جـامعة بيروت العربية
BEIRUT ARAB UNIVERSITY

"StudentHub360: Empowering Academic Success through a Unified Online Support Ecosystem"

By

Abdallah Hamdan : 202202370

Ali Iskandar: 202201535

Abdulrahman Enshassi: 202200248

Mohammad Barakeh: 202202962

Department of Electrical and Computer Engineering

Faculty of Engineering

COMP442- Software Engineering

Supervised by:

Dr. Imane Haidar

Spring 2023/2024

Project Overview:

StudentHub360 is a dynamic online platform designed to provide students with comprehensive support and resources, fostering academic success, and personal growth. As a basic prototype, the project will be built as a website platform which will provide various types of services for all students in need. Also, this platform is not only useful for students, but also a powerful tool for the instructors which will make there work more flexible. Our website isn't just like any other traditional- info providing – website, it will be a more advanced platform where it will include a chatbot for interactive experience for the student. The users will be able to ask the chatbot any question instead of heading to their advisors or instructor's office, and the chatbot will be trained to provide answers to all appropriate questions asked.

Objectives:

1. Enhance Old and New Students Support:

- Provide easy and academic support to students through a user-friendly online platform.
- Encourage personal growth by offering resources beyond academic assistance.

2. Facilitate Instructor Flexibility:

- Develop tools that enable instructors to manage and deliver content more flexibly.
- Easy communication between instructors and students.

3. Integrate Chatbot for Quick Assistance:

- Implement a chatbot to address student queries promptly.
- Reduce the need for students to visit physical offices for routine information.

4. Create a Dynamic Learning Environment:

- Establish an interactive platform that adapts to the evolving needs of students and instructors.

Background:

StudentHub360 originated from the firsthand experiences of students facing challenges and inefficiencies during their initial semester at school. The founders observed a repetitive and time-consuming routine of students queuing up in front of advisors' offices, leading to wasted time and frustration. Recognizing the need for a more streamlined and effective support system, the idea for StudentHub360 was conceived.

Literature Review:

1. Chatbot creation:

A chatbot is an artificially created virtual entity that interacts with users using interactive textual or speech skills. This CHATBOT directly chats with the people using artificial intelligence and Machine Learning concepts. Please refer to this paper about Design and Development of chatbot:

Refer to References:[1]

2. Integrating chatbots for college enquiries:

A system was developed to assist university students in their inquiries. The primary goal was to develop a specific architecture, create a model for managing communication, and provide the proper responses to the students. Please refer to this research which focuses on presenting a chatbot system in an educational domain:

Refer to References:[2]

3. Chatbots applications in education: A systematic review

Refer to References:[3]

Applications:

1. Student Services Portal:

- Centralized access to academic resources.
- Submission and tracking of assignments.
- Schedule management and reminders.

2. Instructor Dashboard:

- Content management for courses.
- Communication tools for announcements and discussions.

3. Chatbot Assistance:

- Instant answers to frequently asked questions.
- Guidance on course selection, deadlines, and academic policies.

4. Campus Resources and Services:

- Libraries
- Computer Labs
- health services

Alternative Designs:

1.User Interface Options:

- Explore different UI designs to optimize user experience for both students and instructors.

2. Chatbot Variations:

- Evaluate different chatbot platforms and technologies.
- Experiment with natural language processing for improved chatbot responses.

3. Learning Models:

- Investigate alternative educational models, such as adaptive learning or personalized learning paths.

4. Monolithic Architecture:

The first alternative design option is to implement a monolithic architecture for the website. In this approach, all the functionality (chatbot, GPA calculator, instructors' schedules, etc.) is built as a single, tightly coupled application. The monolithic architecture simplifies development and deployment, as all the components reside within a single codebase. However, it may pose challenges in terms of scalability and maintainability as the application grows and complexity.

Pros:

- Simplicity in development and deployment.
- Easier communication between different components.
- Lower operational overhead.

Cons:

- Lack of flexibility and scalability.
- Potential difficulties in managing and maintaining a large codebase.
- Limited fault isolation if one component fails.

5. Microservices Architecture:

The second alternative design option is to adopt a microservices architecture. In this approach, the website is divided into separate, loosely-coupled services, each responsible for a specific functionality such as the chatbot, GPA calculator, and instructors' schedules. Each service operates independently and can be developed, deployed, and scaled individually. Communication between services can be achieved through well-defined APIs or message queues.

Pros:

- Scalability and flexibility to independently develop and deploy services.
- Improved fault isolation, as failures in one service do not affect the entire system.
- Ability to use different programming languages and technologies for each service.

Cons:

- Increased complexity in managing multiple services.
- Higher operational overhead due to the distributed nature of the architecture.

Project Planning:

i. Constraints:

a) Implementation Environment of the Current System:

The current system may be built using outdated technologies, and the implementation environment may require upgrading server infrastructure, software versions, or database systems to support the StudentHub360 platform.

b) Partner or Collaborative Applications:

The StudentHub360 platform may need to integrate with popular learning management systems, such as Moodle or Canvas, to synchronize course information, grades, and assignments for seamless student experience and data consistency.

c)Off-the-shelf Software:

The project may leverage existing open source chatbot frameworks to develop the chatbot functionality quickly and efficiently, rather than building the entire NLP engine from scratch.

d)Anticipated Workplace Environment:

The StudentHub360 website should be responsive and optimized for various devices, including desktops, laptops, tablets, and mobile phones, ensuring a consistent user experience across different screen sizes and operating systems.

e) Schedule Constraints:

The project must be completed before the start of the next academic semester to ensure that students and instructors can access and utilize the StudentHub360 platform from day one, aligning with the academic calendar.

f) Budget Constraints:

The project budget allows for the purchase of necessary software licenses, cloud hosting infrastructure, and hiring additional developers or designers as needed while maintaining a cost-effective approach.

ii. Project Issues:

a) Issues that have been raised and do not yet have a conclusion:

- Data Security and Privacy: Ensuring the security and privacy of student data is crucial. It is essential to establish robust data protection measures, comply with relevant regulations and implement secure authentication and encryption mechanisms to safeguard sensitive information.

- Technical Integration Challenges: Integrating the StudentHub360 platform with various partner or collaborative applications may present technical challenges. Ensuring seamless data exchange, compatibility, and smooth integration workflows require careful planning and coordination with the respective application providers.

- Maintenance and Support: Once the StudentHub360 platform is deployed, ongoing maintenance, bug fixing, and user support will be necessary. Establishing a dedicated support mechanism and addressing user feedback promptly will contribute to a positive user experience and long-term success.

b) Migration to the New Product:

The migration process requires transferring student profiles, course data, and historical records from the existing student support system to the StudentHub360 platform while ensuring data integrity and minimal disruption to ongoing operations.

c)Risks:

One potential risk is the Scalability of the Chatbot Feature. If a large number of students start using the chatbot simultaneously, it may cause performance issues or inaccurate responses. Mitigation strategies could involve load testing, optimizing code, and monitoring system performance.

Another potential risk might be Data Breach or Security Vulnerability where there is a risk of unauthorized access or data breach, where sensitive student information or user credentials could be compromised. This could lead to reputational damage, legal consequences, and loss of trust from students, instructors, and other users.

iii. Team Members Tasks:

→**Manager:** Abdallah Hamdan who oversees the entire project, creates the project plan, assigns tasks to team members, coordinates meetings, and ensures that the project stays on track, such as monitoring progress, resolving conflicts, and communicating with team members.

→**Designer:** Abdulrahman Enshassi and Abdallah Hamdan who is responsible for creating wireframes, prototypes, and visual designs for the StudentHub360 platform, considering factors such as user-friendly navigation, intuitive interfaces, and adherence to branding guidelines.

→**Developers:** Ali Iskandar, Mohammad Barakeh and Abdallah Hamdan who are responsible for implementing the backend functionality, frontend components, database integration, and API connections. For example, they would write code to handle user authentication, chatbot logic, and data retrieval from external systems.

iv. Ethical Issues:

Ethical issues include ensuring that student data is handled securely and confidentially, obtaining proper consent for data collection, and designing the chatbot system to avoid biased or discriminatory responses. It would also involve addressing accessibility requirements, such as providing alternative text for visual content and accommodating users with disabilities.

v. Software Model Process:

→ Waterfall Model:

- Requirement Gathering: Gather in-depth requirements for the website.
- Design: Create a website design based on the gathered requirements.
- Development: Develop the website using the chosen template.
- Testing: Thoroughly test the website for functionality and quality assurance.
- Deployment: Deploy the website to a production environment for public access.

→ Incremental model:

- Divide: Divide the project into smaller parts or increments.
- Develop: Begin development of the first increment.
- Test: Test the functionality of the developed increment.
- Feedback: Gather user feedback and incorporate necessary improvements.
- Repeat: Repeat steps 2-4 for the remaining increments.

