

**METROPOLITAN TIRANA UNIVERSITY**

**COMPUTER SCIENCES AND IT FACULTY**

**SOFTWARE ENGINEERING**

GUIDELINER

**Subject**: “Software Project Management”

**Topic**: “Student’s Dean Website”

**Worked by:** Megi Ismaili, Laura Cekaj, Juliana Kojashi, Maria Bebi, Erisa Nasufi

**Table of Contents**

1. **EXECUTIVE SUMMARY**

1.1 Project Overview

1.2 Purpose and scope of this specification

2. **PRODUCT/SERVICE DESCRIPTION**

2.1 Product Context

2.2 User Characteristics

2.3 Assumptions

2.4 Constraints

2.5 Dependencies

3. **REQUIREMENTS**

3.1 Functional Requirements

3.2 Non-Functional Requirements

3.2.1 User Interface Requirements

3.2.2 Usability

3.2.3 Performance

3.2.4 Manageability/Mainteinability

3.2.5 Security

3.2.6 Standards Compliance

3.2.7 Portability

4. **USER SCENARIOS/USE CASES**

5. **CODE**

1. **Executive Summary**

**1.1 Project Overview**

Our software project aims to provide a comprehensive solution for students to voice their concerns and submit complaints related to various aspects of their academic experience. This platform offers a unique feature that allows students to track the status of their complaints. Instead of waiting for endless days to receive an email response or update, students can log into the system and easily view the progress of their submitted complaints. This feature promotes transparency and keeps students informed about the resolution process. By implementing this functionality, students can stay updated on the status of their complaints, including whether it is under review, in progress, or resolved.

Additionally, it serves as a platform for the dean of students to post and manage different internship opportunities for the students to apply. Students can access the internships posted by the dean and submit their applications through the software.

**1.2 Purpose and Scope of this Specification**

Specification: Student Complaint and Internship Management System

Purpose:

The purpose of this specification is to define the requirements and functionality of the Student Complaint and Internship Management System. The software aims to provide a comprehensive solution for students to submit complaints directly to the dean of students/administrator and to facilitate the management of internship opportunities for the students. This specification serves as a guideline for the development team to understand the project's objectives and deliver a successful software solution.

Scope:

The scope of this specification encompasses the following aspects: 1. Student Complaint Management:

a. Facilities Complaints: Students can submit complaints regarding school facilities and suggest improvements.

b. Review of Grade: Students can voice concerns or disputes related to the grading process and assessment criteria.

c. Attendance Issues: Students can report attendance-related problems and discrepancies in attendance records.

d. School System Inclusion: Students can address matters concerning the inclusion and support of diverse student populations.

e. Didactic Materials: Students can request specific educational resources or improvements to existing materials.

f. Documentation Requests: Students can seek assistance with obtaining necessary documents from the school administration.

g. Scholarship Applications: Students can inquire about or apply for scholarships, financial aid, or grants.

h. Ethical Commission Reporting: Students can report ethical concerns or violations within the academic environment.

2. Internship Management:

a. Internship Listings: The dean of students can post and manage various internship opportunities for students.

b.Internship Details: The listings include information about the company/organization, positions, requirements, and compensation.

c. Application Process: Students can review the internships and submit applications

d.Internship quote

The software will provide an intuitive user interface for both students and the dean of students whos the administrator to interact with the system efficiently. It will ensure secure data handling, user authentication, and appropriate access controls. The scope of this specification does not include other academic processes or administrative functionalities unrelated to student complaints and internships.

1. **Product/Service Description**

**2.1 Product Context**

The Student Complaint and Internship Management System is a software product designed to operate within the context of an educational institution.

Educational Institution:

The system operates within the specific educational institution, such as a university, college, or school. It aligns with the institution's policies, guidelines, and organizational structure.

Students:

The primary users of the system are the students enrolled in the educational institution. They access the system to submit complaints related to different aspects of their academic experience and to explore and apply for internship opportunities.

Dean of Students/Administrator:

The dean of students or an appointed administrator serves as the main authority overseeing student affairs. They act as the system administrator, responsible for managing complaints, reviewing and addressing student concerns, and posting internship opportunities.

Faculty and Staff:

Faculty members and administrative staff may interact with the system indirectly, providing support in resolving student complaints or promoting internship opportunities for the students.

External Entities:

The system may integrate with external entities and systems to enhance functionality. For example:

Companies and Organizations(Internship Providers): The system may connect with external entities offering internships, allowing the dean of students to post internship listings and receive applications from students.

Email and Notification Systems: The system may integrate with email services or notification systems to send updates, notifications, and reminders to students and the dean of students/administrator.

Data Management:

The system operates within the context of data management policies and practices of the educational institution. It securely stores and processes sensitive student data.

Data Sources:

The system may integrate with various data sources, such as student databases, scheduling systems, and internship databases, to retrieve and update relevant information. These sources contribute to the accuracy and effectiveness of the complaint and internship management processes.

**2.2 User characteristics**

A total of 2 users will interact with the system:

1. Students:

- Technical Proficiency: Students should have basic computer literacy skills to navigate and interact with the system.

- Academic Information: Students should have their personal information, such as name, surname, faculty, and study branch, readily available for registration purposes.

- Complaint Submission: Students should be able to write and submit complaints by selecting the appropriate category and providing a detailed message.

- Profile Update: Students should have the ability to update their personal information, such as contact details or academic information, as necessary.

2. Dean of Students/Administrator:

- System Access: The dean of students/administrator should have login credentials to access the system's administrative functions.

- User Registration: The dean of students/administrator should be able to register students by entering their name, surname, faculty, and other relevant details.

- Access Management: The dean of students/administrator should have the ability to view and manage the list of registered students who have access to the system.

- Internship Management: The dean of students/administrator should be able to add internship opportunities by providing details such as title, description, requirements, deadline, and available quota.

