TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

Khwopa College Of Engineering
Libali, Bhaktapur
Department of Computer Engineering

A REPORT ON
STUDY APP: STUDY BUG

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF COMPUTER ENGINEERING

Submitted by

Anjana Silinchhe Shrestha	KCE080BCT004
Pradipta Joshi	KCE080BCT021
Prasant Rai	KCE080BCT024
Sushma Shrestha	KCE080BCT043

Khwopa College Of Engineering
Libali, Bhaktapur

January 28,2026

Certificate of Approval

This is to certify that the project entitled **"StudyBug"** submitted by Anjana Silinchhe Shrestha, Pradipta Joshi, Prasant Rai and Sushma Shrestha to the **Department of Computer Engineering** as a **group project**, in partial fullfillment of the requirements for the award of the degree/course in Computer Engineering. The project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled, bona fide and ready to undertake any commercial and industrial work related to their field of study and hence we recommend the award of Bachelor of Computer Engineering degree.

....
Er. Dinesh Gothe
Head of Department
Department of Computer Engineering, KhCE

....
Er. Sunil Banmala
Supervisor
Department of Computer Engineering, KhCE

....
Er. Ramesh Prajapati
Supervisor
Department of Computer Engineering, KhCE

Copyright

Acknowledgement

We take this opportunity to express our deepest and sincere gratitude to our HoD Er. Dinesh Gothe, for his insightful advice, motivating suggestions for this project and also for his constant encouragement and advice throughout our Bachelor's program.
We are deeply indebted to our Supervisor Er. Sunil Banmala for boosting our efforts and moral by his valuable advises and suggestion regarding the project, giving us realization of the practical scenario of the project and supporting us in tackling various difficulties.

Anjana Silinchhe Shrestha	KCE080BCT004
Pradipta Joshi	KCE080BCT021
Prasant Rai	KCE080BCT024
Sushma Shrestha	KCE080BCT043

Abstract

StudyBug is a web-based aesthetic study platform developed to address common challenges faced by students such as lack of focus, poor time management, and inconsistent study habits. The system integrates essential study-support features including task management, study scheduling, focus timers, and progress tracking within a single, user-friendly interface. Special emphasis is placed on aesthetic design to create a calm and engaging study environment that reduces distractions and increases motivation. The website is fully responsive and accessible across both desktop and mobile devices, allowing users to study flexibly. By combining productivity tools with thoughtful UI/UX design, StudyBug aims to improve study consistency, enhance concentration, and support effective learning practices.

Keywords:*Study website, Focus timer, Study planning, Task management, Aesthetic UI, Student productivity*

Contents

List of Figures

List of Abbreviation

Abbreviations	Meaning
UI	User Interface
UX	User Experience
MVP	Minimum Viable Product
DFD	Data Flow Diagram
API	Application Programming Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
OAuth	Open Authorization
CRUD	Create, Read, Update, Delete

Chapter 1

Introduction

1.1 Background Introduction

In today's academic environment, students are expected to manage heavy workloads, tight deadlines, and continuous assessments. Effective studying requires concentration, proper planning, and consistency. However, many students struggle to maintain focus during study sessions due to distractions such as mobile phones, social media, and mental fatigue. In addition to focus-related issues, the absence of a structured study schedule often leads to procrastination, stress, and inefficient learning.

Although several digital tools such as calendars, to-do lists, and focus timers exist, most of them are either too complex or lack long-term engagement. Many study applications focus solely on functionality and ignore the emotional and visual aspects that influence motivation. As a result, students often abandon these tools after short-term use.

To address these challenges, this project proposes the development of an aesthetic study app that combines focus enhancement, structured scheduling, and an engaging visual experience. The goal is to create a study environment that not only improves productivity but also makes studying more enjoyable and sustainable.

1.2 Motivation

The motivation for this project originates from our personal experience as students. While studying, we frequently faced loss of focus during study sessions, lack of a proper and consistent study schedule and reduced motivation because studying felt monotonous and stressful
We observed that even when study resources were available, the absence of an engaging and organized system made it difficult to stay disciplined. Traditional study methods and existing apps felt repetitive and uninteresting, which further reduced consistency.
This realization led to the idea of developing a study app that is visually aesthetic, structured, and enjoyable to use. By making the study process more appealing and organized, the app aims to help students stay focused, manage time better,

and develop healthier study habits.

1.3 Problem Definition

Despite the availability of numerous productivity and study-related applications, many students continue to struggle with maintaining focus and consistency in their academic routines.
Common challenges include frequent distractions during study sessions, ineffective time management due to a lack of structured scheduling, and low engagement with existing tools that fail to provide visual or emotional motivation. These issues highlight the need for an integrated study application that not only supports focused study and effective planning but also enhances motivation through an aesthetic, intuitive, and user-friendly interface.

1.4 Goals and Objectives

The main objective of this project is:

- To analyze common problems faced by students during studying.

1.5 Scope and Applications

The major scope of the project is to assist students in developing effective study habits. The proposed system will be able to provide:

- Study scheduling and task planning
- Focus timer functionality for structured study sessions
- Progress tracking to monitor study consistency
- Aesthetic UI design to improve motivation and user engagement

1.6 Report Organization

1.6.1 Introduction

The main purpose of the introduction is to set the scene for readers so that they can understand the problems that students face during studying and how the proposed StudyBug system aims to help them.

1.6.2 Literature Review

The literature review is there to familiarize ourselves with the current state of knowledge on study productivity tools and ensure that the proposed solution addresses gaps not covered by existing applications.

1.6.3 Feasibility Study

It is used to determine the viability of an idea, such as ensuring our project is legally and technically feasible as well as economically justifiable or not.

1.6.4 Methodology

It critically helps us to analyze and select correct method for our project to avoid unnecessary hurdles.

1.6.5 Requirement Analysis

It gives us a clear vision about the necessary programming languages and software required to build the StudyBug web application.

