**Chapter 3**

**Chapter Overview:**

This chapter delves into the crucial aspects of project management and allocation within the context of a Django web application. The primary focus is on creating an efficient system for registering students and supervisors, allocating projects based on skills and interests, and managing project-related information. This chapter also provides a comprehensive overview of the system analysis, design, and control aspects related to project management and allocation. Readers will gain insights into creating an effective and user-friendly system for managing projects within an educational setting.

**SYSTEM ANALYSIS:**

This Project is to enhance the project allocation and management process within an educational institution. This system facilitates student and supervisor registration, fair project allocation, and efficient tracking of project details, milestones, topics, and lectures. It lays the foundation for the subsequent chapters on system design, input design, output design, and system control. The comprehensive understanding of user requirements and system objectives will guide the development process to meet the educational institution's needs.

**ANALYSIS OF THE EXISTING SYSTEM:**

The analysis of the existing project allocation and management system within the educational institution reveals a set of functionalities and processes that form the foundation of the current workflow. This examination serves as a crucial precursor to the system enhancement project, providing insights into the strengths, weaknesses, and areas of improvement. The existing system comprises several key components, including student and supervisor registration, project allocation mechanisms, and tracking functionalities for project details, milestones, topics, and lectures.

1. Student and Supervisor Registration:

The current system incorporates a registration process for both students and supervisors. Students are able to enroll in the project allocation system, while supervisors can register to participate in guiding and overseeing student projects. The analysis will delve into the user experience of this registration process, identifying any bottlenecks or inefficiencies that may exist.

2. Fair Project Allocation:

Project allocation is a critical aspect of the system, and the analysis will evaluate the fairness and efficiency of the current allocation mechanisms. This involves an examination of the criteria used for allocation, the transparency of the process, and the tools employed to match students with suitable supervisors and projects.

3. Tracking of Project Details and Milestones:

Efficient project tracking is essential for both students and supervisors. The analysis will scrutinize the current system's capability to track project details, monitor progress milestones, and manage timelines effectively. This includes an assessment of the tools and interfaces provided for updating project status, communicating between students and supervisors, and addressing any issues that may arise during the project lifecycle.

4. Topic and Lecture Management:

The existing system likely includes features for managing project topics and associated lectures. The analysis will explore how topics are proposed, approved, and integrated into the project allocation process. Additionally, it will assess the tools available for scheduling and conducting lectures related to project management and development.

5. System Objectives and User Requirements:

A fundamental aspect of the analysis involves a comprehensive understanding of the system's objectives and user requirements. This includes gathering feedback from students, supervisors, and administrators to identify any gaps between the current system and their expectations. The goal is to ensure that the enhanced system aligns closely with the needs and goals of the educational institution.

6. Foundation for System Design:

The findings from this analysis will serve as the foundation for subsequent chapters on system design, input design, output design, and system control. Understanding the current system's intricacies will guide the development process, allowing for targeted enhancements that address specific pain points and improve overall functionality.

In summary, the analysis of the existing project allocation and management system provides a holistic view of its components, functionalities, and performance. This thorough examination is a critical step in the system enhancement project, ensuring that the upcoming design and development phases are informed by a deep understanding of user needs and system objectives within the educational institution.

**PROBLEMS OF THE EXISTING SYSTEM:**

The traditional way of project allocation and research writing is faced with so many deadlocks, which tends to alter the actual flow of the research.

1. **Stressful Activities:** It is very tedious to get a topic assigned to a student, due to the rigorous activities involved in choosing a project topic to a student.

**2. More costly:** Students tend to incur more cost to photocopy and printing of project work and correction.

**3. Duplication of topics**: Different supervisors tend to approve the same topic (sometimes with slightly different titles) for different students in the department.

**4. Time difference between student and supervisor**: Most times, submitting the project proposal or visiting the supervisor becomes hectic due to time schedule or personal issues.

**5. Slow system:** because the student needs to be physically present with the supervisor to submit every project document the system tends to be very slow, because if any of the individual (supervisor or student) happens to travel, the project would come to a pause, pending when such person arrives.

**ANALYSIS OF THE PROPOSED SYSTEM**

We will be looking at the proposed system “Development of a Web-Based Student Project Allocation and Management System” from two perspectives, the “Project allocation section” and the “Project Writing section”.

Even though they tend to be seemingly two different systems, they are working together to achieve a common goal of digitizing or automating and simplifying the process of student’s project activities for both the supervisors and students.

**ANALYSIS OF PROJECT ALLOCATION**

In this section we will be more precise on the major activities carried out before a student is assigned a supervisor, from the point of the choice of topic selection, area of interest and project supervisor allocation, and also, we will further give some attention to two important modules, the validation and test modules. Students are faced with the challenge of research topic selection when it comes to project research, and it entails that more time is wasted looking for topics to research on. This system provides a platform that enables supervisors and admin to add project topics for students.

Haven allocated a topic after registration, there is a provision for milestones were a supervisor can access the student performance, and also communicate with the student.

