73:Graduation projects system

Supervisor: Walaa Alkady.

Teamid:73

|  |  |  |
| --- | --- | --- |
| Name | Id | dep |
| فتحي ناصر فتحي | 20201700575 | sc |
| مازن عبداللطيف لطفي عبد اللطيف | 20201700642 | sc |
| احمد ياسر محسن صالح | 20201700095 | sc |
| محمد امير السيد على | 20201700672 | sc |
| محمد احمد الحسين | 20201701165 | is |
| محمد ابراهيم عبدالحميد محمد | 20201700651 | sc |
| محمد محمود فوزي خليل | 20201700736 | sc |

Introduction

The system can help students to track their progress, collaborate with their peers and advisors, and submit their project work in a timely and organized manner. It can also provide a centralized database for project topics and related resources; the system can help faculty and administration to monitor and evaluate the progress of students and provide necessary guidance and feedback. Overall, a graduation project system can streamline the process of managing and completing final year projects, leading to better outcomes for both students and the institution.

User requirements

The system should allow the user to collaborate between students and advisors, including communication, feedback.

help student to get the idea of his project.

upload the latest versions of project phases so the supervisor can see all updates of the project.

The system should enable students and supervisors to manage the progress of the projects, including scheduling meetings, setting deadlines, and tracking deliverables.

The system should allow supervisors to evaluate and grade the completed projects according to predefined criteria, and provide feedback to students on their performance.

The system should generate reports and analytics on project progress, including student participation, time spent on tasks, and overall project outcomes.

Functional requirements

1. 1: creating account.

• Description/Action: The system must allow users to create an account, login, and logout securely.

• Requirements/Inputs: User must have faculty email, id, username, and password.

• Source: User.

• Pre-condition: The user must have a valid email address and a unique username.

• Post-condition: Successful authentication and verification mail.

• Output: User profile page and system dashboard.

1. Login/out:

* Description/ Action: The system should allow users (students, supervisors, and administrators) to log in to their account and access the  system.
* Requirements/ Inputs: Username and password for the user's account
* Source: user
* Pre-condition: The user must have registered with the graduation projects system and have an account.
* Post-condition: The user will be able to access the features of the graduation projects system based on their role (student, supervisor).
* Output: Confirmation of successful login. Error message if the username or password is incorrect.

1. Project registration:

• Description/Action: The system must allow users to register projects.

• Requirements/Inputs: Project title, description, department, team members.

• Source: User.

• Pre-condition: User must be logged in and authorized to register projects.

• Post-condition: The project is created and saved in the system's database.

• Output: A confirmation message indicating that the project was successfully created and project details, including title, description, and objectives

1. Select advisor:

* Description/ Action: The system should allow students to select a supervisor for their graduation projects.
* Requirements/ Inputs: project information
* Pre-condition: the user must have registered with graduation project and have an account.
* post-condition: The student will be matched with a suitable supervisor for their project. The project will be added to the supervisor's workload.
* Output: The name of the selected supervisor. Confirmation of the supervisor's acceptance of the project.

1. Upload finished phase.

* Description/ Action: The system should allow students to upload the finished phase of their graduation project to their supervisor for review.
* Source: user
* Requirements/ Inputs: Finished phase document.
* Pre-condition: The user must have registered with the graduation projects system and have an account. The student must have selected a supervisor for their graduation project .The student must have completed the necessary work for the finished phase of their project.
* Post-condition: The supervisor will be able to access and review the finished phase document.
* Output: Confirmation of successful upload.

1. Project Feedback.

* Description/Action: The system must allow supervisor to provide feedback on the project.
* Requirements/Inputs: Feedback content, project ID.
* Source: supervisor.
* Pre-condition: Project must exist, supervisor must be authorized to provide feedback.
* Post-condition: The feedback is saved in the system's database and alert the student.
* Output: A confirmation message indicating that the feedback was successfully submitted and sent to the student.

Nonfunctional requirements

1. Performance: The system should be responsive and able to handle many users simultaneously. The system should also have a fast response time to ensure that users can access the system quickly.
2. Scalability: The system should be scalable and able to handle an increasing number of users and projects as the university grows.
3. Reliability: The system should be always reliable and available to users. The system should have a high uptime percentage to ensure that users can access the system whenever they need to.
4. Security: The system should have a high level of security to protect user data from unauthorized access, modification, or theft. This should include measures such as encryption, access control, and regular backups.
5. Usability: The system should be user-friendly and easy to navigate. The interface should be intuitive and easy to understand, even for non-technical users.
6. Accessibility: The system should be accessible to users with disabilities, such as those who are visually impaired or have limited mobility. The system should comply with accessibility standards, such as WCAG 2.0.
7. Maintainability: The system should be easy to maintain and upgrade. The system should be designed to be modular and have well-documented code to facilitate future modifications or enhancements.
8. Compatibility: The system should be compatible with a wide range of devices and web browsers to ensure that users can access the system from their preferred devices.