1.6.6 System Design and Architecture

The main purpose of this is to evaluate the contribution of each component for overall performance of the system using different diagrams.

1.6.7 Block Diagram and Description of Proposed System

It provides us quick and high-level view of different topics which enhances our ability to understand that topic.

1.6.8 Expected Outcome

It helps to define the anticipated results of the project and ensures that the development stays aligned with the set goals.

1.6.9 Conclusion and Future enhancements

The conclusion section summarizes our main thoughts on project and future enhancements provides us scope to upgrade our project.

Chapter 2

Literature Review

2.1 Assessing the efficacy of the Pomodoro technique in enhancing anatomy lesson retention during study sessions: a scoping review

In this paper, [1] presented a scoping review analyzing the impact of the Pomodoro Technique (PT) on cognitive performance and retention, specifically within the context of anatomy education. The author highlighted that anatomy requires substantial cognitive effort, often leading to mental fatigue when students rely on self-paced study habits. The review found that structured Pomodoro intervals (typically 25 minutes of work followed by 5-minute breaks) were significantly more effective than unstructured, self-regulated breaks.

The study reported that students using the Pomodoro technique experienced approximately 20% lower fatigue levels and a 15–25% increase in self-rated focus compared to control groups. Furthermore, the use of digital tools and timers to enforce these intervals was found to enhance student engagement by 10–18%. The authors concluded that time-structured interventions consistently outperformed self-paced study sessions by reducing distractibility and sustaining motivation over longer periods. This suggests that integrating Pomodoro timers into productivity applications can serve as a critical mechanism for preventing cognitive overload and improving long-term retention of complex material.

2.2 Analyzing the Impact of AI Tools on Student Study Habits and Academic Performance

The integration of mobile technology in higher education has created new opportunities for enhancing student productivity and self-regulation. [2] conducted a comprehensive survey involving 269 academic staff and higher-degree students, revealing that while nearly 95% of students possess smartphones, the use of mobile applications for academic purposes is largely driven by personal motivation rather than institutional requirements. The study found that current app usage is predominantly focused on basic document storage and communication tools such as Dropbox, rather than specialized academic process-management applications. However, a significant gap exists between current usage patterns and student needs. [2] reported that students specifically recommended applications for project and assignment planning, and non-users expressed a strong willingness to adopt academic apps if they were easier to use and more appropriate for their academic context. These findings indicate a clear demand for simplified, student-centric productivity applications that extend beyond basic file storage.

To complement the structural benefits of planning applications, recent research highlights the cognitive advantages of time-management techniques such as the Pomodoro method. [1] presented a scoping review examining the impact of the Pomodoro Technique on anatomy students, a population frequently exposed to high cognitive load. The review demonstrated that structured study intervals, typically consisting of 25 minutes of focused work followed by 5-minute breaks, were significantly more effective than unstructured, self-regulated breaks. Students using this technique reported approximately 20% lower fatigue levels and a 15–25% increase in self-rated focus compared to control groups. Importantly, [1] emphasized that digital tools and timers were critical in enforcing these intervals, resulting in a 10–18% increase in student engagement. Taken together, these studies suggest that a productivity application combining autonomous planning features with structured focus techniques such as Pomodoro timers would effectively address the gaps in academic app usage identified by [2].

2.3 A Study of Mobile App Use for Teaching and Research in Higher Education

In a comprehensive study on the integration of mobile technology in universities, [2] surveyed 269 academic staff and higher-degree students to examine how mobile applications are utilized in academic settings. The study established that the hardware barrier to mobile learning is virtually non-existent, with recent data indicating that approximately 95% of students possess smartphones. Despite this widespread availability, the authors found that mobile app usage is primarily driven by *personal motivation* rather than institutional planning. This suggests that students independently select digital tools to manage their academic activities, thereby validating the demand for student-centric productivity applications.

The study further revealed that the current academic app ecosystem is dominated by basic utility tools. Applications for document storage, such as Dropbox, and communication were reported as the most frequently used, whereas specialized study or process-management tools were comparatively underutilized. However, the findings also highlighted a clear demand for applications that support structured academic work. When participants were asked to recommend purposes for academic app usage, *project and assignment planning* emerged as a key category. Additionally, among participants who did not currently use mobile apps for academic purposes, a significant proportion expressed an intention to adopt such tools in the future, particularly for project or assignment planning and note-taking.

Chapter 3

Feasibility Study

3.1 Technical Feasibility

The proposed aesthetic study website is technically feasible as all required features can be implemented using standard web technologies. Features such as task planning, scheduling, focus timers, progress tracking, reminders, and theme customization can be developed using HTML, CSS, JavaScript, and modern frameworks like React. The system can support multiple devices through a responsive design. Data storage can be handled using client-side storage for offline use or databases such as Supabase or PostgresSQL for persistent, login-based access.

3.2 Operational Feasibility

Operational feasibility assesses how well the system will work in real conditions and how easily users can adopt it. The proposed website is operationally feasible because it addresses real student challenges: lack of focus, unstructured schedules, and low motivation. The system is designed to be simple and practical—users can plan tasks, schedule study time, run focus sessions, and track progress within one platform. Since the website is accessible through any browser, users do not need installation, which improves accessibility and adoption. Module-wise operational feasibility is described below:

3.2.1 Focus Timer and Study Session Module

This module supports focused study techniques such as Pomodoro. Users can start and pause study sessions, take short breaks, and record study time. A distraction-minimized focus view improves usability. Since timers and session counters are basic web functions, this module can operate reliably in a browser environment.

3.2.2 Study Scheduling and Planner Module

The scheduling module helps users create daily or weekly study plans and allocate time blocks for subjects or tasks. This improves structure and reduces procrastination. The module is operationally practical as users can edit schedules easily, and the interface can present schedules in a clean and understandable format.

3.2.3 Aesthetic UI, Themes, and Personalization

This module focuses on the visual appeal of the website through themes, layouts, fonts, and calming design elements. Aesthetic design improves engagement and increases long-term use by making the study environment more pleasant. This supports the core idea of the project: studying should feel less stressful and more enjoyable.