**SOFTWARE REQUIREMENT SPECIFICATIONS**

A software requirements specification (SRS) is a description of a software system to be developed, it’s laid out functional and non-functional requirements, and may include some use cases or instances of the descriptive implementation or use of the system, below are some functional and non-functional requirements of the intended system.

**FUNCTIONAL REQUIREMENT**

The functional requirements specify the behaviors or functionalities of this system as well as its actions and reactions to some encoded events, this intended System possesses the following functionalities.

**1. Registration:** both students and supervisors should be able to sign up on the platform and upload the credentials that will be required of them.

**2. Login:** both students, supervisors and admin should be able to login or sign in the system to perform necessary actions.

**3. Dashboard:** the dashboard is a control panel or a space with linking to all activities and settings for a particular user or groups in the systems with easy access, to both admin, supervisors and students individually.

**NON-FUNCTIONAL REQUIREMENT**

Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are often called "quality attributes" of a system. Other terms for non-functional requirements are quality of service requirements, constraints, non-behavioral requirements, or technical requirements. Informally these are sometimes called the "ilities", from attributes like stability and portability.

The following are the nonfunctional requirements of “Student Project Allocation and Management System”

1. Usability: The software should have an easy to use an interactive user interface, as well as a user guide for ease of navigation.

2. Platform constraints: the software should be able to work effectively on any system that can run a good web Browser.

3. Portability: the system can easily be moved to any hosting platform that supports HTML, CSS, JavaScript, and python/django and above.

4. Security: the system is protected from SQL injections (SQLi) and Cross Site Scripting (XSS) to secure users’ data and system policy.

5. Modifiability: the system can easily be modified to suite the users’ requirements.

**SYSTEM DESIGN**

The system design phase involves translating the requirements and analysis into a structured design that outlines the architecture, components, and interactions of the Educational Project Management System.

**1. High-Level Architecture:**

The system will follow a client-server architecture. The server, implemented using Django, will handle the core logic and data storage, while the client-side will be a user-friendly web interface accessible to students, supervisors, and administrators.

**2. Database Design:**

Django's Object-Relational Mapping (ORM) will be utilized to design the database. Key entities include `Student`, `Supervisor`, `Project`, `ProjectAllocation`, `Milestone`, `Topic`, and `Lecture`. Relationships will be established using foreign keys.

**3. User Authentication and Authorization:**

Django's built-in authentication system will be employed for user management. Different user roles (Student, Supervisor, Administrator) will have distinct permissions, ensuring appropriate access levels.

**4. Web-Based Interface:**

**- Dashboard for Students:**

- View allocated projects.

- Submit project proposals.

- Access real-time project updates.

**5. - Dashboard for Supervisors:**

- Approve project proposals.

- Provide feedback to students.

- Monitor project progre**ss.**

**6. Project Allocation:**

- Supervisor manually allocates project topics for students.

- Real-time updates on project allocation status.

**7. Project Management:**

**-** Milestones, topics, and lectures managed within the system.

- Real-time collaboration between students and supervisors for reviews and corrections.

- Version control to track changes.

**8. Centralized Approval Dashboard:**

- Project coordinator has a centralized dashboard for real-time monitoring.

- Automated notifications for pending approvals.

**LOGICAL DESIGN**

The logical design deals with the abstracted activities that goes on in the system, these actions are not visible to the users, nevertheless they are the basic drivers of the system.

**Header redirection:** this check the login criteria for each user to determine which users is accessing the system, for students, they are redirected to the students’ panel, for admin, they are redirected to the admin panel, and for staffs they are redirected to staff’s panel.

**Host servers:** the host servers validate a stream of activities in the various panels linked to the directory server, which in turns makes sure to give privileges to each user according to their profiles. Directory server: this is a gateway to access the information on the database, it is the link that takes in information, sends it to the database and also retrieves such information to the required user in need of it.

**Database:** here the database doesn’t only play the role of saving information it also aids the total validation of all the processes.