- Complaint Management: The dean of students/administrator should be able to view student complaints, including the student's name, surname, faculty, study branch, and the content of the complaint. They should also have the ability to delete inappropriate or irrelevant complaints.

- Internship Application Management: The dean of students/administrator should have access to view all internship applications submitted by students, including their personal details and application information.

**2.3 Assumptions:**

1. Technical Infrastructure: It is assumed that the necessary technical infrastructure, such as servers, databases, and network connectivity, is available to support the Student Complaint and Internship Management System.

2. User Availability: It is assumed that students, the dean of students/administrator, and internship providers will have regular access to the system and will actively engage with its functionalities.

3. Authentication and Authorization: It is assumed that the system will have mechanisms in place to authenticate and authorize users, ensuring that only authorized individuals can access and perform specific actions within the system.

4. Data Accuracy: It is assumed that the information provided by students during registration, complaint submission, and internship applications is accurate and valid.

**2.4 Constraints:**

1. Budget: The development and implementation of the Student Complaint and Internship Management System may be subject to budgetary constraints. The project should be planned and executed within the allocated budget.

2. Time Constraints: There may be time constraints in delivering the system, requiring efficient project management and timely development and deployment.

3. Compliance: The system should comply with relevant legal and regulatory requirements, including data privacy regulations, security standards, and any institutional policies.

4. Technical Compatibility: The system should be compatible with the existing technical infrastructure and software environment of the educational institution, ensuring seamless integration and interoperability.

**2.5 Dependencies:**

1. Data Sources: The system may depend on existing data sources, such as student databases, scheduling systems, and internship databases, to retrieve and update relevant information. Integration with these data sources may be necessary for the system's functionality.

2. External Services: The system may depend on external services, such as email notifications or SMS gateways, for communication purposes. Proper integration and availability of these services may be essential for seamless operation.

3. User Feedback: The system may rely on feedback from users, including students and the dean of students/administrator, to enhance its functionality and address any issues or improvements needed. Regular communication and feedback loops should be established.

4. IT Support: The availability of IT support and technical expertise may be crucial for system maintenance, troubleshooting, and resolving any technical issues that may arise during the system's operation.

1. **Requirements**
   1. **Functional Requirements**

**Requirement: Student Login (FR-01)**

**Description:** Students should be able to log in to the platform using their unique credentials.

**Objective:** Provide secure access for students to interact with the platform and access personalized features.

**Inputs:**

* Username
* Password

**Process/Actions:**

1. Students enter their username and password on the login page.
2. The system verifies the entered credentials against the stored user database.
3. If the credentials are valid, the system grants access to the student’s personalized dashboard.

**Outputs/Results:**

* Successful login message.
* Access to the student’s personalized dashboard.

**Exceptions and Error Handling:**

* If the entered credentials are incorrect, display an error message indicating invalid login.
* Provide a password reset functionality for students who forget their password.

**Acceptance Criteria:**

* Students should be able to successfully log in with valid credentials.
* Invalid credentials should result in an error message and prevent login.

**Requirement: Submit Complaints and Requests (FR-02)**

**Description:** Students should be able to submit complaints and requests through the platform.

**Objective:** Enable students to communicate their issues and requirements effectively.

**Inputs:**

* Complaint/Request details
* Supporting documents (optional)

**Process/Actions:**

1. Students navigate to the "Submit Complaint/Request" section of their dashboard.
2. They enter the details of their complaint or request, including a description, category, and any additional relevant information.
3. Students may choose to upload supporting documents if applicable.
4. They submit the complaint or request.

**Outputs/Results:**

* Confirmation message indicating successful submission.
* The complaint or request is recorded in the system database.

**Exceptions and Error Handling:**

* Validate that all required fields are filled before submission.
* Handle errors in case of submission failures, such as network issues or server errors.

**Acceptance Criteria:**

* Students should be able to submit complaints and requests successfully.
* Required fields should be validated to ensure completeness.

**Requirement: Administrator Account Management (FR-03)**

**Description:** Administrators should be able to add and manage student accounts within the platform.

**Objective:** Provide administrators with the ability to control student account creation and management

**Inputs:**

* Student information (e.g., name, email, student ID)
* Account access privileges (e.g., role, permissions)

**Process/Actions:**

1. Administrators log in to the administrative dashboard.
2. They navigate to the "Manage Student Accounts" section.
3. Administrators can add new student accounts by entering the required student information.
4. They can edit existing student account details if needed.
5. Administrators have the ability to deactivate or delete student accounts if necessary.

**Outputs/Results:**

* Successful addition, update, or deletion of student accounts.
* Updated student account information in the system.

**Exceptions and Error Handling:**

* Validate that all required fields are filled when adding or updating student accounts.
* Handle errors if account creation, update, or deletion fails due to database issues or other errors.

**Acceptance Criteria:**

* Administrators should be able to add, edit, and delete student accounts.
* Required fields should be validated to ensure completeness.

**Requirement: Internship Listing Management (FR-04)**

**Description:** Administrators should be able to add and manage available internships within the platform.

**Objective:** Enable administrators to maintain and update the list of internships for student applications.

Inputs:

* Internship details (e.g., title, description, location, duration, requirements

**Process/Actions:**

1. Administrators log in to the administrative dashboard.
2. They navigate to the "Manage Internships" section.
3. Administrators can add new internships by entering the necessary details.
4. They can edit existing internship details, such as updating the description or requirements.
5. Administrators have the ability to remove internships that are no longer available.

**Outputs/Results:**

* Successful addition, update, or deletion of internship listings.
* Updated list of available internships in the system.

**Exceptions and Error Handling:**

* Validate that all required fields are filled when adding or updating internship listings.
* Handle errors if internship listing creation, update, or deletion fails due to database issues or other errors.