3.3 Economic Feasibility

The project is economically feasible because it can be developed using free and open-source tools. Hosting options such as GitHub Pages, Netlify, and Vercel offer free or low-cost deployment, while backend services are available on free tiers if required. Development costs are minimal and mainly involve time and effort, making the project suitable for an academic budget.

3.4 Scheduling Feasibility

The proposed aesthetic study website is schedule-feasible as it can be developed within an academic semester using a phased and modular approach. Initial weeks can be dedicated to requirement analysis, literature review, and user interface planning, followed by wireframe design and theme selection. Core development, including task management, study scheduling, focus timers, and data storage, can be completed in the middle phase of the timeline. Subsequent weeks can focus on progress tracking features, usability testing, responsive design improvements, and bug fixing. The final phase can be used for documentation, deployment, and presentation preparation. Due to the manageable scope and clear module separation, the project can be completed within the available time without schedule overruns.

Chapter 4

Methodology

4.1 Agile Method as Software Development Model

The development of the Aesthetic Study Website follows the Agile software development methodology. Agile was chosen because the project emphasizes user experience, iterative design, and continuous improvement. Since the website includes multiple interactive features such as scheduling, focus timers, task management, and aesthetic customization, Agile allows these components to be developed and refined incrementally. The Agile approach enables frequent evaluation of progress, early detection of issues, and flexibility in incorporating feedback. Each iteration focuses on delivering a functional and usable version of the website, which is then enhanced in subsequent iterations.

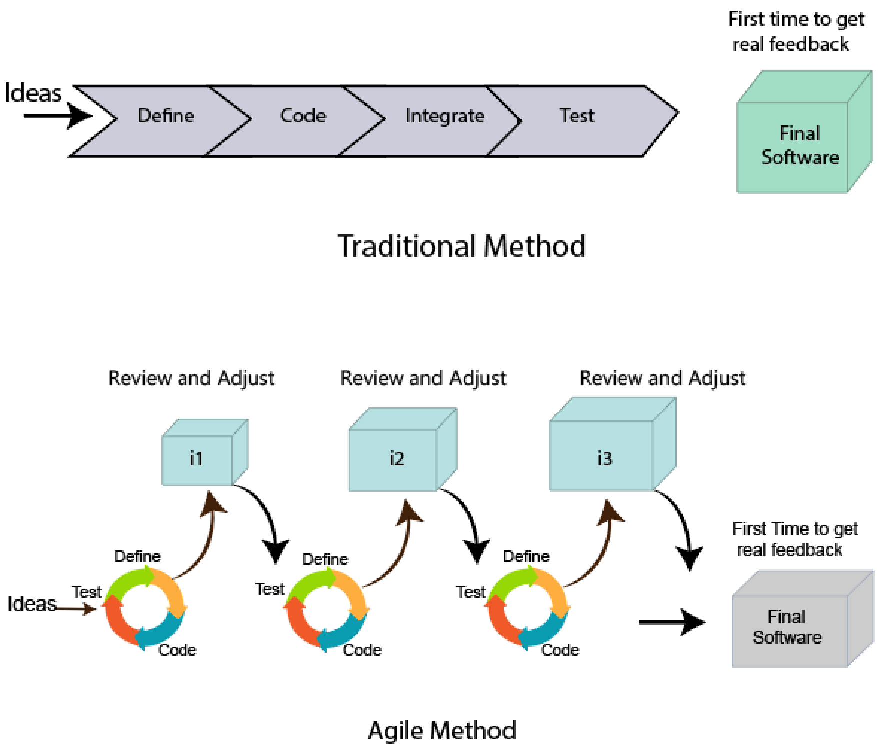

Figure 4.1: Agile Model as Software Development Model

4.2 Figma for Ideation and Prototyping

Figma was used for ideation and prototyping to design wireframes and interactive layouts of the StudyBug website. It helped visualize user flows, refine the interface, and ensure an aesthetic and user-friendly design before development.
The Project will be managed in Notion in five sessions:

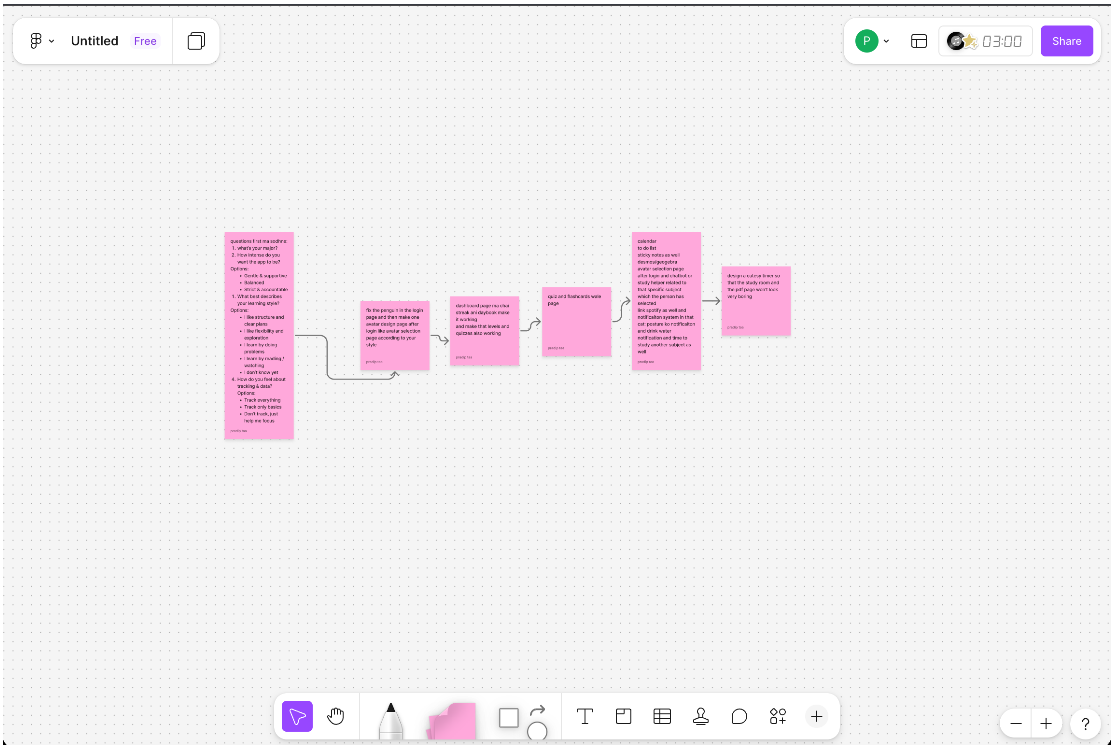

Figure 4.2: Figma Workflow

4.2.1 Project Plan

Planning for further steps will be done in detail in this session. All the tasks to be done, their sub tasks, deadline of the tasks and subtasks will be recorded here. Then the task will be linked to respective iteration session by the assigned member before doing the task.

4.2.2 Iteration1

Everything related to the first iteration of our project will be recorded here including meeting minutes and reports.

4.2.3 Iteration2

Everything related to the second iteration of our project will be recorded here including meeting minutes and reports.

4.3 Overall Phase to be Followed

The overall project will be completed in three main phases which are:

1. Planning Phase
2. Development Phase
3. Integration

4.3.1 Planning Phase

The planning phase involved identifying the problem statement, defining project objectives, and gathering requirements based on student study challenges. During this phase, features were finalized and the overall structure of the StudyBug website was designed using Figma wireframes.
First the project was divided into four parts:

1. UI/UX design
2. Frontend
3. Backend
4. Database

These parts were then assigned to each project members who then studied about the respective parts in detail.

4.3.2 Development Phase

In this phase, the divided parts will be studied and developed. Then each of those developed parts will be tested separately.

4.3.3 Integration

In this phase, the separately developed parts will be integrated to form a system and integration testing will be done.

4.4 Task Workflow

Each task of every session will be done and recorded in a procedural manner and the programmes will be recorded and stored in GitHub.
The workflow of the tasks will follow these steps:

1. Get task assigned on the basis of agreement.

2. A shared GitHub repository will be created for the StudyBug project.

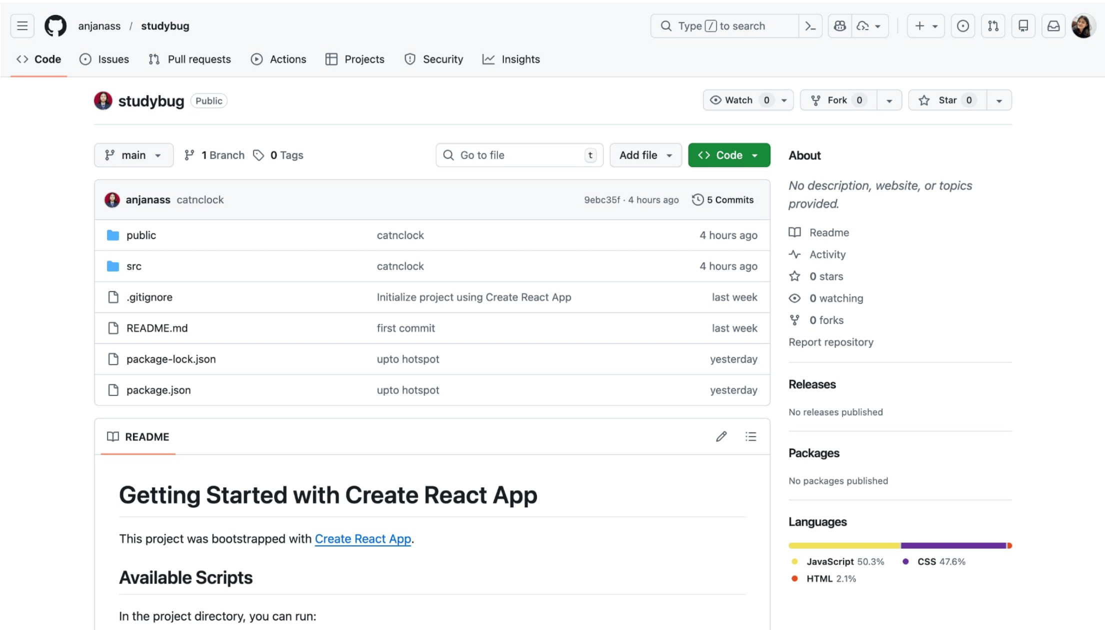

Figure 4.3: Create a shared github repository

3. Each team member will work on their assigned tasks using separate branches.

4. Changes will be implemented locally and pushed to the GitHub repository.

5. Completed work will be reviewed and merged into the main branch for integration.

Chapter 5

Requirement Analysis

5.1 Software Requirement

The proposed system is a web-based application and requires commonly available software for both development and usage. To access the website, users require a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge. Overall, the required software tools are easily accessible and widely supported.

5.2 Hardware Requirement

Since the system is web-based, the hardware requirements are minimal. For development purposes, a standard laptop or desktop computer with sufficient processing capability and a stable internet connection is adequate. For end users, any device capable of running a modern web browser—such as a laptop, tablet, or smartphone—can be used to access the website. No specialized hardware is required, making the system accessible to a wide range of users.

5.3 Functional Requirement

The aesthetic study website is designed to assist users in improving focus, organization, and study consistency. The system allows users to create and manage study tasks, organize daily and weekly study schedules, and conduct focused study sessions using a built-in timer. Users can start, pause, and reset study sessions as needed. The system records completed tasks and study sessions and displays progress summaries to help users track their productivity. Additionally, the website provides an aesthetic and distraction-minimized interface to enhance motivation. Users can personalize the appearance of the website through theme or layout options, and study data is stored either locally or in a database to allow continued usage over time. Optional login functionality may be provided to enable access across multiple devices.

