

Functional Requirements

1) User Authentication

- a) The user be able to register a new account or log in using an exist one.
 - ✚ [User registration functionality \(backend\)](#)
 - ✚ [User login functionality \(backend\)](#)
 - ✚ [Email verification \(backend\)](#)
- b) The user should be able to reset their password.
 - ✚ [Password reset functionality \(backend\)](#)

2) Profile Management

- a) Users should be able to view and modify their profile information.
 - ✚ [Endpoints for reading and modifying user profile \(backend\)](#)

3) Subject Menu

- a) The subject menu should contain a list of subjects in the class.
 - ✚ [Endpoint for retrieving the menu \(backend\)](#)
- b) The user should be able to click one subject of the list to view its details(marks or homework).
 - ✚ [Endpoint for getting subject details \(backend\)](#)

4) Teacher homework menu

- a) The teacher should be able to add and modify homework.
 - ✚ [Endpoints for managing the menu \(CRUD\) \(backend\)](#)
- b) The student should be able to view homework menu.
 - ✚ [Endpoint for retrieving the menu \(backend\)](#)
- c) The student should be able to click one homework of the list to send its solution.

5) Solution homework menu

- a) The student should be able to add and modify solution.
 - ✚ [Endpoints for managing the menu \(CRUD\) \(backend\)](#)
- b) The teacher should be able to view homework menu.
 - ✚ [Endpoint for retrieving the menu \(backend\)](#)

Non-Functional Requirements

- ✚ In the event of system errors or failures, users should be provided with appropriate error messages and notifications.
- ✚ The code should be well-structured and modular, following best practices for maintainability and future enhancements.