User Request



*Diagram of logical design.*

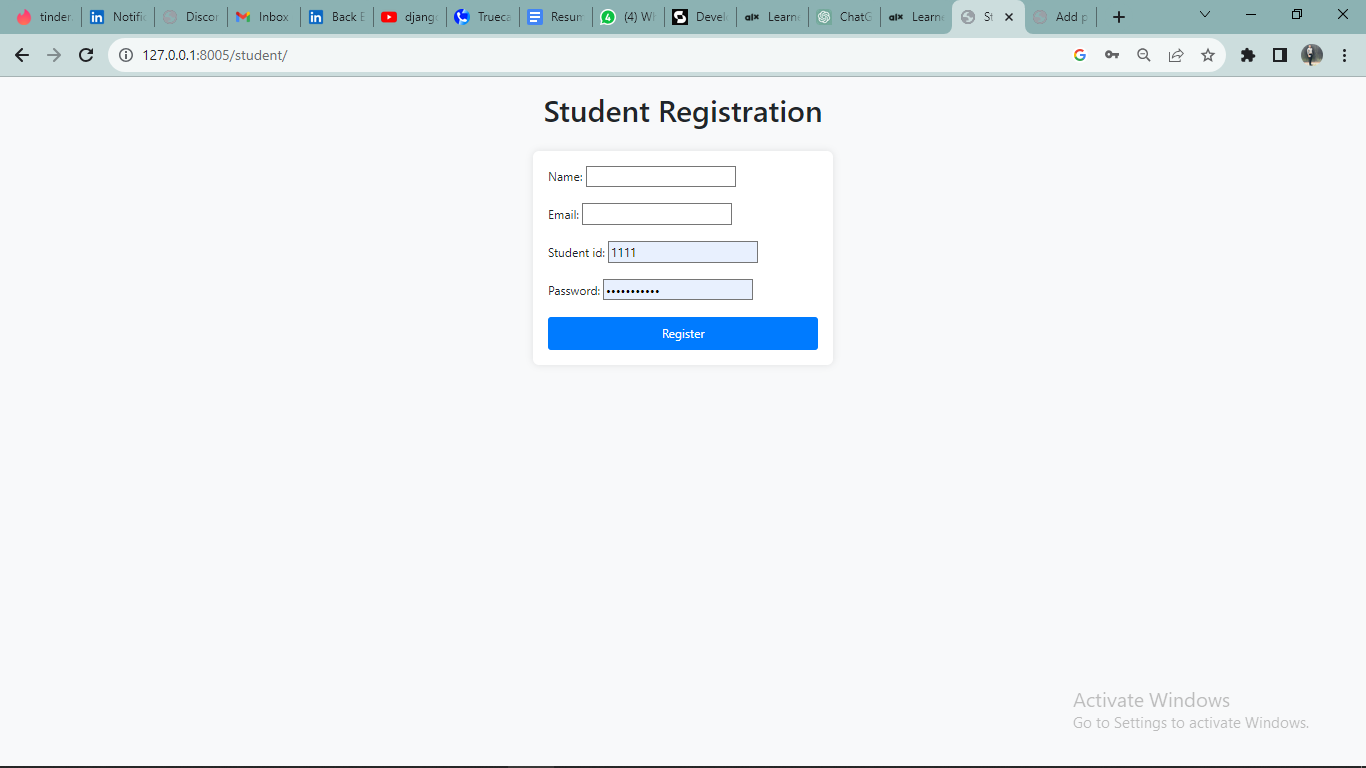
**INPUT DESIGN**

The input entails all forms of data entry in this system, which would be more understood from the below diagram, detailing the front end the Processing/validation and the database. The front end comprises HTML, CSS, and JavaScript, which is used to design and validate the input forms, while the process and validation takes in all inputs from the front-end using PYTHON/DJANGO, then validates and transfers the data to the database using SQLITE instructions.