- Integrate: Integrate all the developed increments.
- Final Testing: Perform final testing to ensure seamless functionality.
- Deployment: Deploy the website for public access.
- → **Reuse model:**
- Requirements Specification: Define and document the specific requirements for the website.
- Component Analysis: Identify existing software components that can be reused to fulfil the requirements.
- Requirements Modification: Modify and refine the requirements as needed to align with the capabilities of the reusable components.
- System Design with Reuse: Design the system architecture and structure, incorporating the identified reusable components.
- Development and Integration: Develop and integrate the website, utilizing the selected reusable components during the implementation process.
- System Validation: Thoroughly test and validate the entire system to ensure it meets the defined requirements and functions correctly.

The choice of software model process for StudentHub360 project depends on various factors, including project requirements, team composition, timeline, and the level of flexibility needed throughout the development process. We chose Incremental development for various reasons which are:

- Early Delivery and User Feedback: The Incremental model allows for delivering functional increments of the StudentHub360 platform at regular intervals. This enables us to get early feedback from users, instructors, and other users, which can help validate the project's direction and identify areas for improvement. It ensures that the project team remains aligned with user needs and expectations throughout the development process.

- Flexibility and Adaptability: The StudentHub360 project may involve evolving requirements, as user needs and educational trends can change over time. The Incremental model provides the flexibility to incorporate changes and enhancements in subsequent increments, allowing the development team to adapt and respond to evolving requirements effectively.

- Risk reduction: Breaking the project into smaller increments reduces the overall project risk. Each increment is developed, tested, and validated independently, which helps identify and address potential issues early on. It allows our team to mitigate risks associated with technology, requirements, and resource constraints at a smaller scale, reducing the impact on the overall project.

Additionally, the Incremental model aligns well with the iterative and feedback-driven nature of software development. It allows for continuous improvement, encourages collaboration, and ensures that the StudentHub360 platform evolves based on user needs and feedback.

vi. Feasibility Study:

Economic Feasibility:

- Development costs (website, chatbot) need to be assessed.
- Potential revenue models (depending on the target audience and long-term goals) include:
    - Freemium (basic features free, premium features with additional resources or functionalities)
    - Subscription model for access to premium content or personalized support services
    - Partnerships with universities/institutions (discounted access for their students)
    - Sponsorships from relevant educational organizations or companies

Operational Feasibility:

- Content creation and maintenance require ongoing effort for resources, student support services, and the chatbot's knowledge base.
- The chatbot needs to be trained and updated regularly to provide accurate and helpful answers to student inquiries.
- Depending on the complexity of support services offered, a team of moderators or support personnel might be necessary to address intricate student needs beyond the chatbot's capabilities.

vii. Tools/Technology:

The project will be implemented using HTML and CSS PHP for server-side scripting, SQL for database management, and JavaScript for specific functionalities, all within the VS Code integrated development environment (IDE).

viii. Standards:

1. Coding Standards: Adhere to industry-standard coding conventions for HTML, CSS, PHP, and Python to ensure clean, readable, and maintainable code.
    - Related Standard: IEEE 1012 (for ensuring code quality) and possibly ISO/IEC 9126 (for measuring code quality metrics).
2. Accessibility Standards: Ensure compliance with accessibility standards to make the StudentHub360 platform accessible to users with disabilities.
    - Related Standard: ISO/IEC 27001 (for managing security risks, which may include accessibility considerations).
3. Security Protocols: Implement secure coding practices, including input validation, data sanitization, and encryption of sensitive data.
    - Related Standard: ISO/IEC 27001 (for managing security risks) and possibly IEEE 1012 (for security testing and validation).
4. Documentation Standards: Document project requirements, architecture, design decisions, APIs, and codebase.
    - Related Standard: IEEE 1471 (for architectural description and documentation), IEEE 830 (for software requirements specification), and ISO/IEC 12207 (for overall software documentation).
5. Maintenance and Support Standards: Establish procedures for bug tracking, issue resolution, and software updates. Implement logging and monitoring mechanisms to facilitate troubleshooting and proactive maintenance. Provide user support channels and documentation for troubleshooting common issues.
    - Related Standard: ISO/IEC 12207 (for maintenance processes) and possibly CMMI (for process improvement, including maintenance and support processes).
6. Testing Standards: Develop and execute comprehensive test plans to verify the functionality, performance, and reliability of the software. This includes unit testing, integration testing, system testing, and acceptance testing.
    - Related Standard: IEEE 1012 (for software verification and validation) and possibly ISO/IEC 12207 (for testing processes).

ix. Milestones:

- Frontend Development Complete: fully functional frontend implementation of the StudentHub360 platform using HTML, CSS.

Basic navigation, layout, and user interaction features are implemented.

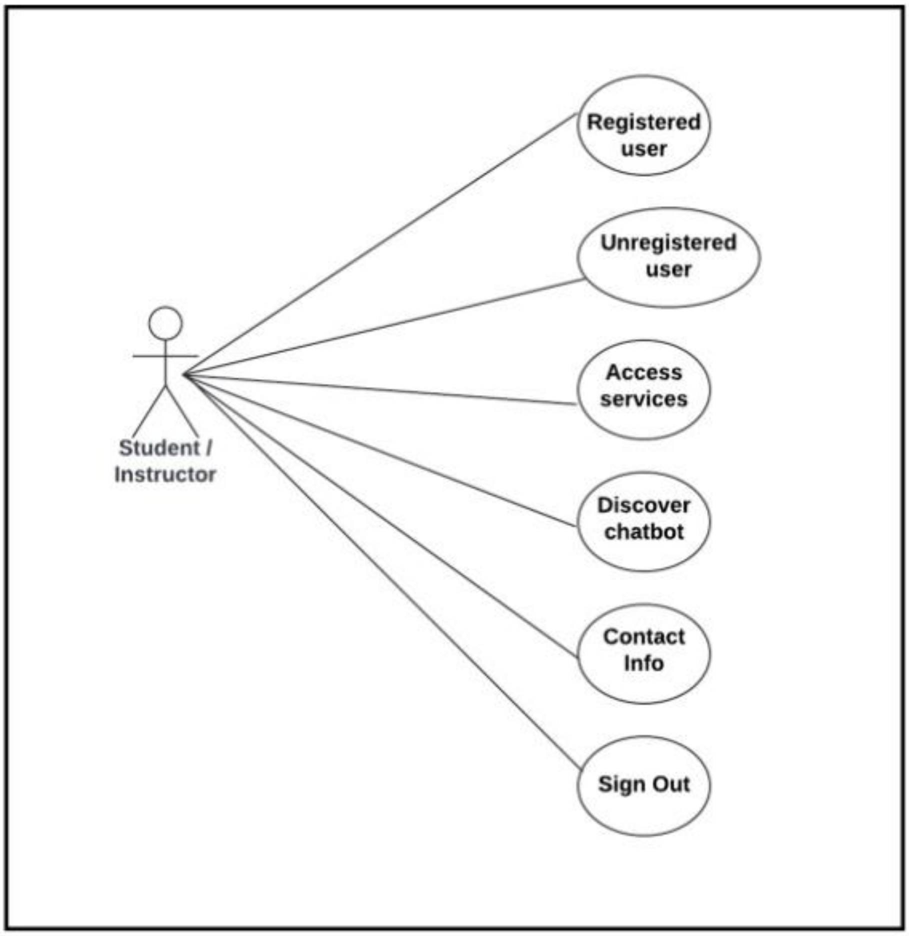
- Backend Development Complete: backend implementation using PHP and JavaScript, including server-side logic, database integration, and API connections.

Core functionalities such as user authentication, data retrieval, and processing are implemented.

- Deployment and Launch: deploy the StudentHub360 platform to a production environment. Also, conduct final testing, ensure scalability and performance, and address any last-minute issues. Finally, officially launch the StudentHub360 platform for users to access and utilize.

Requirements:

i. Use Case:



The provided use case diagram illustrates the key interactions between users and the StudentHub360 platform, focusing particularly on the functionality related to the chatbot feature. The diagram showcases the primary actors, such as students and instructors, and their interactions with the system to achieve specific goals. The central feature highlighted in the diagram is the chatbot, which serves as an interactive tool for students to seek academic support and assistance. By initiating conversations with the chatbot, students can efficiently access resources, resolve queries, and receive guidance without the need for direct human intervention. Additionally, the diagram acknowledges alternative paths and scenarios, emphasizing the platform's flexibility in accommodating various user needs and potential outcomes. Overall, this use case diagram provides a clear visual representation of the core functionalities and user interactions within the StudentHub360 platform, facilitating a comprehensive understanding of its capabilities and benefits.

ii. Functional Requirements

1. User Management:

a. Users (students and instructors) can register for accounts with username, password, and email address.

b. Users can log in and out of the platform securely.

c. The system should implement basic user profile management for storing information like name and institution.

2. Content and Resources:

a. The platform should provide access to curated educational resources categorized by subject or topic (e.g., text documents, video lectures, practice problems).

b. The system should allow for content search using keywords or filters based on subject, resource type, etc.

3. Chatbot Support:

a. Implement a chatbot feature that can answer frequently asked questions (FAQs) related to academic support, resources, and platform functionalities.

b. The chatbot should be able to understand user queries and provide relevant and accurate information.

c. The system should allow for training and updating the chatbot's knowledge base to improve its response accuracy over time.