**Acceptance Criteria:**

* Administrators should be able to add, edit, and delete internship listings.
* Required fields should be validated to ensure completeness.

**Requirement: Notification System (FR-05)**

**Description:** The platform should provide a notification mechanism to inform students and administrators about important updates and events.

**Objective:** Facilitate effective communication and keep users informed about relevant activities and changes within the platform

**Inputs:** N/A (System-generated notifications)

**Process/Actions:**

1. When a student submits a complaint or request, the system generates a notification to acknowledge the submission.
2. When there is a status update on a student's complaint or request (e.g., assigned to an administrator, in progress, resolved), the system sends a notification to the student.
3. Administrators receive notifications when new complaints or requests are submitted.
4. Administrators also receive notifications for updates on complaints or requests they are handling.
5. Notifications can be delivered through email, in-platform notifications, or both, based on user preferences.

**Outputs/Results:**

* Students receive notifications regarding the acknowledgment and status updates of their complaints and requests.
* Administrators receive notifications for new submissions and updates on complaints and requests they are handling.

**Exceptions and Error Handling:**

* Handle errors in delivering notifications, such as email delivery failures or errors with the notification display within the platform.
* Provide a mechanism for users to update their notification preferences or opt-out if desired.

**Acceptance Criteria:**

* Students should receive timely and accurate notifications for their complaints and requests.
* Administrators should receive notifications for new submissions and updates on complaints and requests.
  1. **Non-functional Requirements**

**3.2.1 Performance:**

1. The platform shall have a maximum page load time of 3 seconds for all user interactions.
2. The system shall support a concurrent user load of 500 students without significant performance degradation.
3. The platform shall provide real-time updates on the status of complaints and requests with a maximum delay of 5 seconds.

**3.2.2 Reliability:**

1. The platform shall have an uptime of at least 99.9% per month, excluding scheduled maintenance periods.
2. The system shall have automated backup mechanisms to perform regular data backups and ensure data integrity.
3. In case of a server failure, the platform shall have a recovery time objective (RTO) of 4 hours to restore service.

**3.2.3 Security:**

1. The platform shall use secure HTTPS protocol for all communications to ensure data confidentiality during transmission.
2. User passwords shall be stored using industry-standard cryptographic algorithms, such as bcrypt, to protect against unauthorized access.
3. The system shall implement role-based access control (RBAC) to restrict access to sensitive functionalities and data based on user roles and permissions.

**3.2.4 Usability:**

1. The platform shall have a user-friendly and intuitive interface with clear instructions and error messages to guide users in submitting complaints and requests.
2. The system shall follow accessibility guidelines (e.g., WCAG 2.1) to ensure that users with disabilities can access and use the platform effectively.
3. The platform shall be responsive and accessible across different devices and screen sizes, including desktops, tablets, and mobile devices.

**3.2.5 Scalability:**

1. The platform shall be designed to handle a potential increase in the number of users, complaints, requests, and internships without significant degradation in performance.
2. The system shall scale horizontally by adding more servers to distribute the load as the user base grows.

**3.2.6 Compatibility:**

1. The platform shall be compatible with the latest versions of popular web browsers such as Chrome, Firefox, Safari, and Edge.
2. The system shall be responsive and functional on different operating systems (Windows, macOS, Linux) commonly used by students and administrators.

**3.2.7 Maintainability:**

1. The codebase shall follow coding best practices, including clear documentation and standardized naming conventions, to facilitate future maintenance and updates.
2. The system shall be modularly designed to allow for easy modification or extension of individual components without impacting the overall functionality.
3. **User Scenarios/Use cases**

**Scenario 1:** Student Complaint Submission

**Start State:** The student is logged in to the complaint management system.

**Scenario Flow:**

* The student navigates to the complaint submission page.
* The student selects the type of complaint from the available options (Facilities, Review of Grade, Attendance, School System Inclusion, Didactic Materials, Documentation Request, Scholarship, Ethical Commission).
* The student provides details about the complaint, such as the specific issue and any supporting evidence.
* The system verifies the complaint details and sends a confirmation email to the student.
* The dean of students/admin receives the complaint and is notified about the new submission.
* The dean of students/admin reviews the complaint, investigates the issue if necessary, and takes appropriate action.
* The system updates the complaint status to "Under Review" or "Investigation in Progress."
* The student can log in to the system and view the complaint status at any time.
* The dean of students/admin resolves the complaint and updates the status accordingly.
* The system sends a notification to the student informing them of the resolution.
* The student can view the complaint status as "Resolved" or "Closed."

**End State:** The student successfully submits a complaint, and the dean of students/admin manages and resolves the complaint, with the student being able to view the complaint status.

**Scenario 2:** Internship Posting by the Dean of Students

**Start State:** The dean of students/admin is logged in to the internship management system.

**Scenario Flow**:

* The dean of students/admin navigates to the internship posting page.
* The dean of students/admin provides details about the internship, such as the job description, required qualifications, and application deadline.
* The system verifies the internship details and saves the posting in the system.
* The internship is now visible to all students who log in to the system.
* Students can view the internship details and choose to apply if interested.
* Students submit their application materials (resume, cover letter, etc.) through the system.
* The dean of students/admin receives the applications and reviews them.
* The dean of students/admin selects suitable candidates for the internship.
* The system notifies the selected students about their acceptance and provides further instructions.
* The dean of students/admin updates the internship status as "Filled" or "Closed" on the system.

**End State:** The dean of students/admin successfully posts internships, receives applications, selects candidates, and updates the internship status, while students can view and apply for the internships through the system.