*Abstract diagram for general input design of SPAMS*

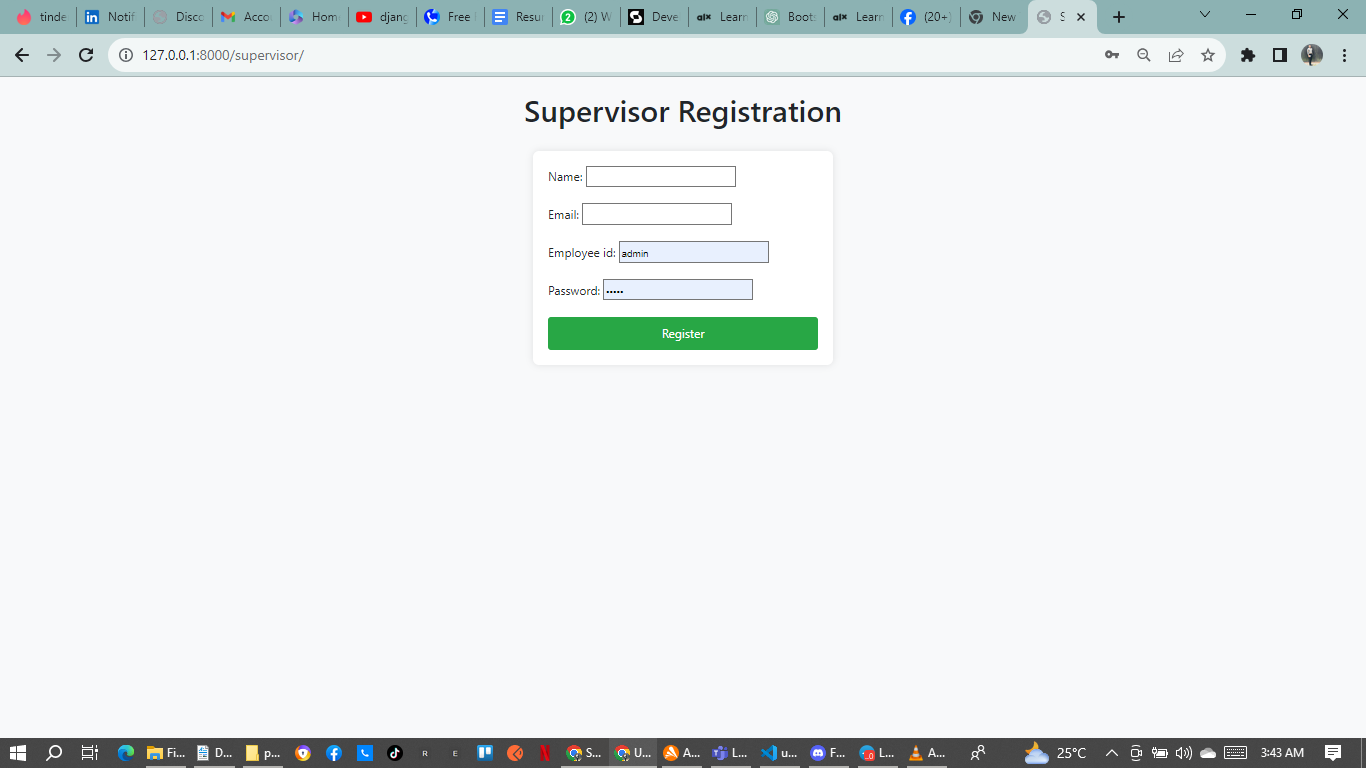
STUDENT REGISTRATION FORM



*Input design for student registration*

The above form (student registration form) captures the intended project students, then validates, after registration each student would then have a unique ID to enable the system to allocate the right project supervisor for me specialized in such topics.

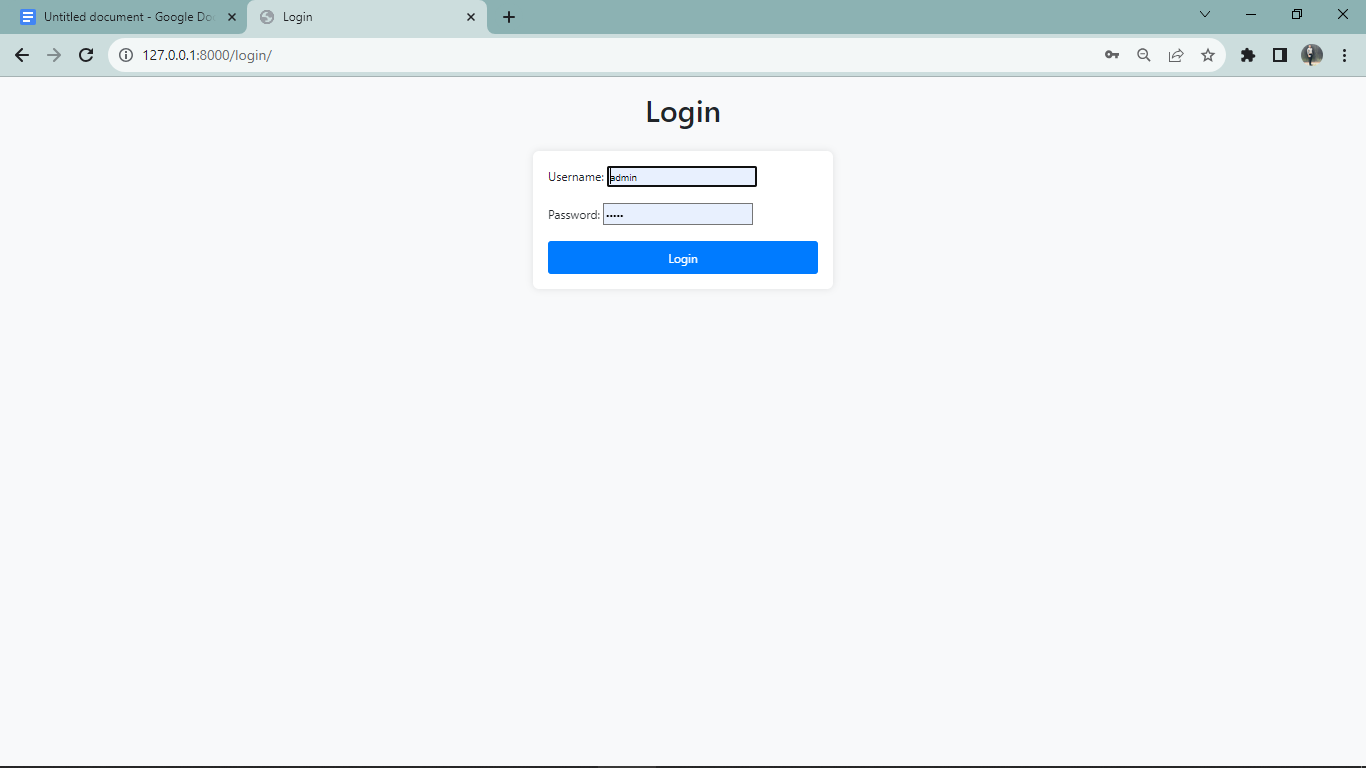
SUPERVISORS REGISTRATION FORM



*Input design for supervisor*

The supervisor (lectures) registration form is designed to collect staff's as well as validating each staff member to make sure he/she is truly a staff member of the institution.