4. User Interface (UI):

a. Develop a user-friendly and intuitive interface for easy navigation and access to features.

b. The UI should be responsive and function well on various devices (desktops, tablets, smartphones).

c. Ensure clear labelling and organization of content categories and resources.

iii. Non-Functional Requirements:

1) Performance Requirements:

- The platform should load within 3 seconds on average under normal traffic conditions.
- Chatbot responses should be generated within 1 second of receiving the user query.
- The system should be able to handle a peak load of 1000 concurrent users without significant degradation in performance.

2) Dependability Requirements:

- The platform should have a system uptime of at least 99.9% to ensure continuous availability for users.
- Regular backups of user data should be performed to prevent data loss in case of system failures.
- The platform should have automated failover mechanisms to switch to backup servers in case of hardware or software failures.

3) Maintainability and Supportability Requirements:

- The codebase should be well-documented, adhering to coding standards and best practices to facilitate future maintenance by developers.
- The platform should have a ticketing system for users to report issues or request support, with response times not exceeding 24 hours during business days.
- Regular software updates and patches should be released to address bugs, security vulnerabilities, and enhance functionality based on user feedback.

4) Security Requirements:

- User passwords should be securely hashed and stored using industry-standard encryption algorithms.
- The platform should implement SSL/TLS encryption to secure data transmission between users and the server.
- Access controls should be enforced to restrict unauthorized access to sensitive resources and functionalities within the platform.

5)Usability and Humanity Requirements:

- The platform should provide user-friendly error messages and guidance to help users recover from mistakes or misunderstandings.
- The chatbot should use natural language processing techniques to understand colloquial language and provide human-like responses.
- User interfaces should be accessible to individuals with disabilities, complying with WCAG (Web Content Accessibility Guidelines) standards.

6) Look and Feel Requirements:

- The platform should have a modern and visually appealing design, with a clean layout and intuitive navigation.
- Visual elements such as colours, fonts, and icons should be consistent across the platform to maintain a cohesive look and feel.

7) Operational and Environmental Requirements:

- The platform should be hosted on reliable and secure servers with sufficient capacity to handle expected traffic loads.
- Environmental factors such as server room temperature and humidity levels should be monitored and controlled to ensure optimal hardware performance and longevity.

8) Cultural and Political Requirements:

- The platform should respect cultural diversity and avoid content or features that may be offensive or inappropriate in different cultural contexts.
- Content moderation policies should be in place to prevent the dissemination of misinformation or hate speech that could incite political or social unrest.

9) Legal Requirements:

- The platform should comply with data protection laws and regulations applicable in the regions where it operates, such as GDPR, CCPA, or HIPAA (Health Insurance Portability and Accountability Act).
- Terms of Service and Privacy Policy documents should be provided to users, outlining their rights and responsibilities regarding data usage and privacy protection.

iv. Data Requirements:

1) User Data:

- User profiles: Including username, password (hashed), email address, profile picture, role (student/instructor), and any additional optional information such as major, interests, etc.
- User preferences: Settings related to notification preferences, language preference, and accessibility options.
- Chatbot interaction logs: Recording user queries, chatbot responses, timestamps, and session duration for analysis and improvement of chatbot performance.

2) Resource Data:

- Resource metadata: Including title, description, author, publication date, category/tags, and file type (e.g., PDF, video, webpage).
- Resource files: Actual study materials, articles, tutorials, videos, and other educational content hosted on the platform.

3) Feedback and Evaluation Data:

- User feedback: Collecting feedback provided by users on the platform's features, usability, and content quality through surveys, rating systems, or direct input forms.

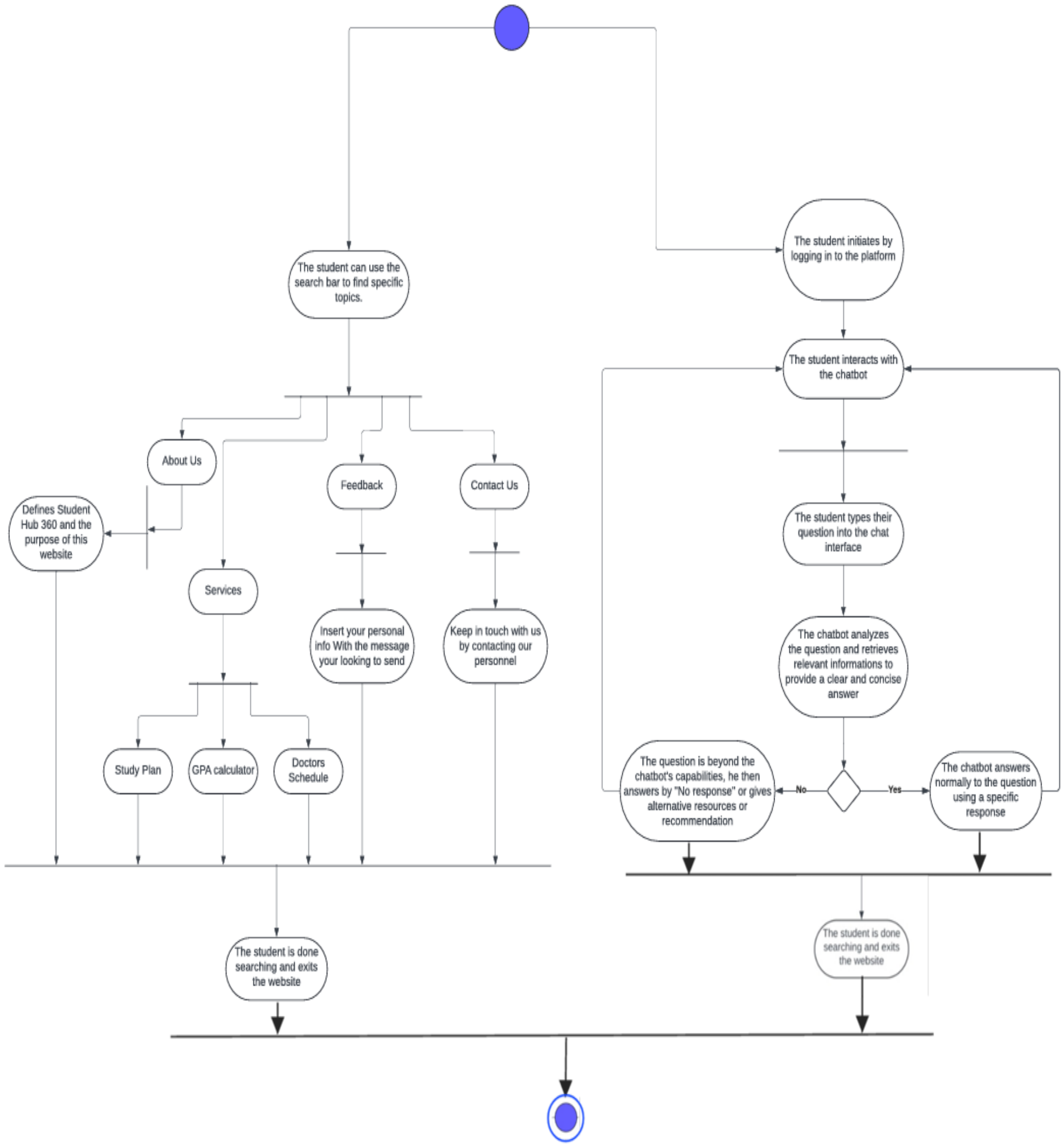
Functional requirements specify the specific functionalities and features that the StudentHub360 platform must provide to meet user needs. These requirements include features such as user authentication and profile management, chatbot functionality for interactive support, resource repository for study materials, interactive collaboration features, notification system for updates and alerts, feedback and evaluation mechanisms, accessibility features, and security controls.

Non-functional requirements define the quality attributes and constraints that the StudentHub360 platform must adhere to, beyond its core functionality. These requirements encompass aspects such as performance, dependability, maintainability, supportability, security, usability, look and feel, operational and environmental considerations, cultural and political sensitivities, and legal compliance. They ensure that the platform delivers a reliable, secure, user-friendly, and compliant experience for all users.