**Scenario 3:** Admin Checks for Complaints

**Start State:** The admin is logged in to the complaint management system.

**Scenario Flow:**

* The admin navigates to the admin dashboard.
* The system displays a list of all pending complaints.
* The admin selects a complaint from the list to review.
* The system presents the details of the selected complaint, including the student's information, complaint type, and description.
* The admin investigates the complaint further if necessary, by contacting relevant parties or reviewing additional evidence.
* The admin takes appropriate action based on the investigation, such as resolving the complaint, forwarding it to the relevant department, or requesting more information from the student.
* The admin updates the complaint status and records any actions taken.
* The system sends a notification to the student, informing them about the status update if necessary.
* The admin can continue reviewing and resolving other pending complaints.

**End State:** The admin reviews and manages complaints, takes necessary actions, and updates the complaint status accordingly.

**Scenario 4:** Admin Adds Students

**Start State**: The admin is logged in to the student management system.

**Scenario Flow:**

* The admin navigates to the student management section.
* The admin selects the option to add a new student.
* The system presents a form for the admin to enter the student's details, such as name, email address, student ID, and any other required information.
* The admin fills in the necessary details for the new student.
* The system validates the entered information and checks for any potential duplicates or errors.
* If the information is valid, the system adds the student to the database and assigns a unique identifier.
* The admin receives a confirmation message indicating that the student has been successfully added.
* The student can now log in to the system using the provided credentials and access the available features and functionalities.

**End State:** The admin successfully adds a new student to the system, allowing the student to log in and use the platform's features.

**Scenario 5**: Applying for an Internship

**Start State**: The student is logged in to the internship management system.

**Scenario Flow:**

* The student navigates to the "Internship Opportunities" section.
* The system displays a list of available internships.
* The student selects a specific internship of interest.
* The system presents the internship details, including the job description, required qualifications, and application deadline.
* The student decides to apply for the internship and clicks on the "Apply" button.
* The system prompts the student to upload their application materials, such as a resume and a cover letter.
* The student uploads the required documents and submits the application.
* The system confirms the successful submission of the application and provides a confirmation message.

**End State**: The student successfully applies for an internship by submitting the required application materials.

**Scenario 6:** Adding Internship Opportunities

**Start State**: The admin is logged in to the internship management system.

**Scenario Flow:**

* The admin navigates to the "Add Internship" section.
* The system presents a form for the admin to enter the internship details, such as the job description, required qualifications, and application deadline.
* The admin fills in the necessary information for the internship opportunity.
* The system validates the entered information and checks for any errors or missing details.
* If the information is valid, the system adds the internship to the available opportunities.
* The internship is now visible to students who log in to the system and browse the internship section.
* The admin can continue adding other internship opportunities.

**End State**: The admin successfully adds an internship opportunity, making it visible to students who can then apply for the internship through the system.

**Scenario 7**: Managing Student Accounts

**Start State:** The admin is logged in to the student management system.

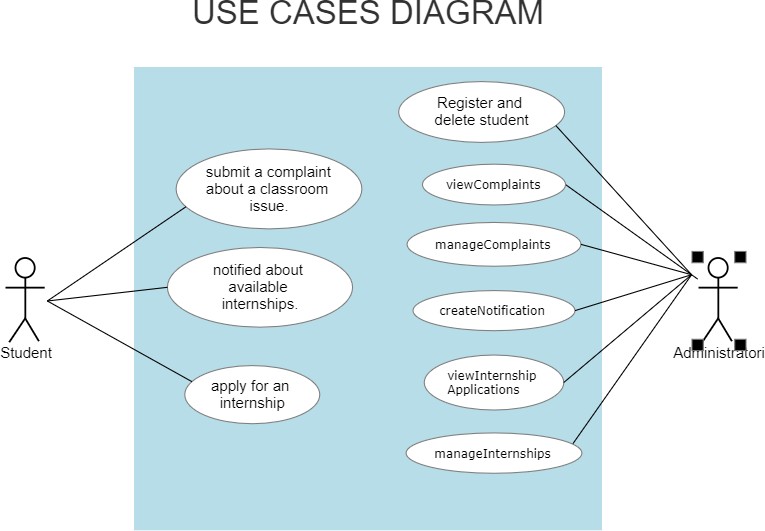
**Scenario Flow:**

* The admin accesses the admin dashboard and selects the "Student Accounts" section.
* The system displays a list of registered students.
* The admin selects a student account to manage.
* The system presents the student's details, including their name, email address, and contact information.
* The admin can update the student's information, such as correcting any errors or modifying contact details.
* The admin can also deactivate or suspend a student account if necessary.
* The system confirms any changes made to the student account and updates the database accordingly.
* The admin can continue managing other student accounts as needed.

**End State:** The admin successfully manages student accounts by updating information and making necessary changes, ensuring accurate and up-to-date student records in the system.

**Classes:**

1. Complaint: This class represents a complaint submitted by a student, including details such as the category, description, and date submitted.
2. Internship: This class represents an available internship, including details such as the title, description, requirements, and application deadline.
3. Student: This class represents a student using the website, including details such as their name, email, and student ID.
4. School Admin: This class represents a school administrator using the website, including details such as their name, email, and admin ID.
5. Notification: This class represents a notification for an available internship, including details such as the title, description, and application deadline.



**Actions:**

1. Store complaint: When a student submits a complaint, the website stores it in the appropriate table in the database.

2. Retrieve complaints: When school admins view the list of complaints, the website retrieves the data from the database and displays it on the appropriate page.