5.4 Non-Functional Requirement

Non-functional requirements define the quality attributes of the system. The website should be reliable and function smoothly without frequent errors or data loss during normal usage. The system should be easy to maintain, with modular code that allows future enhancements and feature additions. Performance is also an important consideration; the website should load quickly and respond efficiently to user interactions such as scheduling tasks or starting focus sessions. Finally, usability is a key requirement, as the interface should be intuitive, visually appealing, and responsive across different screen sizes to ensure a positive user experience for all users.

Chapter 6

System (or Project) Design and Architecture

6.1 Use Case Diagram

The Use Case Diagram of the proposed system.

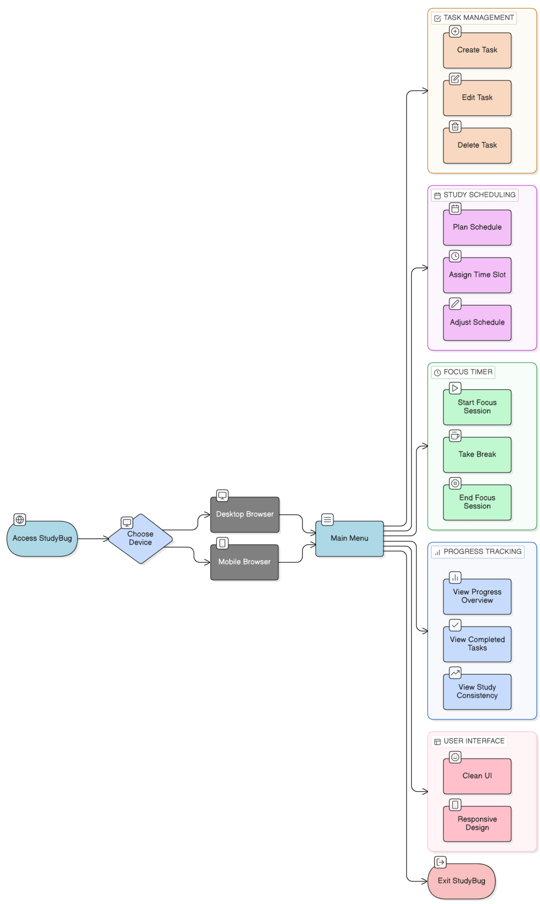

Figure 6.1: Use case Diagram

6.2 Context Diagram

The Context Diagram shows the top level picture of the proposed system.

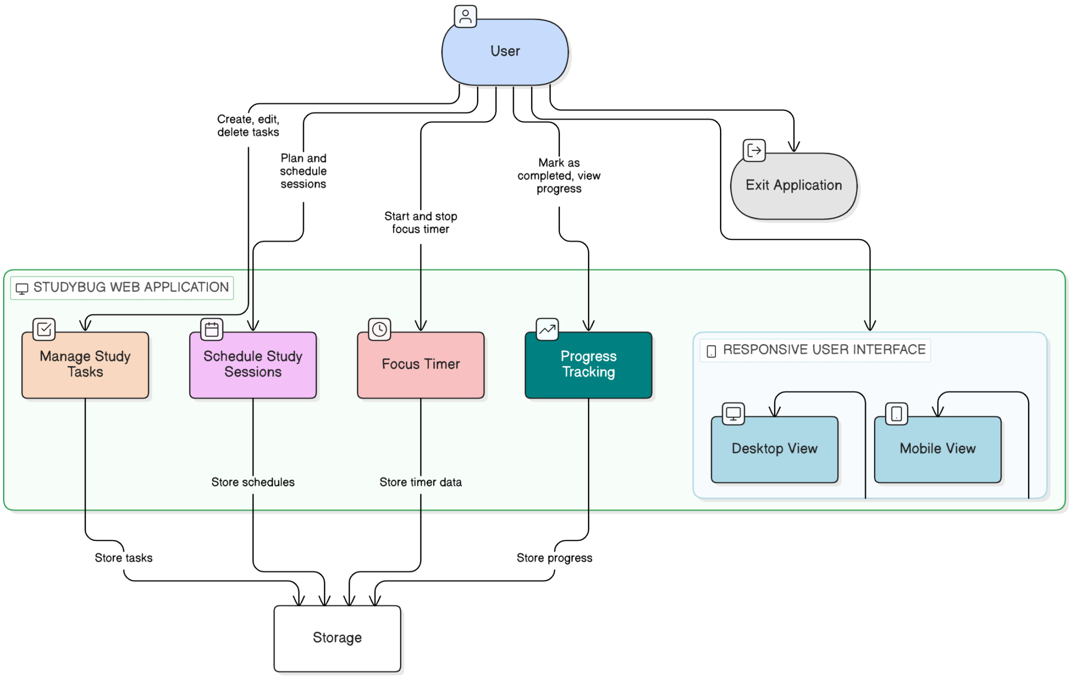

Figure 6.2: Context Diagram of the Proposed System

6.3 Data Flow Diagram

The Data Flow Diagram shows the flow of the data between the subsystems of the proposed system. The Data Flow Diagram is shown below:

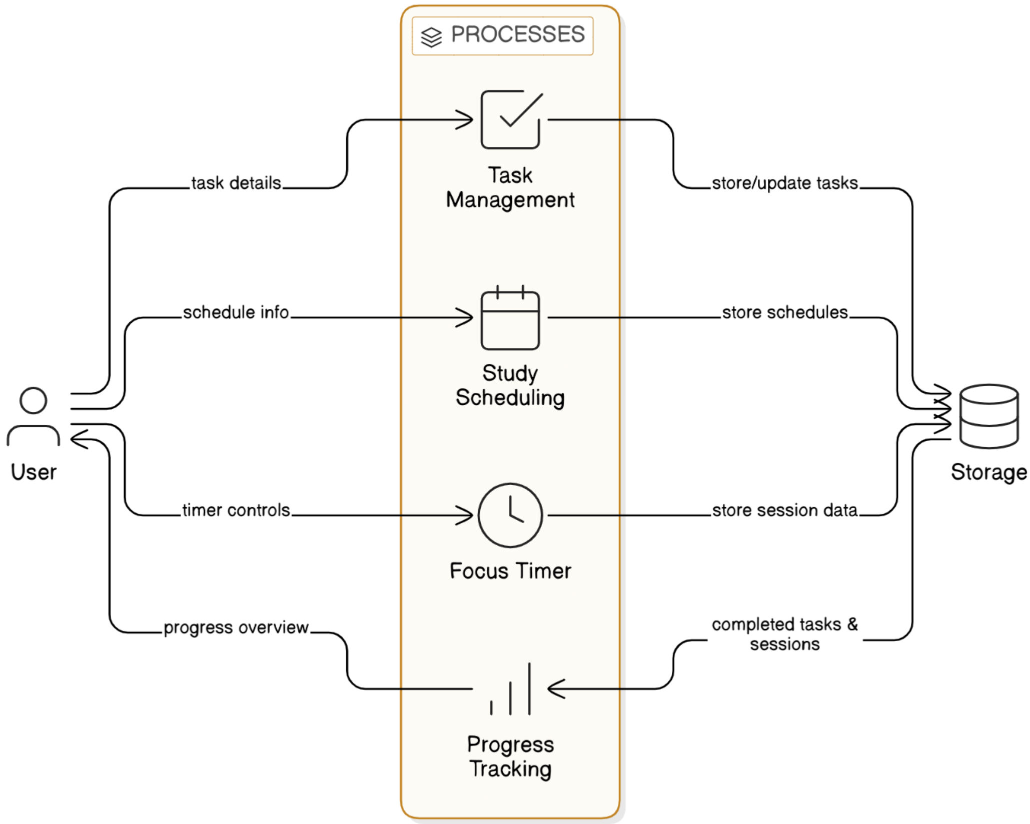

Figure 6.3: Data Flow Diagram of the Proposed System

6.4 Workflow Diagram

The Sequence Diagram shows the flow of the process in between the subsystems. And the state when the subsystems are active and when the subsystems are passive. The Sequence Diagram of the proposed system is shown below:

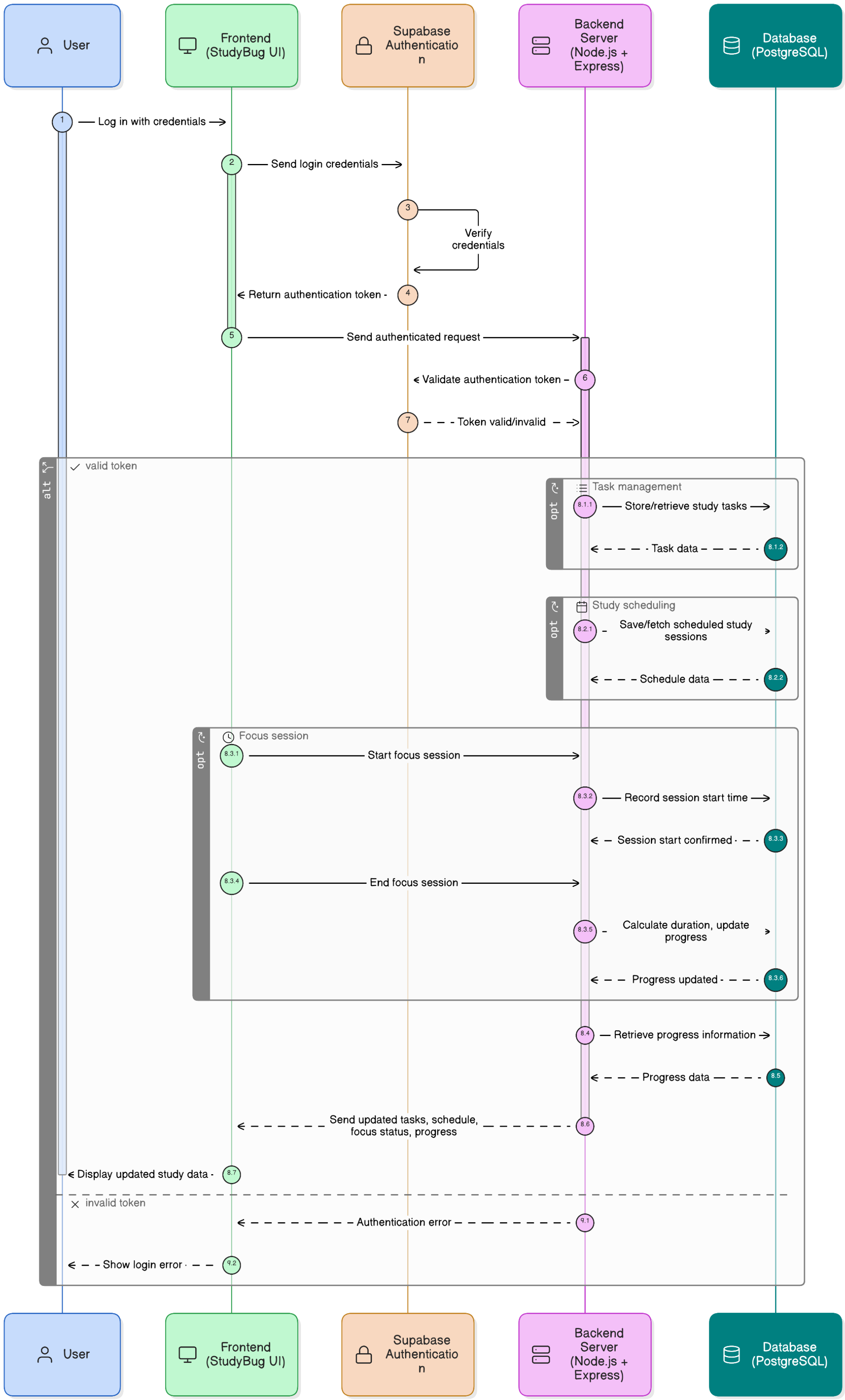

Figure 6.4: Sequence Diagram of the Proposed System

6.5 Entity Relationship Analysis

The Entity-Relationship (ER) analysis defines the proposed data architecture of the system, identifying the primary objects and how they will interact to support the task and focus tracking workflow.