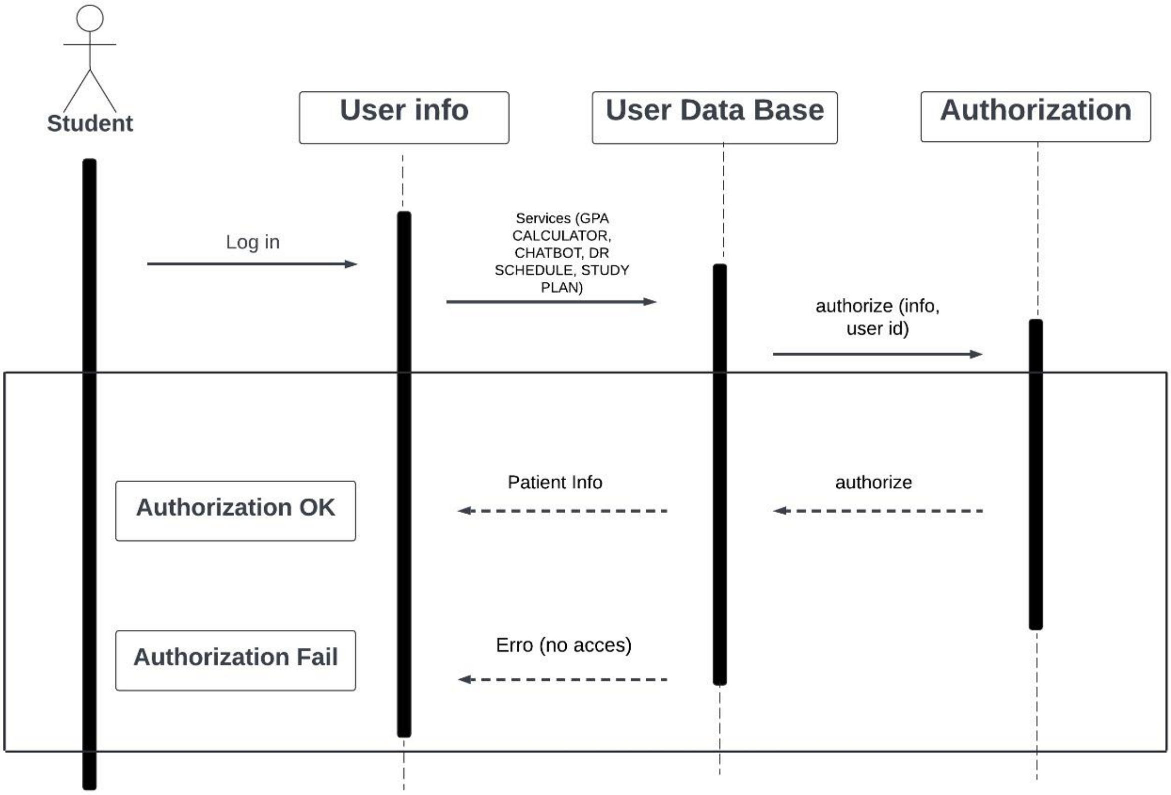
Data requirements outline the types of data that need to be collected, stored, and managed by the StudentHub360 platform to support its functionality, user interactions, analytics, security, compliance, and administrative operations. These requirements include user data such as profiles and preferences, resource data such as metadata and files, communication data such as messages and notifications, analytics data for monitoring and optimization, security and compliance data for auditing and regulatory purposes, feedback and evaluation data for improvement, and administrative data for configuration and support. Proper handling and protection of this data are essential to ensure privacy, security, and regulatory compliance for all users.

Design:

1) Activity Diagram:


The student can use the search bar to find specific topics.
The student initiates by logging in to the platform
The student interacts with the chatbot
About Us
Feedback
Contact Us
Defines Student Hub 360 and the purpose of this website
Services
Insert your personal info With the message your looking to send
Keep in touch with us by contacting our personnel
The student types their question into the chat interface
The chatbot analyzes the question and retrieves relevant informations to provide a clear and concise answer
Study Plan
GPA calculator
Doctors Schedule
The question is beyond the chatbot's capabilities, he then answers by "No response" or gives alternative resources or recommendation
No
Yes
The chatbot answers normally to the question using a specific response
The student is done searching and exits the website
The student is done searching and exits the website

2) Sequence Diagram:



3. Subsystem Decomposition:

- The StudentHub360 website is designed to provide a comprehensive platform for students to access information, provide feedback, and interact with a chatbot. The website is organized into different subsystems/modules, each responsible for specific functionalities. This modular approach helps manage complexity and promotes maintainability.

- Front-End Subsystem:

The front-end subsystem includes the HTML, CSS, and JavaScript components responsible for the user interface and user interaction. The HTML files define the structure and content of the web pages, while CSS is used for styling and layout. JavaScript adds interactivity and dynamic behaviour to the website, enabling features like form submission and chatbot functionality.

- User Authentication and Database Subsystem

Upon logging in, the user's email and password are saved and stored in a database table called "user." This subsystem handles user authentication and data management. SQL language is used to interact with the database, allowing for efficient storage and retrieval of user information. This separation of concerns ensures that user-related functionality is decoupled from other subsystems.

- Home and About Subsystem

The Home and About subsystem provides simple web pages that introduce the StudentHub360 website. These pages are designed using HTML and CSS and offer a brief explanation of the website's purpose and features. By separating this subsystem, it becomes easier to update and modify the content without affecting other functionalities.

- Feedback Subsystem

The Feedback subsystem allows users to submit their feedback through a form. When the user presses the "send" button, the form data is processed and an email is sent to the StudentHub360 email address. This functionality is implemented using formspree, an external service that handles form submissions and email notifications. By encapsulating the feedback functionality within this subsystem, it can be modified or replaced without affecting other parts of the website.

- Chatbot Subsystem

The Chatbot subsystem is the central feature of the StudentHub360 website. It is implemented using JSON for information storage and JavaScript for displaying replies on the website. The JSON file contains information and responses for the chatbot, while JavaScript handles the logic of retrieving appropriate responses based on user input. This modular design allows for easy updates and modifications to the chatbot's behaviour without impacting other subsystems.

In addition to the previously mentioned subsystems, the StudentHub360 website includes a "Services" button that provides users with access to three additional subservices:

- GPA Calculator Subservice:

The GPA Calculator allows students to calculate their Grade Point Average (GPA) based on their course grades. This subservice was implemented using HTML, CSS, and JavaScript. The HTML and CSS ensure an intuitive and user-friendly interface, while JavaScript handles the calculation logic. The GPA Calculator was thoroughly tested before being integrated into the website to ensure accurate results.

- Study Plan Subservice:

The Study Plan subservice provides an image of the study plan specific to the Computer Engineering major. This image visually represents the courses and their sequencing throughout the academic program. By including this subservice, students can easily refer to and plan their course selections. The Study Plan image was carefully chosen and integrated into the website.

- Doctors’ Schedules Subservice:

The Doctor Schedules subservice displays images of schedules for different doctors. It allows users to access the schedules of doctors associated with StudentHub360. By providing these schedules, users can conveniently check the availability and plan appointments with doctors. The images of doctor schedules were integrated into the website to provide a comprehensive healthcare resource for students.

- Integration and Testing

Each subsystem was developed independently and tested thoroughly before integration into the complete website. This approach helps identify and resolve issues early in the development process. Once all subsystems were integrated, comprehensive testing was performed to ensure that the website functions as intended and that interactions between subsystems are seamless.

4. User Interface:

The user interface (UI) design of StudentHub360 aims to provide a visually appealing and intuitive experience for users. The following elements are incorporated into the UI design:

a) Layout and Structure: The UI utilizes a well-organized layout with a clear hierarchy of information. It employs a grid-based structure to ensure consistency and ease of navigation. The interface is divided into sections, including a navigation bar, main content area, sidebar, and footer, enabling users to understand the system's structure and locate information efficiently.

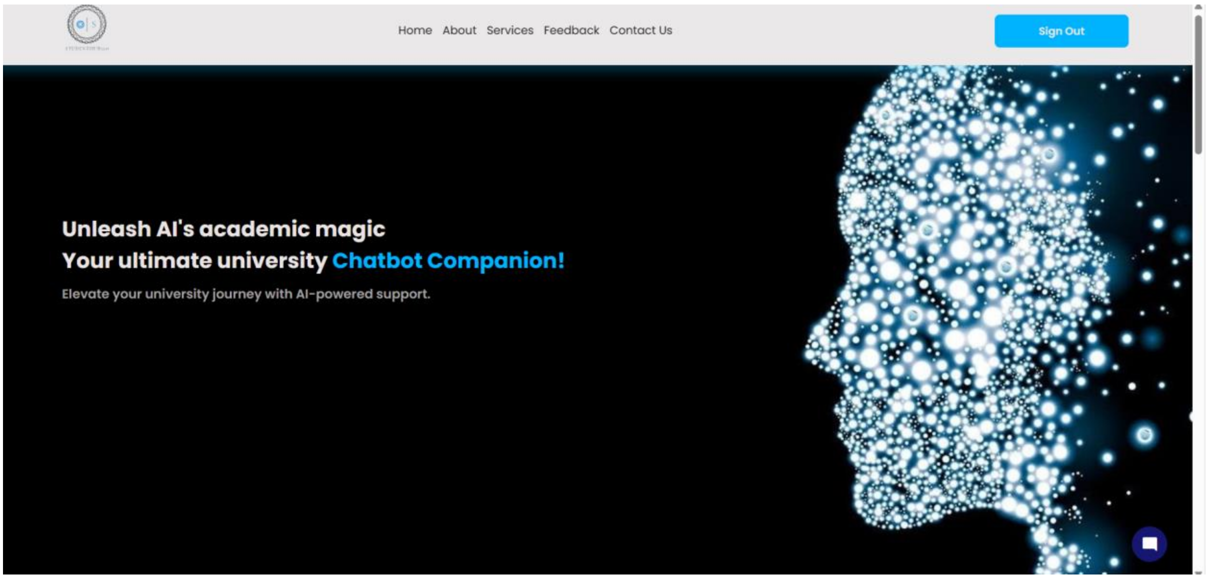
b) Graphical Elements: A visually pleasing colour scheme is implemented throughout the UI, aligning with the branding of StudentHub360. Consistent and easily readable typography is used for headings, body text, and other elements. Relevant icons and visual cues are incorporated to enhance usability, aiding users in understanding and interacting with the interface effectively.

c) Navigation and Interaction Patterns: The UI features a clear and accessible navigation system, allowing users to navigate between different sections and functionalities seamlessly. Interactive elements such as buttons and dropdown menus are employed to facilitate user actions and selections. Consistent interaction patterns are employed, ensuring a predictable user experience. Feedback mechanisms such as visual cues and notifications provide users with timely feedback on their interactions and system responses.