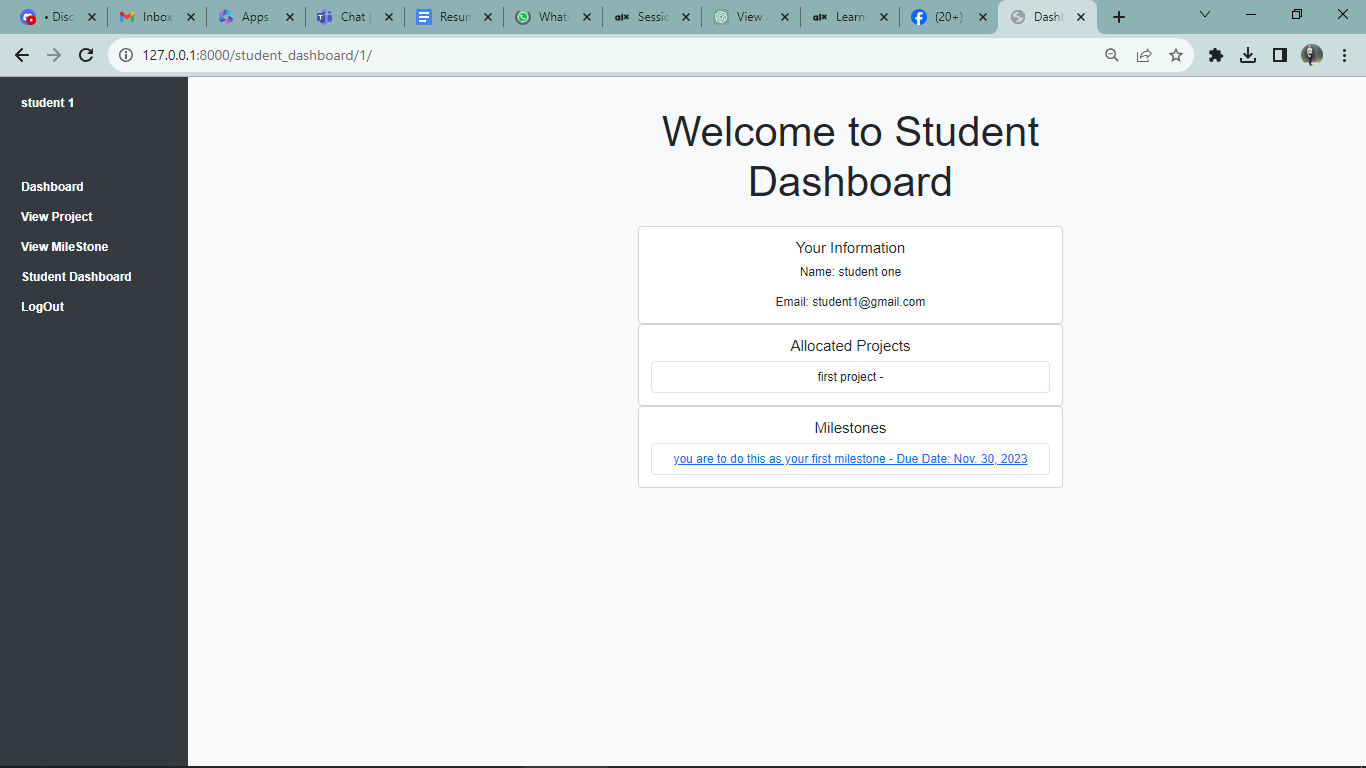
LOGIN PAGE



*Login design for both student and supervisor*

Students and Lecturers are allowed to have access to their various dashboard buy providing their authentication credentials.

