_id}/ai-feedback Generate AI feedback

GET /instructor/teams/{team_id}/github/commits Retrieve team commit history

GET /student/github/commits Retrieve student commit history

GET /student/projects Retrieve student projects

GET /student/milestones Retrieve student milestones



POST /student/milestones/{milestone_id}/submit Submit a milestone

GET /student/feedback Retrieve feedback

1. Authentication Endpoints

1.1 Register a New User

- Method: `POST`
- Endpoint: `/auth/register`
- Description: Registers a new user as either a student or an instructor.
- Request Body:

  ```
  {
    "username": "string",
    "email": "string",
    "password": "string",
    "role": "string" // Allowed values: "student", "instructor"
  }
  ```

- Responses:
  - 201: User successfully registered.
  - 400: Invalid input or user already exists.

1.2 Log in a User

- Method: `POST`
- Endpoint: `/auth/login`
- Description: Logs in a user and returns a JWT token.
- Request Body:

  ```
  {
    "email": "string",
    "password": "string"
  }
  ```

- Responses:
  - 200: JWT token provided upon successful login.
  - 401: Invalid credentials.

1.3 GitHub OAuth Login

- Method: `GET`
- Endpoint: `/auth/github`
- Description: Logs in a user via GitHub OAuth.
- Responses:
  - 200: Successfully authenticated.
  - 401: GitHub OAuth failed.

2. Project and Milestone Management (Instructor)

2.1 Create a Project

- Method: `POST`
- Endpoint: `/instructor/projects`
- Description: Creates a new project for a course.
- Request Body:

  ```
  {
    "title": "string",
    "description": "string",
  }
  ```

- Responses:
  - 201: Project successfully created.
  - 400: Invalid input.

2.2 Retrieve All Projects

- Method: `GET`
- Endpoint: `/instructor/projects`
- Description: Retrieves all projects created by the instructor.
- Responses:
  - 200: Returns a list of projects.

2.3 Update a Project

- Method: `PUT`
- Endpoint: `/instructor/projects/{project_id}`
- Description: Updates an existing project.
- Path Parameters:
  - `project_id` (integer): ID of the project to be updated.
- Request Body:

  ```
  {
    "title": "string",
    "description": "string"
  }
  ```

- Responses:
  - 200: Project successfully updated.
  - 404: Project not found.

2.4 Delete a Project

- Method: `DELETE`
- Endpoint: `/instructor/projects/{project_id}`
- Description: Deletes a specific project.
- Path Parameters:
  - `project_id` (integer): ID of the project to delete.
- Responses:
  - 204: Project successfully deleted.
  - 404: Project not found.

2.5 Create a Milestone

- Method: `POST`
- Endpoint: `/instructor/projects/{project_id}/milestones`
- Description: Creates a milestone under a project.
- Path Parameters:
  - `project_id` (integer): ID of the project for which the milestone is created.
- Request Body:

```
{
  "title": "string",
  "description": "string",
  "due_date": "YYYY-MM-DD"
}
```

- Responses:
  - 201: Milestone successfully created.
  - 404: Project not found.

2.6 Update a Milestone

- Method: `PUT`
- Endpoint:`/instructor/projects/{project_id}/milestones/{milestone_id}`
- Description: Updates a specific milestone.
- Path Parameters:
  - `milestone_id` (integer): ID of the milestone to update.
- Request Body:

```
{
  "title": "string",
  "description": "string",
  "due_date": "YYYY-MM-DD"
}
```

- Responses:
  - 200: Milestone successfully updated.
  - 404: Milestone not found.

2.7 Delete a Milestone

- Method: `DELETE`
- Endpoint:`/instructor/projects/{project_id}/milestones/{milestone_id}`
- Description: Deletes a specific milestone.
- Path Parameters:
  - `milestone_id` (integer): ID of the milestone to delete.
- Responses:
  - 204: Milestone successfully deleted.
  - 404: Milestone not found.

---

3. Feedback Management (Instructor)

3.1 Add Manual Feedback

- Method: `POST`
- Endpoint:`/instructor/projects/{project_id}/milestones/{milestone_id}/feedback`
- Description: Adds manual feedback to a milestone.
- Path Parameters:
  - `milestone_id` (integer): ID of the milestone for which feedback is added.
- Request Body:

  ```
  {
    "feedback": "string"
  }
  ```

- Responses:
  - 201: Feedback successfully added.
  - 404: Milestone not found.

3.2 Generate AI Feedback

- Method: `GET`
- Endpoint:`/instructor/projects/{project_id}/milestones/{milestone_id}/ai-feedback`
- Description: Generates AI-based feedback for a milestone.
- Path Parameters:

    - `milestone_id` (integer): ID of the milestone.
- Responses:
    - 200: AI feedback generated.
    - 404: Milestone not found.

---

4. GitHub Commit History

4.1 Retrieve Team Commit History (Instructor)

- Method: `GET`
- Endpoint: `/instructor/teams/{team_id}/github/commits`
- Description: Retrieves the commit history for a team's repository.
- Path Parameters:
    - `team_id` (integer): ID of the team.
- Responses:
    - 200: A list of commits.
    - 404: Repository or team not found.

4.2 Retrieve Student Commit History

- Method: `GET`
- Endpoint: `/student/github/commits`
- Description: Retrieves GitHub commit history for the student's team.
- Responses:
    - 200: A list of commits.
    - 404: Repository not found.

---

5. Student Functionalities

5.1 Retrieve Student Projects

- Method: `GET`
- Endpoint: `/student/projects`
- Description: Retrieves all projects assigned to the student's team.
- Responses:
    - 200: A list of projects.

5.2 Retrieve Student Milestones

- Method: `GET`
- Endpoint: `/student/milestones`
- Description: Retrieves milestones assigned to the student's team.
- Responses:
  - 200: A list of milestones.

5.3 Submit a Milestone

- Method: `POST`
- Endpoint: `/student/milestones/{milestone_id}/submit`
- Description: Submits a milestone with a submission URL.
- Path Parameters:
  - `milestone_id` (integer): ID of the milestone.
- Request Body:

```
{
  "submission_url": "string",
 "comments": "string"

}
```

- Responses:
  - 200: Milestone successfully submitted.
  - 404: Milestone not found.

5.4 Retrieve Feedback

- Method: `GET`
- Endpoint: `/student/feedback`
- Description: Retrieves feedback (manual and AI) for milestones.
- Responses:
  - 200: A list of feedback.

Error Codes

Code	Description
200	OK
201	Created
204	No Content
400	Bad Request
401	Unauthorized
404	Not Found
500	Internal Server Error

Authentication

All endpoints, except /auth/register and /auth/login, require a valid JWT token for authorization.