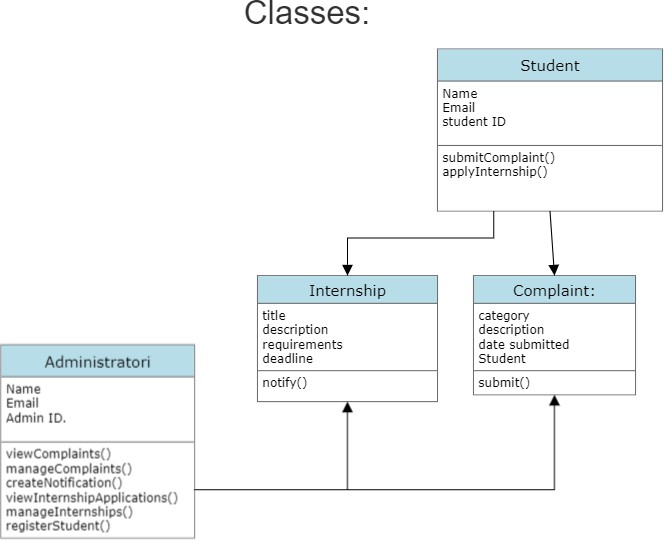
3. Mark complaint as resolved: School admins can mark a complaint as resolved, updating the status in the database.

4. Create notification: School admins can create a notification for an available internship, specifying the details and application deadline.

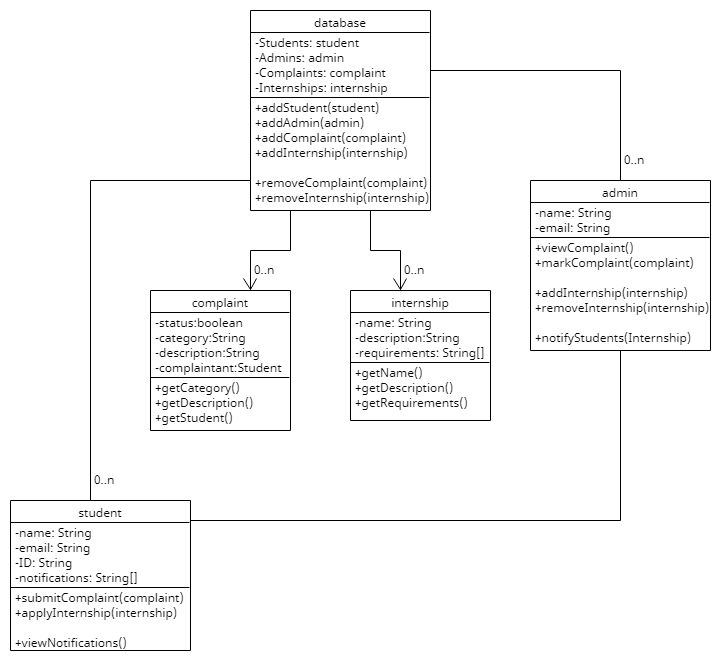
5. Store internship application: When a student applies for an internship, the website stores the application details in the appropriate table in the database.

6. Retrieve internship applications: When school admins view the list of internship applications, the website retrieves the data from the database and displays it on the appropriate page.

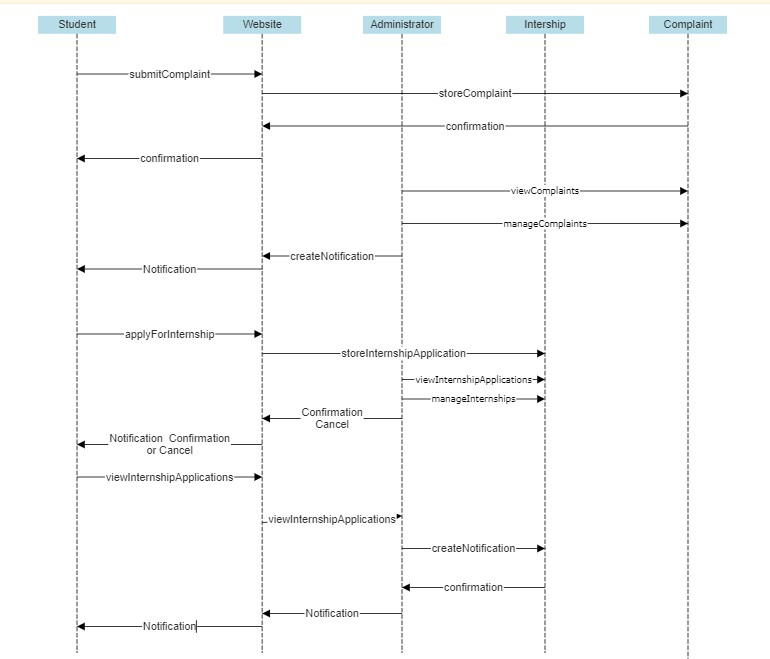
7. Approve or reject internship application: School admins can approve or reject an internship application, updating the status in the database.