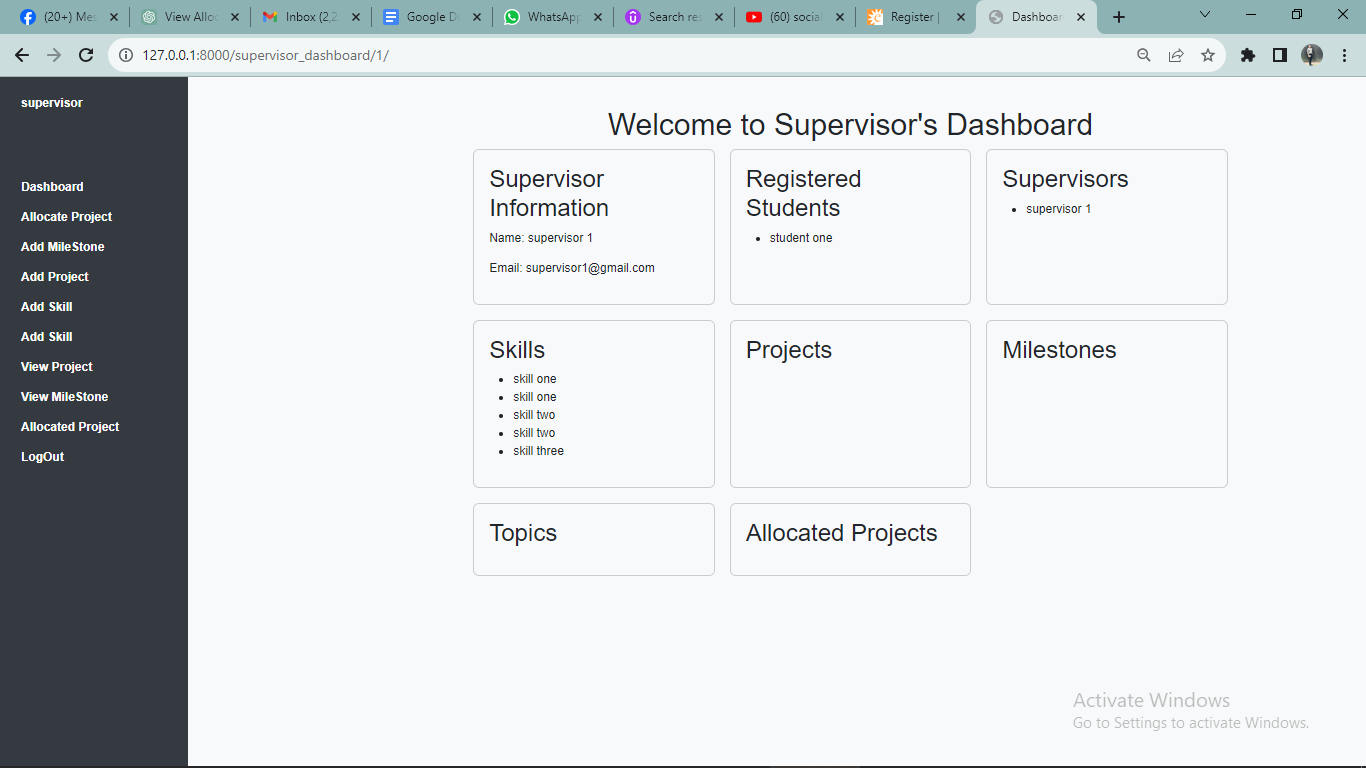
STUDENT DASHBOARD



*Image showing student dashboard*

Once the student user is authorized he or she is granted access to the student dashboard where he can view the project topic that is allocated to him or her, and also view the milestone associated with the project.

SUPERVISORS DASHBOARD



*Diagram showing supervisors dashboard*

USE CASE DIAGRAM FOR STUDENT USER



*Diagram showing what the student can do in the dashboard*

USE CASE DIAGRAM FOR SUPERVISOR USER



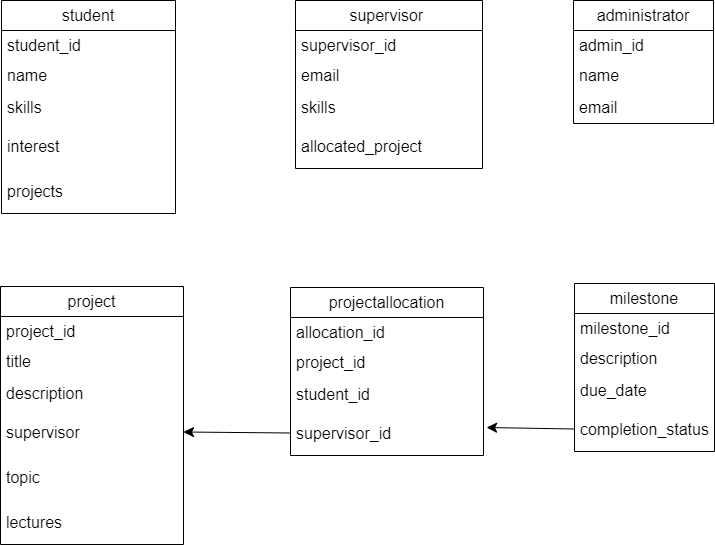
Diagram showing what the supervisor can do in the dashboard after authorization

ACTIVITY DIAGRAM



*Diagram above shows the basic activity a student and supervisor can do.*

CLASS DIAGRAM



*Diagram showing database table and relationship.*

Each box represents a class, and the attributes of the class are listed within the box.

The - symbol indicates private attributes.

The <>----- symbol represents a one-to-many relationship.

The classes include Student, Supervisor, Administrator, Project, ProjectAllocation, and Milestone.

The relationships depict how entities are connected (e.g., a student can have multiple projects, a project has one supervisor).

Chapter 4

CHAPTER OVERVIEW

This chapter generally deals with implementation discussions and choice of language and its features, we will also discuss about the system testing methods, the target computer system requirements which includes the software and hardware requirements of these systems, finally we would round it up by some clues on the maintenance method and some sample outputs of the working system as we discuss.

4.2 FEATURES AND CHOICE OF IMPLEMENTATION LANGUAGE

Considering the fact that almost all the functionalities of this system have to be done electronically and remotely, it is imperative to use web-based languages for the implementation of this system, hence we have to select a language not just for its simplicity and flexibility alone but also a language with much library, supports and built-in functions to handle must of the functionalities we require, below are the selected languages, area of implementations and features.