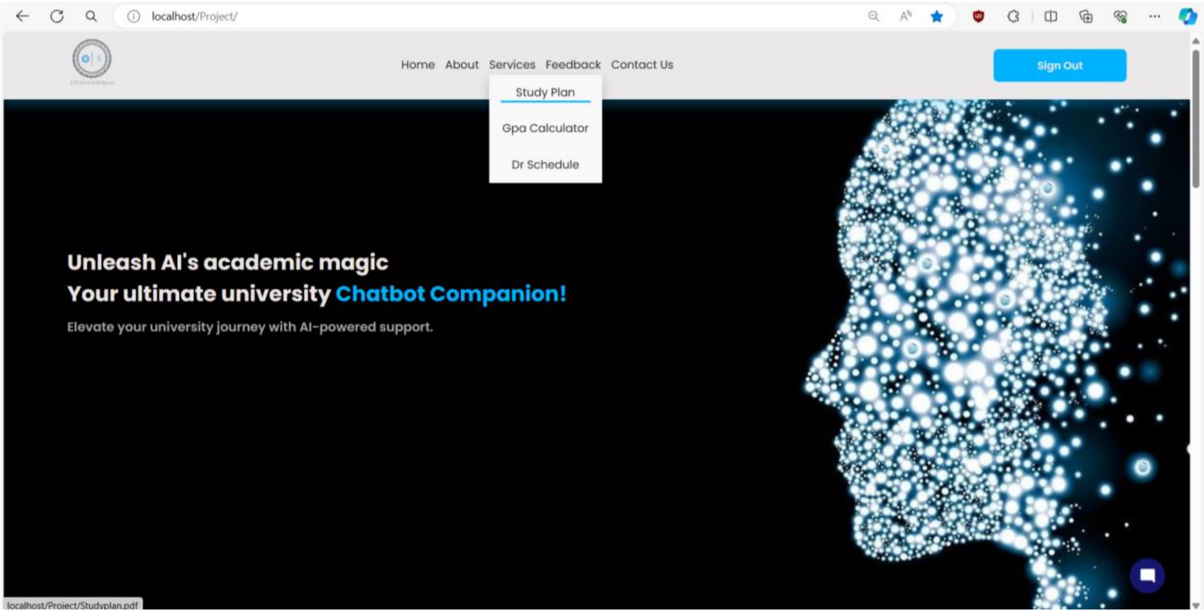
d) Consistency and Branding: The UI design maintains consistency throughout the platform, ensuring a coherent and familiar experience for users. Consistent use of colours, typography, icons, and other visual elements reinforces the branding of StudentHub360. This consistency helps users navigate the interface more easily and fosters a sense of trust and familiarity with the platform.

e) User-Focused Design: The UI design of StudentHub360 puts the users' needs and preferences at the forefront. Extensive user research and usability testing have been conducted to inform design decisions. User feedback and insights have been incorporated to create an interface that is intuitive, user-friendly, and aligns with the expectations and mental models of the target audience. This user-focused design approach aims to enhance user satisfaction and overall engagement with the platform.

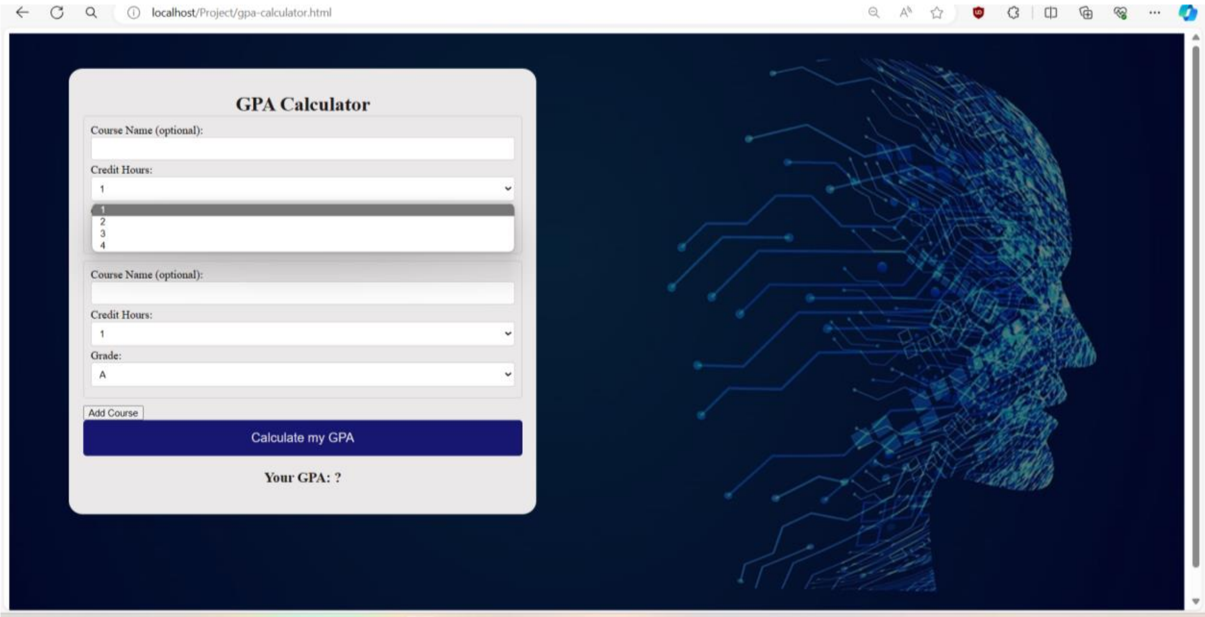
- Here's some screenshots of the user interface:
-



Unleash AI's academic magic
Your ultimate university Chatbot Companion!
Elevate your university journey with AI-powered support.



Unleash AI's academic magic
Your ultimate university Chatbot Companion!
Elevate your university journey with AI-powered support.



GPA Calculator

Course Name (optional):

Credit Hours: 1

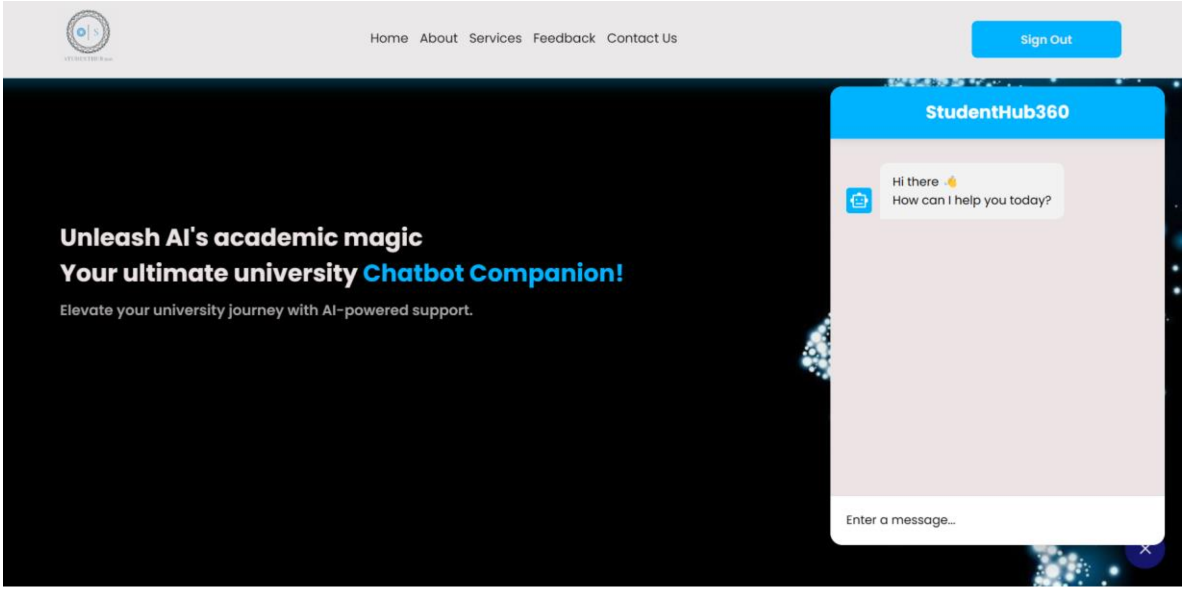
1
2
3
4

Course Name (optional):

Credit Hours: 1

Grade: A

Your GPA: ?



Hi there 👋
How can I help you today?