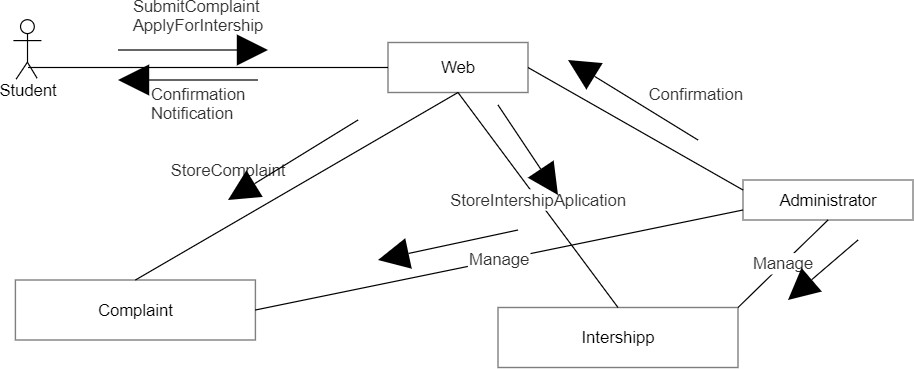
**UML Diagram**



**Sequencence Diagram**



**Collaboration diagram**



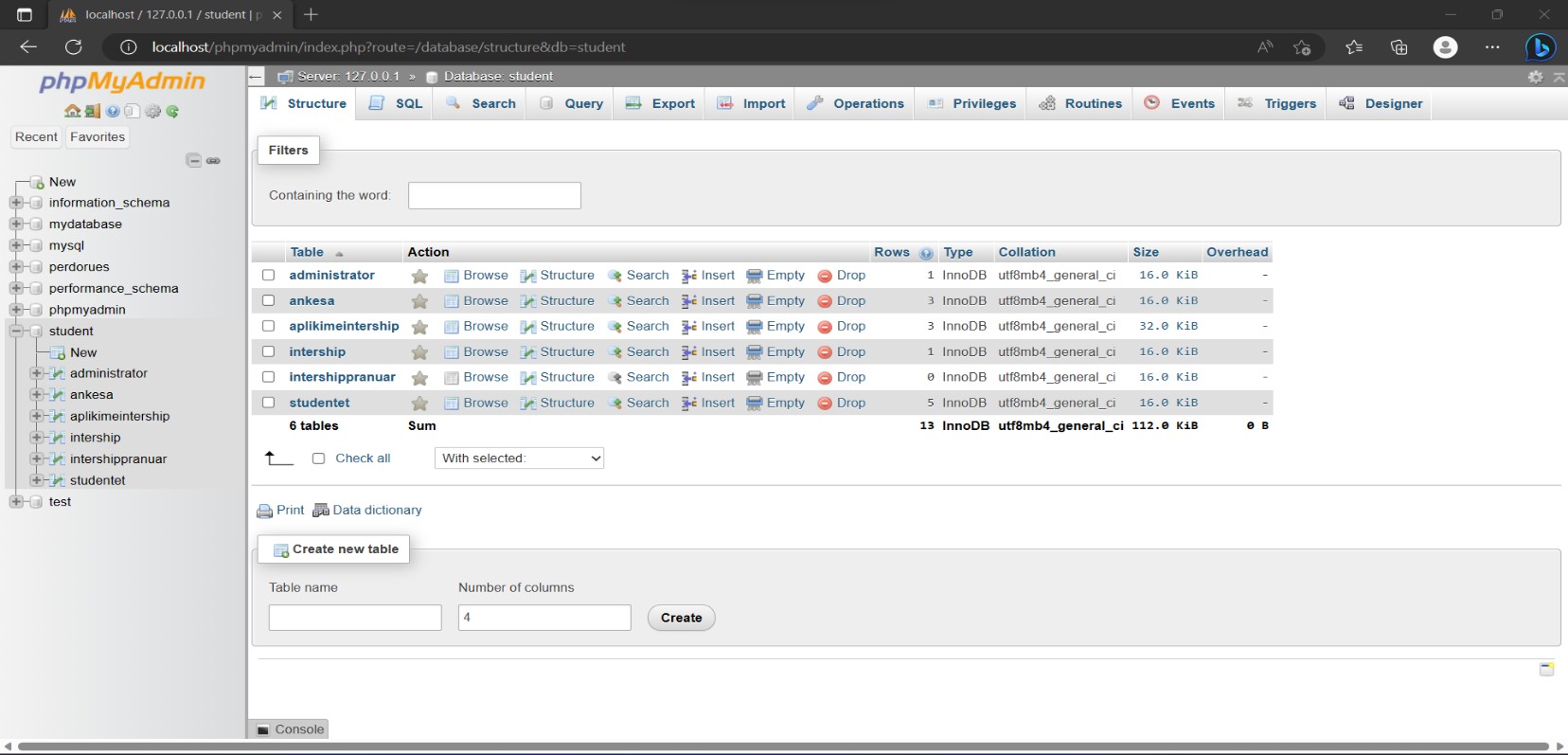
**5.**  **Code**

Below you will find some images taken from web and database. Also, we will do an explanation of all methods used and coded.

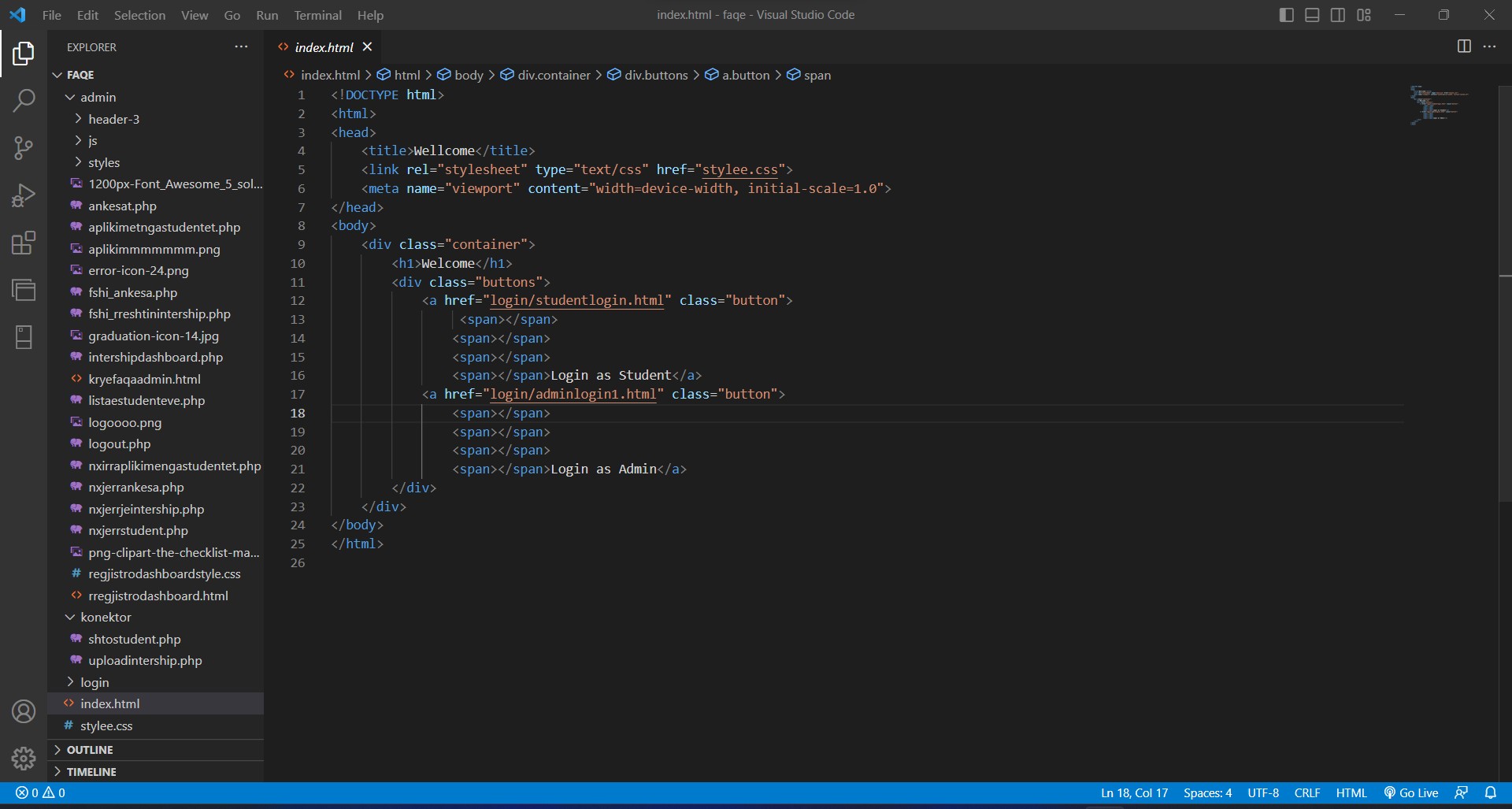
In the beggining, let's create the database where the data has to be put using XAMPP.