4.2.1 FEATURES OF HTML

HTML is an acronym for Hypertext Markup Language, it is the basic element for most webpages, hence it can be referred to as the skeletal structure of a website, this web language was used to hold different elements of the website (software system) like the headers, contents, navigation bars, and web elements like tags (images, forms and documents) on this system as well as the footers, some basic features of HTML are:

1. Web Workers: Certain web applications use heavy scripts to perform functions. Web

Workers uses separate background threads for processing and it does not affect the

performance of a web page.

2. Video: You can embed video without third-party proprietary plug-ins or codec. Video

becomes as easy as embedding an image.

3. Canvas: This feature allows a web developer to render graphics on the fly. As with video, there is no need for a plug in.

4. Application caches: Web pages will start storing more and more information locally on

the visitor's computer. It works like cookies, but where cookies are small, the new feature allows for much larger files. Google Gears is an excellent example of this in action.

5. Geolocation: Best known for use on mobile devices, geolocation is coming with HTML5, with enables a website to pi-point a specific location on Google map.

6. Platform independent language

7. HTML is not case and space sensitive.

4.2.2 FEATURES OF CSS

CSS stands for Cascading Style Sheets, more commonly known as CSS, it has fast become the preferred web design method for the benefits it offers web designers and website visitors alike. CSS is a language used to detail the presentation of a web page's markup language (most commonly HTML or XHTML) – such as colors, fonts, and layout. One of its key benefits is the way it allows the separation of document content (written in HTML or a similar markup language) from document presentation (written in CSS). CSS was employed in this system to design and give a good user interface for the software, below are some key features of CSS:

1. Browser Compatibility

CSS stylesheets increase your website's adaptability and ensure that more visitors will be able to view your website in the way you intended no matter the web browser they are using to view it.

2. Device Compatibility (Viewing Options)

CSS enables all device to view a website in its own unique way by the means of

responsiveness (adjusting to all screen sizes and resolution), which means the way a mobile phone can view a website is different from the way a computer system and a tablet would view that same website.

3. Consistency

By making one change to your website's CSS style sheet, you can automatically make it to every page of your website. The bigger your website, the more time CSS saves you. And not only does CSS save time, it also ensures that your web pages have consistent styling throughout your site.

4. Bandwidth Reduction

When CSS separates your website's content from its design language, you dramatically

reduce your file transfer size. Your CSS document will be stored externally, and will be

accessed only once when a visitor requests your website.

5. CSS Animations and Transitions

This is the feature of CSS that enable developers to create animated images and objects on a website using CSS for more description on a site.

6. Web fonts

CSS avails users the abilities to use an online font to design and give their web documents unique looks.

￼7. Colour

CSS has the ability to give diverse colors to different elements on a webpage.

4.2.3. FEATURES OF JAVASCRIPT

JavaScript is a client-side scripting language. This means that JavaScript codes are written into an HTML page, when a user requests an HTML page with JavaScript in it, the script is sent to the browser to perform some specific tasks. JavaScript is used mainly to dynamically update web pages and enhancing the interaction of a user

within the webpage, in most cases, JavaScript is used to create responsive, interactive elements for web pages enhancing the user experience.

4.2.4. FEATURES OF DJANGO

Django is a high-level web framework written in Python that encourages rapid development and clean, pragmatic design. It follows the Model-View-Controller (MVC) architectural pattern, but in Django, it's often referred to as the Model-View-Template (MVT) pattern. Here are some key features of Django:

1. Object-Relational Mapping (ORM): Django provides a high-level ORM that abstracts the database layer, allowing developers to interact with databases using Python code instead of SQL queries. This makes database operations more Pythonic and less error-prone.

2. Admin Interface: Django automatically generates an admin interface based on your data models. This feature allows developers and administrators to manage database records through a web interface without having to build a custom admin panel.

3. URL Routing: Django uses a clean and flexible URL routing system. You can define URL patterns in a central location, making it easy to manage and understand the structure of your web application.

4. Template Engine: Django comes with its own template engine that allows you to define the structure of your HTML pages dynamically. Templates support template inheritance, making it easy to reuse code and create a consistent look and feel across your application.

5. Middleware: Django middleware provides a way to process requests globally before they reach the view or after the view has processed the request. This can be used for various purposes, such as authentication, security, or modifying the request/response.

6. Forms: Django simplifies form handling and validation. It includes a form handling system that allows developers to define forms in Python and then render them in HTML. The framework also provides server-side validation and error handling.

7. Authentication and Authorization: Django includes a built-in authentication system that handles user authentication and provides features like user registration, login, and password reset. It also includes an authorization system to control access to different parts of your application.

8. Security Features: Django has several built-in security features, such as protection against common web vulnerabilities like Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and SQL injection. It encourages best practices in securing web applications.

9. Testing Framework: Django provides a testing framework for writing and running tests to ensure the correctness of your application. It includes support for unit testing as well as higher-level testing using tools like Selenium.

10. RESTful API Framework: Django Rest Framework (DRF) is a powerful and flexible toolkit for building Web APIs. It is a third-party package that integrates seamlessly with Django, making it easy to build RESTful APIs.