6.5.1 Entity Classification

Entities are classified based on their existential dependency within the system.

- **Strong Entities**: These entities possess independent existence and are identified by their own unique attributes.
  - **USER**: Represents the primary actor; identified by `id`.
  - **TASK**: Represents the work unit created by a user; identified by `task_id`.
  - **SCHEDULE**: Represents planned time allocations on a calendar; identified by `schedule_id`.
- **Weak Entities**: These entities are existence-dependent on a parent strong entity.
  - **PROGRESS**: Dependent on **TASK**. It tracks the quantitative state of a task and cannot exist without a parent task record.
  - **FOCUS-SESSION**: Dependent on **TASK**. It logs specific execution instances and duration toward a task's completion.

6.5.2 Relationships and Cardinality

The following table defines the logic and constraints governing the interactions between the entities.

Entity A	Entity B	Cardinality	Description
USER	TASK	$1 : N$	One user can manage multiple tasks.
TASK	FOCUS-SESSION	$1 : N$	A task can be composed of many work sessions.
TASK	SCHEDULE	$1 : N$	A single task can be scheduled for multiple time slots.
TASK	PROGRESS	$1 : N$	A task generates sequential progress records over time.
SCHEDULE	FOCUS-SESSION	$0 : N$	A session optionally fulfills a schedule (allows spontaneous sessions).
FOCUS-SESSION	PROGRESS	$1 : 1$	Each completed session triggers a specific progress update.

Table 6.1: Entity Relationship and Cardinality Table

6.5.3 Design Assumptions

The proposed database architecture will be built upon the following logical assumptions:

1. **Spontaneous Execution**: The `schedule_id` in the `FOCUS-SESSION` entity is nullable, assuming that users may perform work sessions without prior scheduling.

2. **Derived Progress**: The `percent_complete` attribute is a derived value calculated using the formula:

$$\text{percent\_complete} = \left(\frac{\text{Total Time Spent}}{\text{Estimated Minutes}} \right) \times 100$$

3. **Atomic Tasks**: A `FOCUS-SESSION` can only be associated with one `TASK` at a time to ensure accurate focus tracking.

4. **Data Persistence**: Progress records are maintained as a history of the task's evolution rather than overwriting a single value, allowing for trend analysis.

