

BTP-2022

Group-2

Students Academic Projects Management System

Guided by:

Prof. JayPrakash Lalchandani

Team:

201801207: Vandit Bhut

201801457: Shivangi Zala

Description

- We will create a website for Professors and students of Daiict. Only Daiict students and Professors can log in to this website with the help of their Daiict G-Suite account. Professors will be able to log in and make changes (create or withdraw) to the projects and other details. This website will provide data for upcoming projects. Students will log in and apply (as a team or individuals) to the projects in their areas of interest. Professors can either accept or decline the request of students after the selection process. Professors will receive notifications when a student applies, When someone withdraws the application, and When someone takes on another project. Every team member or student will receive notifications when the professor accepts/declines, a student adds you as a team member, or while applying and scheduling a meeting. Users will be able to edit their profile pages. Users also see their timetables. Professors also Add project status, minutes of meetings, and marks of every meeting. Students see all the information added by a professor. Students and professors will be able to manage their past and current projects. Users of this website are students and faculty mentors of Daiict college.

Functional Requirements

Student(daiict & non-daiict)(user = student)

- Login
 - Users should be able to login only with daiict email
- Logout
- Profile
 - Users should be able to edit and save the profile information
- help
 - Users should be able to view information on how to use this site and other helplines.
- Home page
 - Users should be able to search projects from the search bar according to the professor's name and topic. Users should be able to see all the projects and apply to them as individuals and teams.
 - Users should be able to see the total number of applications for that project on the project card.
 - Users should be able to withdraw the application.
 - In the case of a team, the user who applies will only be able to withdraw it.
- My projects
 - Users should be able to see all projects done in the past and currently working on.
 - Users should be able to see projects with grades.
 - Users should be able to search from the search bar according to year, professor, topics, etc.
 - All of their own projects should only be visible to them.
- Time table
 - Should be able to view all the scheduled events/meetings.
 - Should not be able to create them.
- Progress
 - Users should be able to view the project progress with minutes of meeting.
- Notification
 - When application is accepted/rejected
 - When meeting is scheduled

Professor(user = professor)

- Login
 - Users should be able to login only with daiict email
- Logout
- Profile
 - Users should be able to edit and save the profile information.

- Help
 - Users should be able to view information on how to use this site and other helplines.
- Create project
 - Users should be able to create, edit and save project information by clicking on the create project button.
 - Users should be able to view the student profile for the selection process.
 - Users should be able to accept/reject the application.
- Home page
 - Users should be able to search student profiles from the Search bar.
- My projects
 - Users should be able to see the total number of applications for each project.
 - Users should be able to see all projects done in the past and currently working on with grades assigned.
 - Users should be able to search from the search bar according to students, topics, year, etc.
 - All of their own completed projects should only be visible to them
 - Pending projects should be publicly visible
- Grading
 - Professor can give and edit grades by clicking on a particular project's learn more button.
- Time table
 - Should be able to create, view and edit all the scheduled events/meetings.
- Progress
 - Users should be able to track the project progress and minutes of meetings.
- Notification
 - When someone applies
 - When someone withdraws the application.
 - When someone confirms another project. (for SRI, BTP...)

Non - Functional Requirements

The non-functional requirements of the system are:

- The system should remain accessible 24x7.
- The website should be maintainable.
- Only daiict people should be able to login.
- It should be a web application that should work with every web browser on PC and mobiles.
- The system should encrypt the passwords of all its users.
- The system should be able to send notifications to the User.
- The system should be capable of handling many user logins at the same time.

Use Case

UC: Login using Daiict Google account

- Actors: Students, Professors
- Goal Description: Any Actor will be able to log in using their Daiict Google account.
- Pre-condition: Actor must be a Daiict student or Professor, and they have a Daiict google account.
- Post-condition: Actor will be logged in and will land on the home page
- Flow:
 - User Clicks the login button
 - Redirects to the google API
 - Messages Prompt displays the google account which is logged in their system.
 - When a google account is chosen, it checks if it is daiict mail or not. The user is redirected to the dashboard
- Alternate Flow:
 - Authentication fails, then google API closes, and the process is repeated

UC: Logout

- Actors: Students, Professor
- Goal Description: Any Actor will be able to log out using their Daiict Google account.
- Pre-condition: The actor should be login.
- Post-condition: user will be logged out
- Flow:
 - User clicks on the logout button
- Alternate Flow: none

UC: Create a project

- Actors: Professor
- Goal Description: users can create, edit and save a project
- Pre-condition: user should be logged in
- Post-condition: the project should be created and displayed on home page of students
- Flow:
 - User clicks on create project button
 - User types in all information about project like grades and all
 - The project will be publicly visible till finalisation
- Alternate Flow:
 - If the user wants to edit the project info, click on the edit button and then save.

UC: Apply for a project

- Actors: students
- Goal Description: users will be able to apply for projects.
- Pre-condition: user should be logged in

- Post-condition: a student should see requested badge under that project
- Flow:
 - User searches through a project using search bar
 - Users click on the learn more button and apply for projects as individuals and teams by clicking on the apply button and filling out a text field.
 - Requested badge will be shown on the project after applying to any project.
 - After the confirmation, project will be shown on my projects page
- Alternate Flow: none

UC: withdraw from a project

- Actors: students
- Goal Description: users will be able to withdraw from projects.
- Pre-condition: user should have applied to project
- Post-condition: a student should be able to see apply button
- Flow:
 - User searches the project to be withdrawn using filters
 - Users click on the learn more button and then on withdraw button
 - Professors won't be able to see it in their request list anymore
- Alternate Flow:
 - If the request was already accepted, then too click on the withdraw button and apply will be seen again. The Project won't be shown on my projects page anymore.

UC: Accept/decline the request

- Actors: Professor
- Goal Description: professor should be able to accept/decline the request
- Pre-condition: student must have applied or the project
- Post-condition: a student should get the notification about accept/decline
- Flow:
 - User clicks on accept/decline button
 - After the confirmation, project will not be shown on students home pages anymore
 - After the confirmation, project will be shown on the students my projects page
- Alternate Flow: none

UC: Edit Project Progress

- Actors: Professor
- Goal Description: Any Actor will be able to Edit Project Progress
- Pre-condition: The project must have been created.
- Post-condition: user will be able to add the project progress
- Flow:
 - User clicks on the learn more button and edit
- Alternate Flow: none

UC: View Project Progress

- Actors: student
- Goal Description: Any Actor will be able to view Project Progress.
- Pre-condition: The project must have been created and be a member of it.
- Post-condition: user will be able to view the project progress.
- Flow:
 - User clicks on the learn more button and views.
- Alternate Flow: none

UC: Create meetings

- Actors: Professor
- Goal Description: user will be able to schedule, edit and view meetings
- Pre-condition: user should be logged in
- Post-condition: students will get notifications
- Flow:
 - User navigates to timetable button in header
 - User schedules meeting
- Alternate Flow: none

UC: View meetings

- Actors: Students
- Goal Description: user will be able to view meetings
- Pre-condition: The actor should be login.
- Post-condition: user will be logged out
- Flow:
 - User clicks on the logout button
- Alternate Flow: none

UC: Profile editing

- Actors: Students, Professor
- Goal Description: Actors can edit their profiles.
- Pre-condition: Actors have a profile.
- Post-condition:
- Flow:
 - User navigates to profile page
 - User clicks on edit button
 - Users edit info and then save it
- Alternate Flow:
 - If user wants to edit the profile info again, then click on edit button and then save