11. Internationalization and Localization: Django supports multiple languages and locales, allowing you to build applications that can be easily translated into different languages.

12. Scalability: While Django is excellent for rapid development, it is designed to scale. It includes features like database connection pooling and caching to help optimize performance in production environments.

.

4.1.1 FEATURES OF SQLITE

4.1.2 L DATABASE MANAGEMENT SYSTEM

SQLite is a lightweight, file-based relational database management system (RDBMS) that is widely used in various applications due to its simplicity, efficiency, and ease of integration. Here are some key features and characteristics of SQLite:

1. Serverless and Self-Contained: Unlike client-server databases (such as MySQL or PostgreSQL), SQLite is serverless and self-contained. It operates as a single, stand-alone database engine, and the entire database is stored in a single file on disk. This makes it easy to distribute and deploy in applications without the need for a separate database server.

2. Zero Configuration: SQLite does not require any setup or configuration. It is often considered "zero-configuration" because you can start using it by simply including the SQLite library in your application and creating a database file.

3. Cross-Platform: SQLite is cross-platform and works on various operating systems, including Windows, macOS, Linux, and mobile platforms like Android and iOS. This makes it a versatile choice for applications targeting different environments.

4. SQL Compatibility: SQLite follows the SQL standard and supports a subset of the SQL language for defining, querying, and manipulating data. Most SQL operations that you perform in other databases can be done in SQLite as well.

5. ACID Compliance: SQLite is ACID-compliant, ensuring that database transactions are Atomic, Consistent, Isolated, and Durable. This guarantees the reliability and integrity of data even in the presence of system failures or crashes.

6. Transactional Database: SQLite supports transactions, allowing multiple SQL statements to be executed as a single atomic operation. This ensures that either all changes are applied or none if an error occurs, maintaining data consistency.

7. Single User at a Time: While SQLite supports multiple connections, it operates with a file-level lock, meaning only one write operation can occur at a time. This makes it well-suited for applications with low to moderate concurrency requirements.

8. Low Resource Footprint: SQLite is designed to be lightweight and has a minimal memory and disk space footprint. This makes it suitable for embedded systems, mobile applications, and scenarios where resources are limited.

9. Embeddable: Due to its self-contained nature, SQLite is easily embeddable in applications. Many programming languages provide libraries or modules for interacting with SQLite, making it accessible for developers working in diverse environments.

10. Compatibility with Many Programming Languages: SQLite has bindings for numerous programming languages, including Python, Java, C#, and many others. This broad support makes it easy to integrate SQLite into applications written in different languages.

11. Community and Documentation: SQLite has a large and active community, and there is extensive documentation available. This includes official documentation, forums, and community-contributed resources, making it easy to find help and guidance when working with SQLite.

While SQLite is not suitable for all use cases, particularly those requiring high concurrency or large-scale enterprise applications, its simplicity and versatility make it an excellent choice for smaller projects, prototyping, and applications with moderate data requirements.

4.3

4.3 SOFTWARE MAINTENANCE ISSUES

Every system has a guide and some set out rules to ascertaining full durability of that system, and also to eliminating sudden breakdown during use of that system, some of such maintenance culture employed for this system are as follow.

4.3.1 CORRECTIVE MAINTENANCE

On encountering any anomalies user should contact the admin with the error massage displayed on the screen, after trying some of the below techniques.

1. If encountering a connection problem kindly check your system connection or I.P address, as you may be encountering a network problem from your connection.

2. If encountering any sort of error from submission of form, first refresh the web page if

error persist you may need to restart your web browsers or possibly clear browser cache memory.

3. If you are experiencing a bad design and misplacement of an element kindly change your browser or update to a newer version.

4. For the Admin, Deploy backup system as the last option of repairs.

4.4.2 ADAPTIVE MAINTENANCE

As every web system requires, it is mandatory to always logout after use for

Security reasons more

especially when using a public device, and all information provided by users should be valid and authentic information.

1. A steady or regular routine check should be conducted after the implementation stage monthly at least, so as to identify if there are lapses and debug it.

2. There should be a regular scanning of all drives using anti-virus to avoid virus into the system.

3. Regular update of the website for current information.

￼

4.5.4 PREVENTIVE MAINTENANCE

1. Users (students, lectures(supervisors) and admin) should be trained on how the system works

2. The system should be installed on a very good server or hosting company if it would be deployed remotely.

3. The system should be online only during project allocation and project writing seasons to avoid misuse or abuse.

4. Adequate database size should be allocated to the system to avoid congestion since the system would be handling multiple data uploads including images and documents.

5. Students and supervisors should be limited to only the underlined institution hosting the system.

6. Multiple registration of staff would be handled by the admin, as the system controls what of students.

7. Backup system should always be set and running at all times to avoid complete system failure.

4.6 RESULT

The system output is mainly base on the operations and activities carried out by the different users

on the system. After the allocation of supervisors to the project students, students can login to

view, the project that is allocated to them and also the details and milestone attached to a particular project. The administrator can also view information about the supervisors, the students and their topics.

Appendix of this project.

PROJECT APPROVAL FORM