After Apache and MySQL are activated, we are able to create our database.



Database has been created under 'Student' name and includes tables with their belonging data. Some of these tables are: 'administrator', 'ankesa', 'aplikimeintership', 'intershippranuar' and 'student'. After creating the database, we are ready to start coding.



We will provide you with all classes created and explain each one of them.

index.html Main web page which contains two login options, for admin and students.

style.css To style index.html page.

adminlogin.html Admin login interface.

adminlogin.php Allows you to check the input data with the database data and then if the data is coorect, you can be able to pass on the ohter page.

studentlogin.html Student login interface.

studentlogin.php Allows you to check the input data with the database data and then if the data is coorect, you can be able to pass on the ohter page.

student.php Student’s action interface which contains a lot of data.

student.css To style student’s web page.

tedhenaperdoruesi.php Takes input from database and shows them in the correct table.

studentupdateinfo.php Updates unfilled student's data.

studentbenaplikim.php Makes an internship application and sends it to the database.

nxjerrintershipstudent.php Shows data from database related to active internships.

logout.php Allows you to logout.

kryefaqeadmin.html Main admin work interface and includes all other pages.

ankesa.php Main complaints interface which allows complaints to be processed.

nxjerrankesa.php Allows taking complaints of users from database.

fshi\_ankesa.php Allows you to delete student's complaints.

intershipdashboard.php--- Internship's interface which allows us to process internships.

uploadintership.php--- Allows us to add a new internship, so it can be accessed from the students.

nxjerrintership.php---Shows all active internships.

fshi\_rreshtintership.php---Allows you to delete an internship.

listastudenteve.php---Interface shows student’s list.

nxjerrstudent.php---Allows you you to take student’s list from database.

rregjistrodashboard.html---Registering student’s interface . rregjistrodashboardstyle.css---Page styling.

shtostudent.php---Adds students in database.

aplikimetngastudentet.php---Interface in which students may apply for internships.

nxirraplikimnengastudentet.php---Allows taking data from database.

//style---Includes styling of header and all other pages.

//js--- Includes JS functions in order to provide responsivness from other devices(phone, tablet ect).

Attached in this guideliner you will find all web files and also the database.