Unleash AI's academic magic
Your ultimate university Chatbot Companion!

Elevate your university journey with AI-powered support.

Test Plans:

1) Features to be tested / not to be tested:

- Features to be Tested: The features to be tested include user registration, chatbot functionality, resource search, and event registration in StudentHub360.
- Features not to be Tested: There are no specific features excluded from the testing process.

2) Pass/Fail Criteria:

- Pass criteria examples: Successful completion of user registration process, accurate responses from the chatbot and error-free event registration.
- Fail criteria examples: Inability to complete user registration, incorrect or misleading responses from the chatbot, errors or inconsistencies in search results, and issues occurring during event registration.

3) Approach:

- The testing approach for StudentHub360 involves a combination of manual and automated testing.
- Manual testing will be conducted for scenarios that require human interaction and evaluation, while automated testing will be used for repetitive or regression testing tasks.
- Tools like **Selenium** will be employed for automated testing, ensuring efficient test execution and result validation.

→ **Selenium** is an open-source software testing framework widely used for automating web browsers. It provides a suite of tools and libraries that enable developers and testers to automate web application testing across different browsers and platforms. Selenium is a popular tool for automating web browser interactions and testing web applications. It allows testers to write scripts or codes in various programming languages, such as Java, Python, C#, etc., to automate browser actions like clicking buttons, filling forms, navigating through pages, and verifying expected outcomes.

4) Suspension and resumption:

- Testing may be suspended in case of critical system issues, resource unavailability, or changes in project priorities.
- Once the suspension is lifted, testing will be resumed by identifying the point of suspension and continuing from that stage, ensuring proper documentation and communication.

5) Testing materials (hardware / software requirements):

- Hardware Requirements: No specific hardware requirements are needed for testing StudentHub360.
- Software Requirements: The testing environment should include compatible devices (computers, laptops, mobile devices) with modern web browsers (Chrome, Firefox, Safari) to access and test the StudentHub360 platform.

6) Test cases:

- Test cases will be developed to cover various aspects of StudentHub360's functionality.
- Each test case will include a description of the scenario, steps to execute, expected outcome, and any additional notes or requirements.
- The test cases will cover positive scenarios (expected behaviour) as well as negative scenarios (error handling and edge cases) to ensure comprehensive testing.

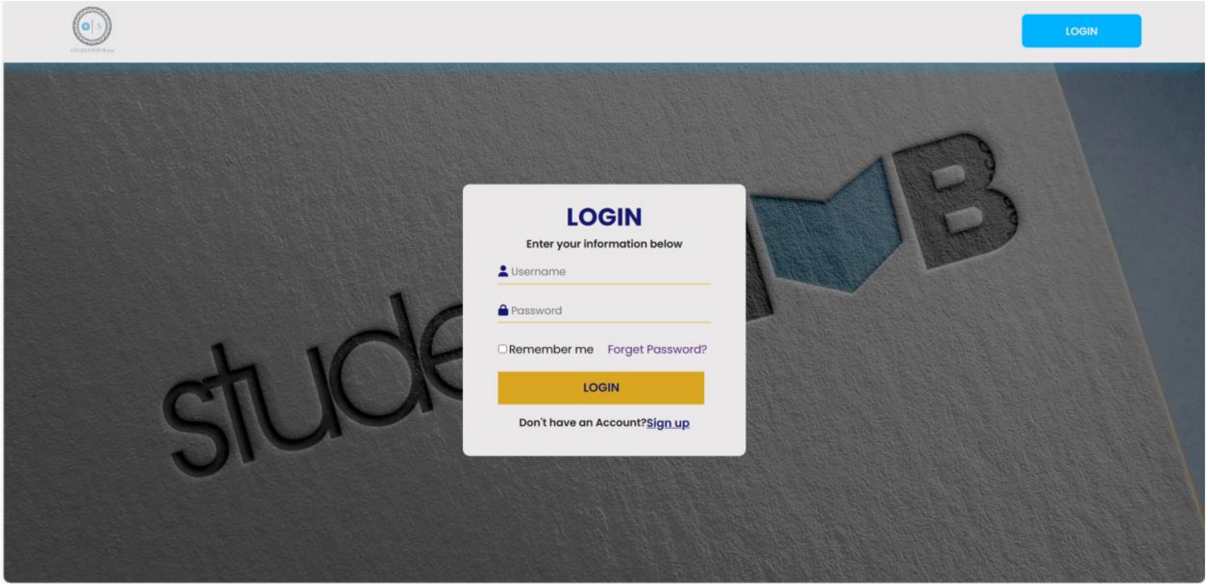
7) Testing schedule:

The testing schedule for the project will encompass regular testing activities as well as responsive testing based on user feedback and complaints. The testing process will be conducted at defined intervals to ensure ongoing quality assurance and to identify any potential issues or bugs in the system. Additionally, whenever negative feedback or complaints are received regarding the website, the testing process will be triggered to investigate and address the reported concerns promptly. This approach aims to continuously improve the website's functionality, usability, and overall user experience, fostering a responsive and user-centric testing methodology.

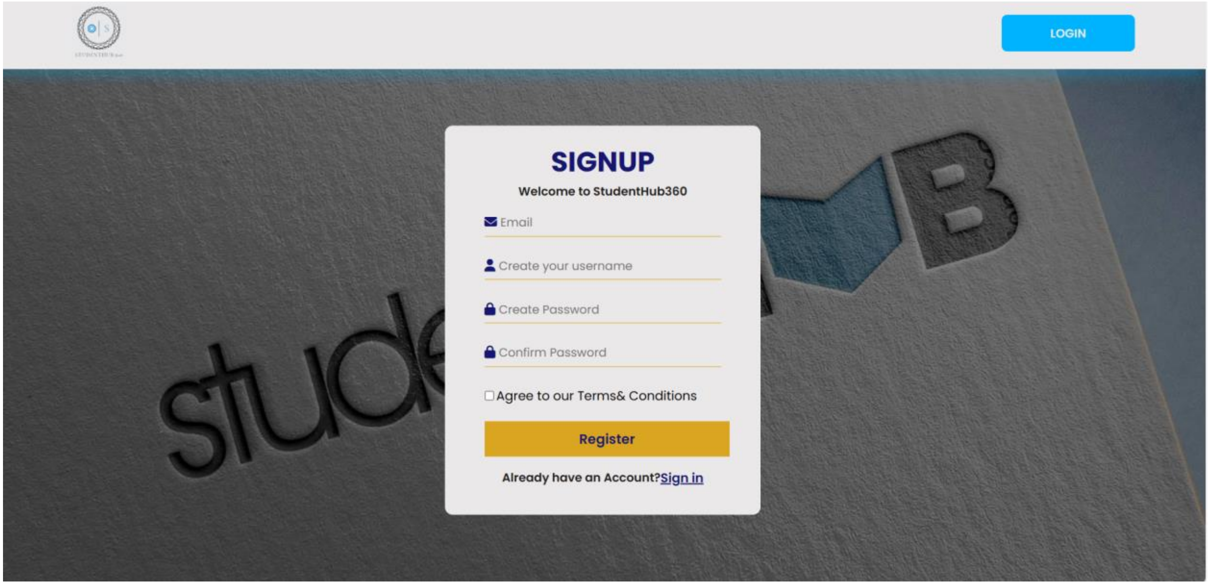
Implementation:

Output of the project:

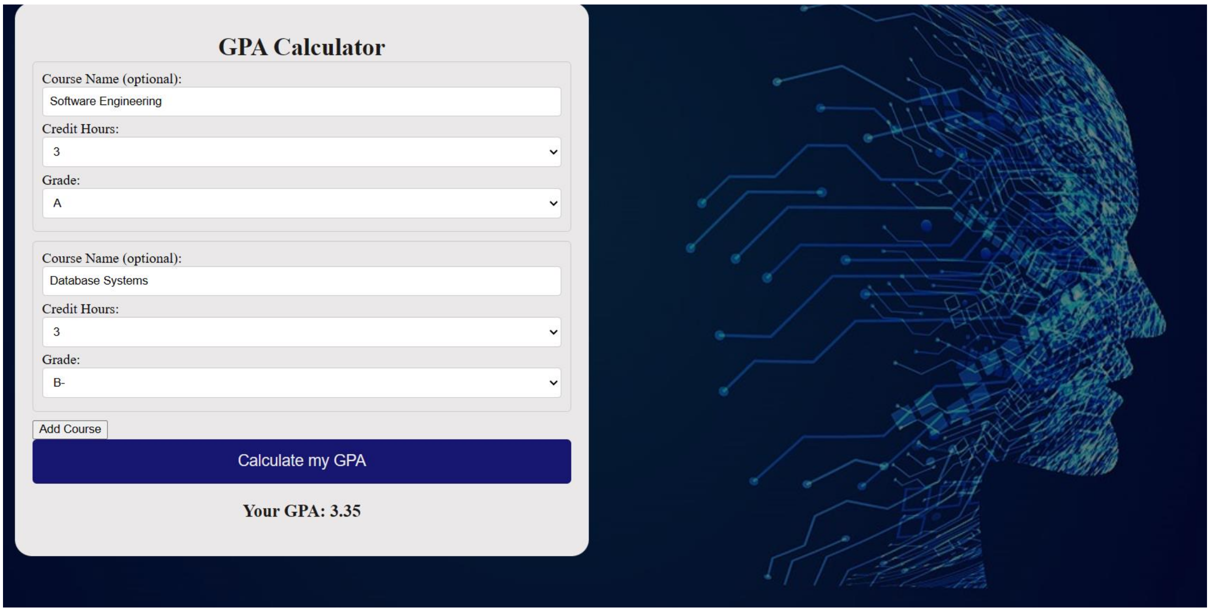
-Login page:



-Sign up page:



-GPA Calculator:



GPA Calculator

Course Name (optional): Software Engineering

Credit Hours: 3

Grade: A

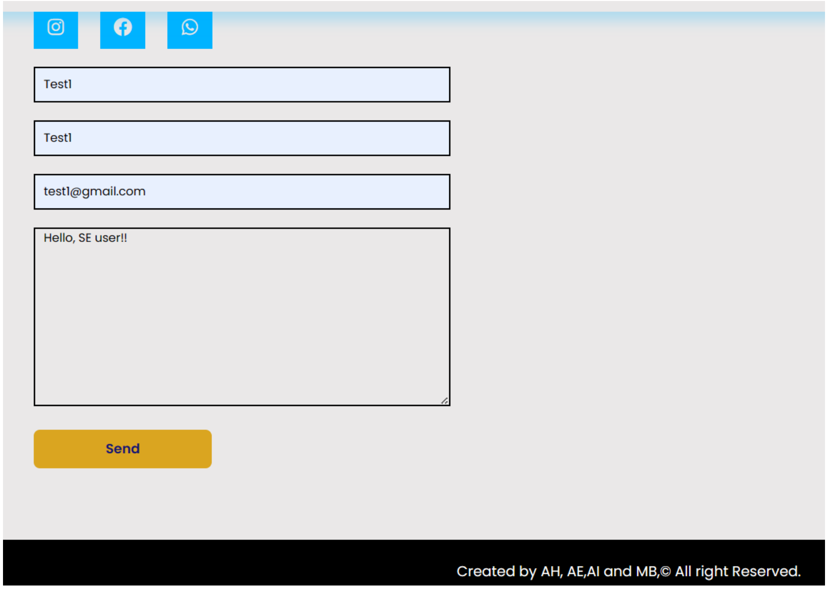
Course Name (optional): Database Systems

Credit Hours: 3

Grade: B-

Your GPA: 3.35

-Feedback system:



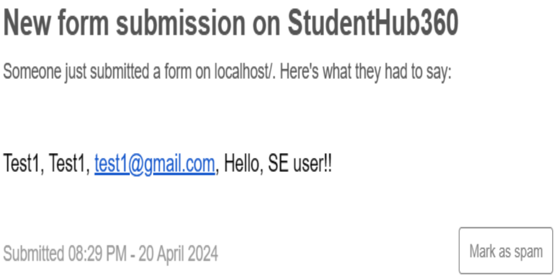
Testl

Testl

testl@gmail.com

Hello, SE user!!

Created by AH, AE,AI and MB,© All right Reserved.



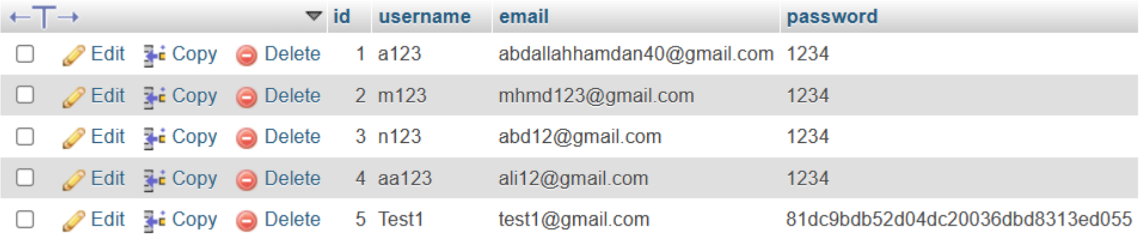
New form submission on StudentHub360

Someone just submitted a form on localhost/. Here's what they had to say:

Test1, Test1, test1@gmail.com, Hello, SE user!!

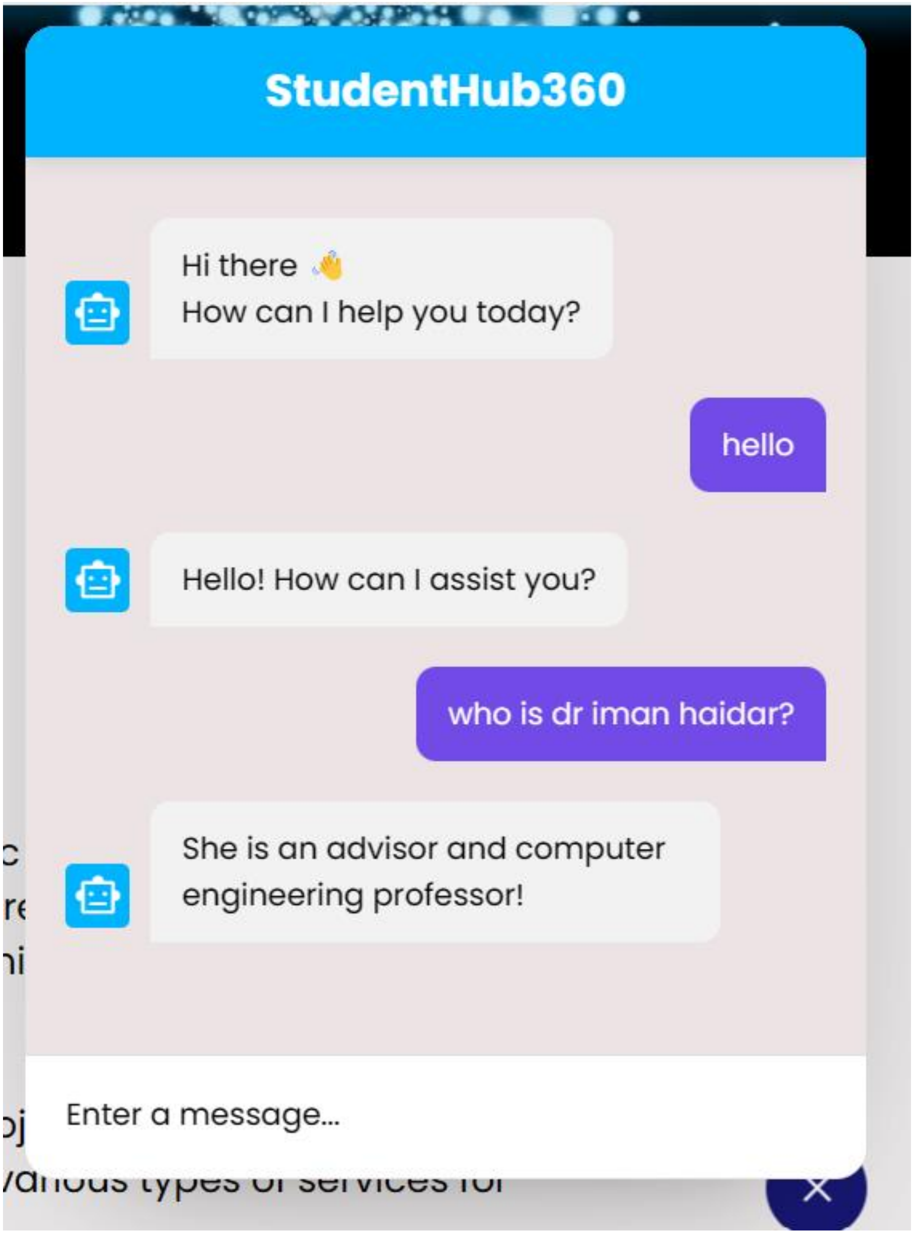
Submitted 08:29 PM - 20 April 2024

-Database for saving users’ credentials:



←T→			▼	id	username	email	password
☐	Edit	Copy	Delete	1	a123	abdallahhamdan40@gmail.com	1234
☐	Edit	Copy	Delete	2	m123	mhmd123@gmail.com	1234
☐	Edit	Copy	Delete	3	n123	abd12@gmail.com	1234
☐	Edit	Copy	Delete	4	aa123	ali12@gmail.com	1234
☐	Edit	Copy	Delete	5	Test1	test1@gmail.com	81dc9bdb52d04dc20036dbd8313ed055

-Chatbot:



StudentHub360

Hi there 👋
How can I help you today?