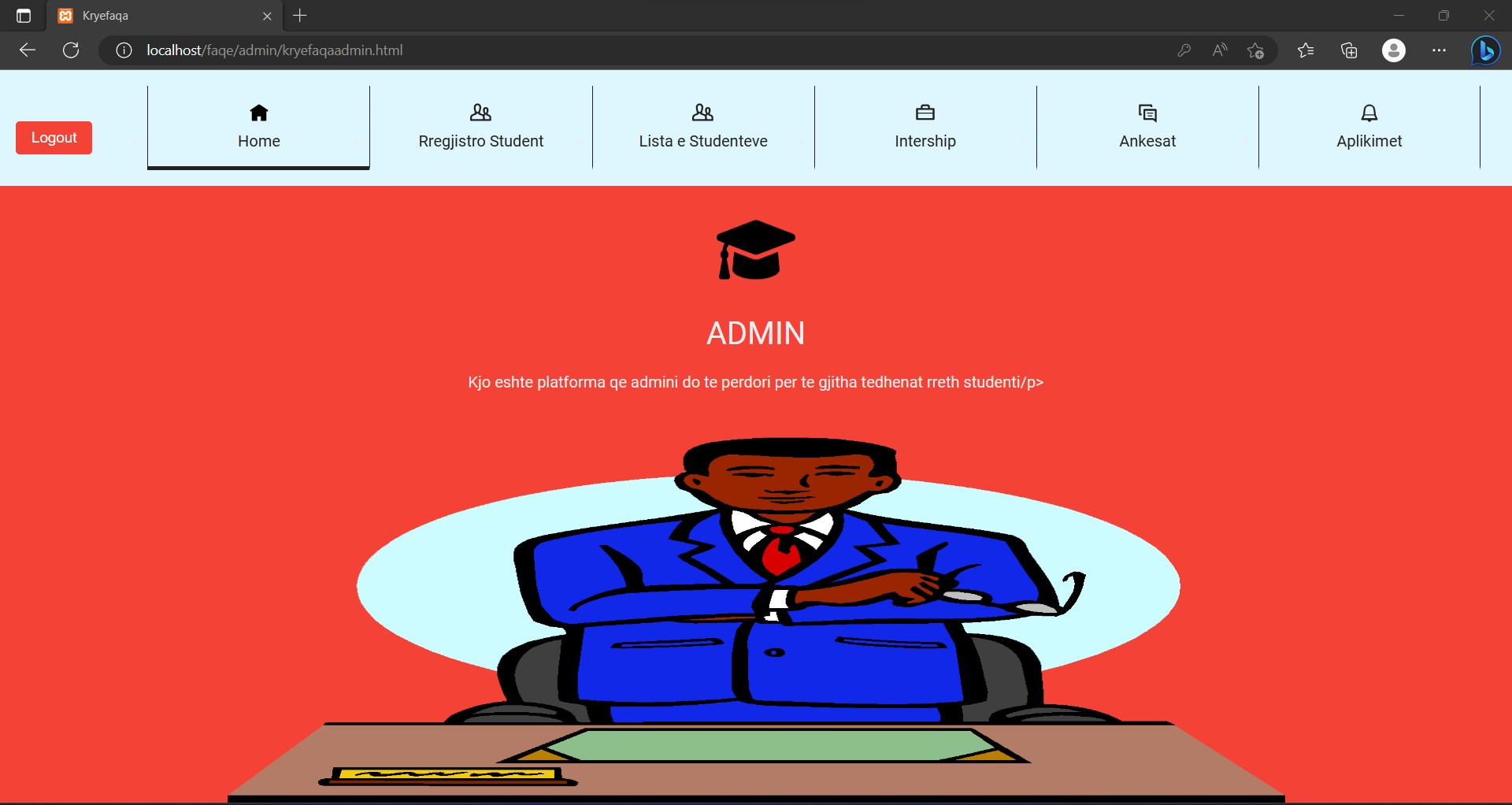
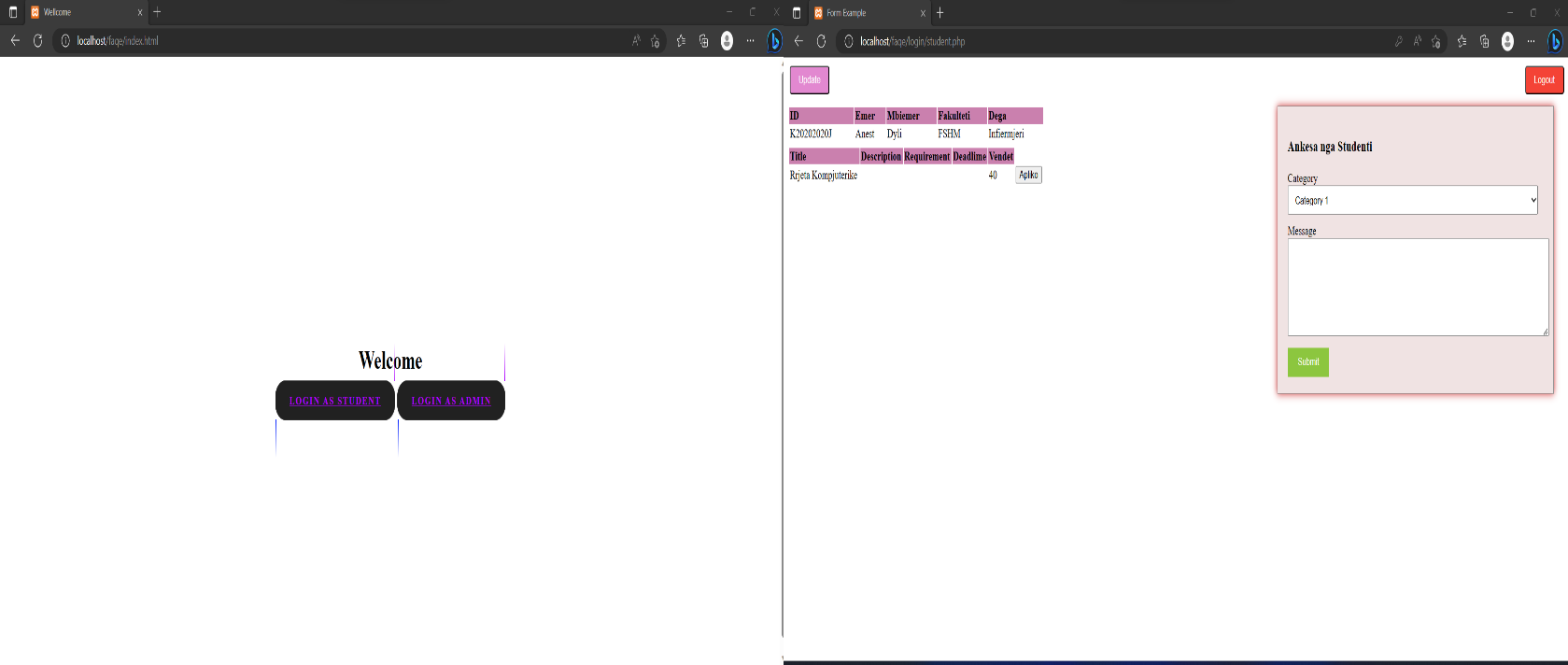
These are the steps to make this folder work.

Step 1: Extract folder.

Step 2: Copy-paste the folder in htdocs folder found in LocalDisk C:.

Step 3: Activate XAMPP, go to MySQL and upload database.

Step 4: Open a browser and search for 'localhost/faqe/'.



End of document.