hello

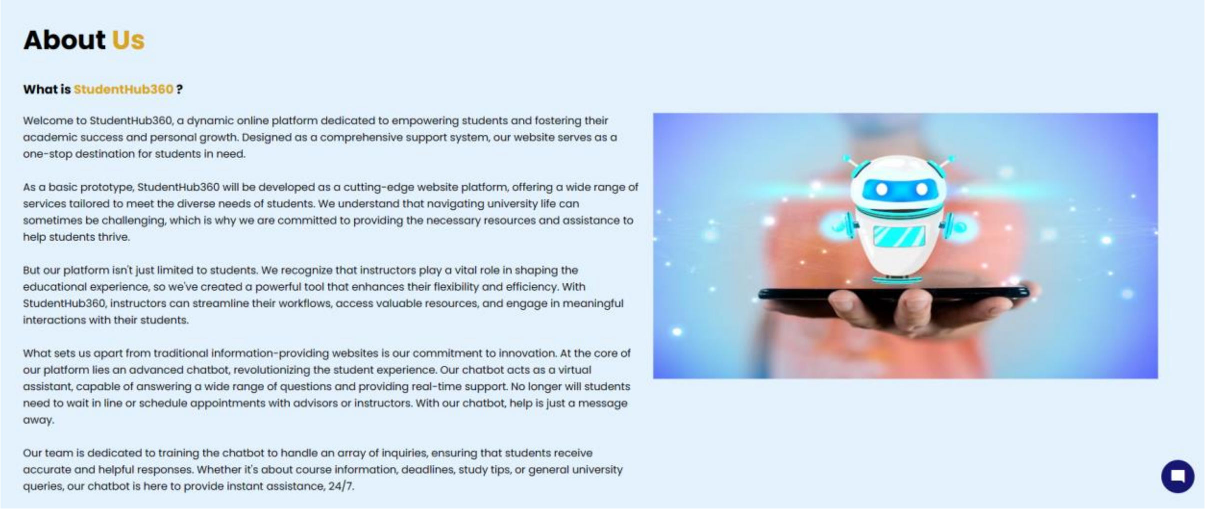
Hello! How can I assist you?

who is dr iman haidar?

She is an advisor and computer engineering professor!

Enter a message...

-About page:



About Us

What is StudentHub360 ?

Welcome to StudentHub360, a dynamic online platform dedicated to empowering students and fostering their academic success and personal growth. Designed as a comprehensive support system, our website serves as a one-stop destination for students in need.

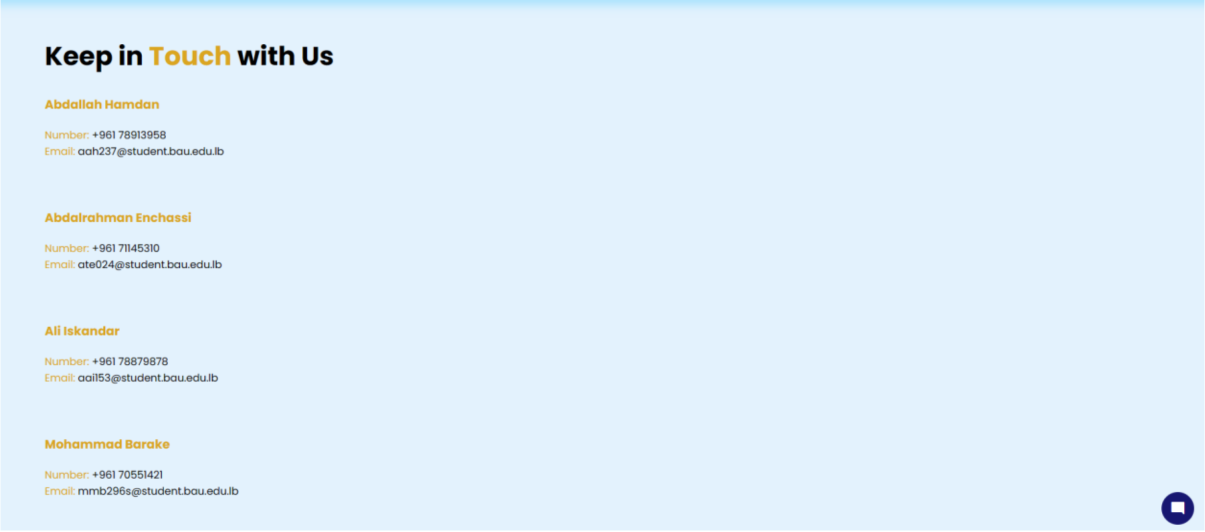
As a basic prototype, StudentHub360 will be developed as a cutting-edge website platform, offering a wide range of services tailored to meet the diverse needs of students. We understand that navigating university life can sometimes be challenging, which is why we are committed to providing the necessary resources and assistance to help students thrive.

But our platform isn't just limited to students. We recognize that instructors play a vital role in shaping the educational experience, so we've created a powerful tool that enhances their flexibility and efficiency. With StudentHub360, instructors can streamline their workflows, access valuable resources, and engage in meaningful interactions with their students.

What sets us apart from traditional information-providing websites is our commitment to innovation. At the core of our platform lies an advanced chatbot, revolutionizing the student experience. Our chatbot acts as a virtual assistant, capable of answering a wide range of questions and providing real-time support. No longer will students need to wait in line or schedule appointments with advisors or instructors. With our chatbot, help is just a message away.

Our team is dedicated to training the chatbot to handle an array of inquiries, ensuring that students receive accurate and helpful responses. Whether it's about course information, deadlines, study tips, or general university queries, our chatbot is here to provide instant assistance, 24/7.

-Contact Us page:



Keep in Touch with Us

Abdallah Hamdan

Number: +961 78913958
Email: aah237@student.bau.edu.lb

Abdalrahman Enchassi

Number: +961 71145310
Email: ate024@student.bau.edu.lb

Ali Iskandar

Number: +961 78879878
Email: aai153@student.bau.edu.lb

Mohammad Barake

Number: +961 70551421
Email: mmb296s@student.bau.edu.lb

Conclusion:

In conclusion, the development of StudentHub360 has successfully addressed the need for a comprehensive online platform catering to the diverse needs of students. Throughout the project, significant milestones were achieved, including the implementation of essential features such as user registration, chatbot functionality, resource search, and event registration. The user interface design has been carefully crafted to ensure a seamless and intuitive experience, while the testing phase has helped identify and rectify any issues or shortcomings.

The feedback received from users during the testing phase has been invaluable in refining and enhancing the platform. User satisfaction levels have shown positive trends, indicating that StudentHub360 is effectively meeting the needs of its target audience. By incorporating user feedback, conducting rigorous testing, and maintaining a user-centric approach, the platform has been optimized for usability, performance, and reliability.

Overall, the successful development of StudentHub360 represents a major milestone in improving the student experience by providing a centralized hub for various academic and non-academic activities. The platform's features and capabilities empower students to efficiently access resources, engage in meaningful interactions, and stay informed about campus events, ultimately enhancing their academic journey.

Future Work:

While StudentHub360 has achieved significant progress, there are several areas for future improvement and expansion. Here are some potential avenues for future work:

1) Mobile Application: Developing a mobile application for StudentHub360 would provide greater accessibility and convenience for users. This would allow students to access the platform on their smartphones or tablets, enabling them to stay connected and engaged regardless of their location.

2) Enhanced Collaboration Features: Introducing features that facilitate collaboration among students, such as group project management tools, discussion forums, and shared document repositories, would further enhance the platform's value. These features could foster a collaborative learning environment and promote teamwork among students.

3) Personalization and Recommendation Engine: Implementing a personalized experience by leveraging user data and preferences could greatly enhance engagement. Incorporating a recommendation engine that suggests relevant resources, events, and opportunities based on user interests and behavior would provide a tailored experience for each student.

4) Integration with Learning Management Systems: Integrating StudentHub360 with existing learning management systems used by educational institutions would streamline workflows and provide a seamless experience for students. This integration could enable direct access to course materials, assignment submissions, and grades, eliminating the need for students to navigate multiple platforms.

References:

[1]: (PDF) Design and Development of CHATBOT: A Review (researchgate.net)

[2]: (PDF) An Interactive Chatbot for College Enquiry (researchgate.net)

[3]: Chatbots applications in education: A systematic review - ScienceDirect

Appendix:

→How to set up the website project:

1. Navigate to the GitHub repository page(AbdallahHamdan2370/StudentHub360 (github.com)) to get all the source codes for the project.

2. Click on the "Code" button, usually located near the top-right of the repository page.

3. In the dropdown menu, click on the "Download ZIP" option. This will initiate the download of a ZIP file containing the entire repository.

4. Once the ZIP file is downloaded, extract its contents to a desired location on your computer.

5. Go to Download XAMPP (apachefriends.org) and download xampp suitable for your computer.

6. Once the download is done, launch the application and press start on “Apache” and “MySQL”.

7. Now, go to This PC→Windows(C:)→xampp→htdocs→paste the extracted zip file here.

8. Go to your browser, type in the following url: http://localhost/Project/ and press enter. The website will be opened successfully!

→How to set up the phpMyAdmin to see the tables:

1. In your browser, open a new tab and enter the following url: localhost / 127.0.0.1 / webproject | phpMyAdmin 5.2.1

2. Navigate through the tables.

→If you’ve got this report as a hard copy:

Here’s the QR codes of each URL mentioned above so you can access them smoothly:

a) AbdallahHamdan2370/StudentHub360 (github.com) :

b) Download XAMPP (apachefriends.org) :