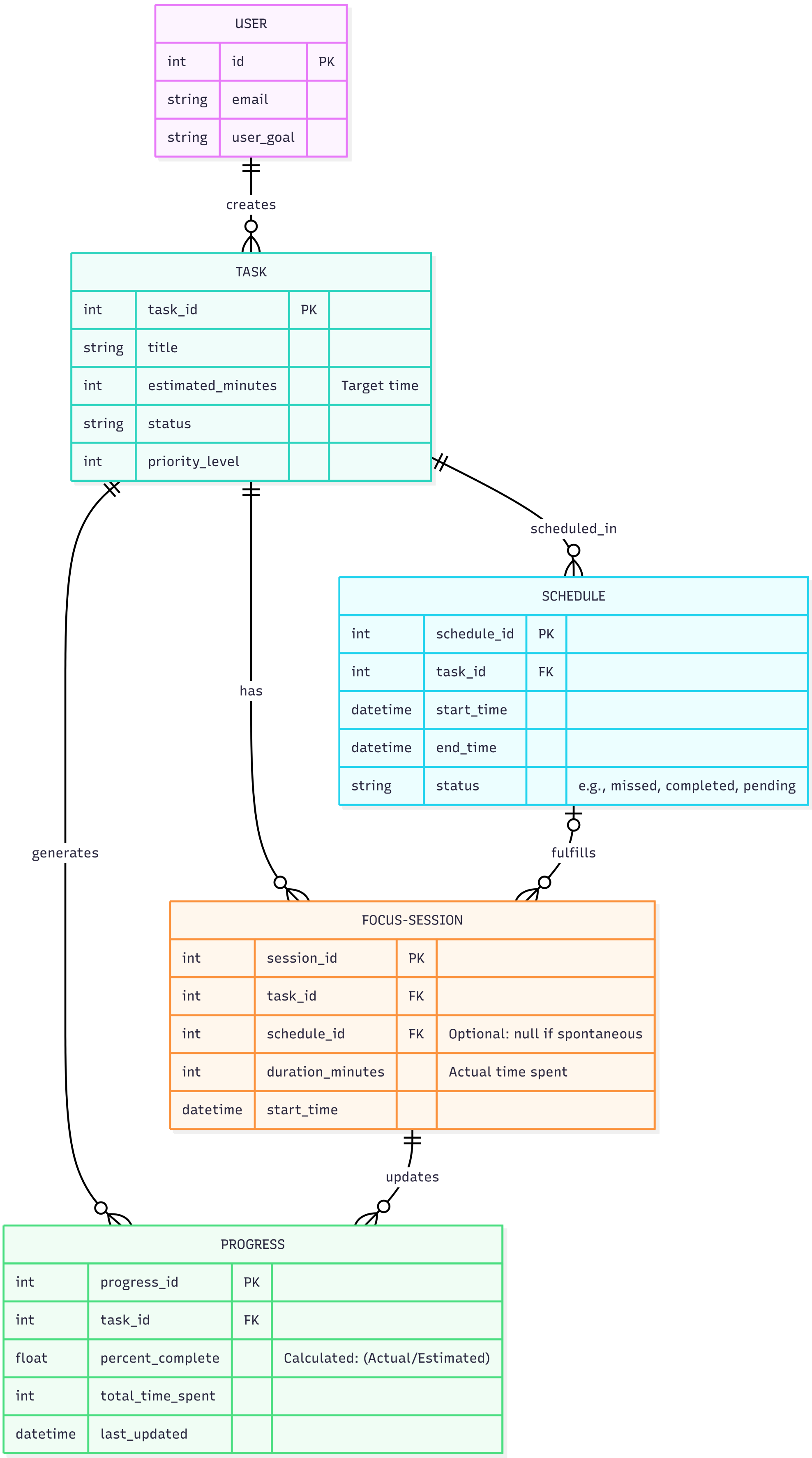

Figure 6.5: Entity Relationship Diagram

Chapter 7

Block Diagram and Description of Proposed System

7.1 System Block Diagram

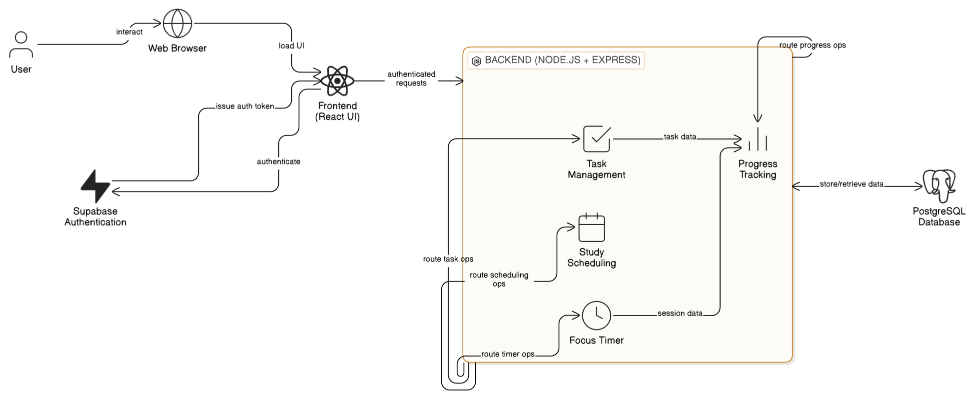

Figure 7.1: System Block Diagram of StudyBug

The system block diagram illustrates the proposed architecture of the *StudyBug* web application and shows how different system modules will interact to provide an integrated study-support platform. The system is designed to follow a client–server architecture and will be accessed using a standard web browser.

When a user opens the StudyBug website, the frontend interface will load in the browser and provide access to various study-support features. User actions such as task creation, study scheduling, and focus session initiation will be sent from the frontend to the backend server through secure API requests. User authentication will be handled through an external authentication service, ensuring secure access to personalized data. All study-related information will be stored and managed in a centralized database. The processed data will then be returned to the frontend and presented to the user in the form of schedules, progress summaries, and visual feedback.

7.2 Frontend Interface Module

The frontend interface module will be responsible for user interaction and visual presentation. It will be developed using modern web technologies and will follow responsive design principles to support both desktop and mobile devices. The frontend will provide interfaces for task management, study scheduling, focus timer execution, progress tracking, and aesthetic customization. Its primary role will be to capture user input, display system output, and communicate with backend services.

7.3 Authentication Module

The authentication module will manage user login and access control. It will verify user credentials and issue authentication tokens that allow secure communication between the frontend and backend. This module is designed to ensure that each user can access only their own study data while maintaining system security.

7.4 Backend Server Module

The backend server will act as the central control unit of the system. It will receive requests from the frontend, process application logic, and coordinate interactions between different modules. The backend will manage task operations, scheduling logic, focus session handling, and progress computation. It will also handle data validation and communicate with the database for data storage and retrieval.

7.5 Task Management Module

The task management module will enable users to create, update, delete, and mark study tasks as completed. Each task will represent a specific academic activity such as assignments, revision topics, or project milestones. Task-related requests will be processed by the backend server and stored persistently in the database, allowing users to access their tasks across sessions and devices.

7.6 Study Scheduling Module

The study scheduling module will allow users to plan daily or weekly study routines by assigning time slots to specific tasks or subjects. This module will retrieve task information from the database and organize it into structured schedules. The generated schedules will then be sent to the frontend for visualization, helping users manage time effectively.

7.7 Focus Timer Module

The focus timer module will support structured study sessions inspired by time-management techniques such as the Pomodoro method. Users will be able to start, pause, and reset focus sessions through the frontend interface. The backend will record focus session data, which will contribute to overall progress tracking and study analysis.

7.8 Progress Tracking Module

The progress tracking module will analyze completed tasks and recorded focus sessions to generate basic progress summaries. These summaries will help users monitor their study consistency and productivity over time. Progress data will be calculated by the backend and stored in the database for future reference.

7.9 Database Module

The database module will be responsible for persistent data storage. It will store user accounts, tasks, study schedules, focus session logs, and related metadata. A centralized database will ensure data consistency, reliability, and accessibility across multiple user sessions and devices.

7.10 System Workflow

The overall system workflow will begin when the user accesses the StudyBug website through a web browser. After authentication, the user will interact with the frontend to manage tasks, plan schedules, or initiate focus sessions. The frontend will send these requests to the backend server, which will process the logic and update the database accordingly. The updated information will then be returned to the frontend and displayed to the user, completing the interaction cycle.

7.11 Advantages of the Proposed System

- Will provide a centralized and secure platform for study planning and tracking
- Will ensure persistent data storage through a backend database
- Will support access across multiple devices and sessions
- Will encourage structured study habits using scheduling and focus timers
- Will offer a clean and distraction-minimized study environment

Chapter 8

Expected Outcome

8.1 Desktop Outcome Expected

On desktop or laptop devices, the website is expected to offer a clean and well-organized interface with easy navigation between tasks, schedules, focus timers, and progress tracking. Users should be able to comfortably plan their study routines, run focused study sessions, and view productivity summaries on a larger screen without visual clutter.

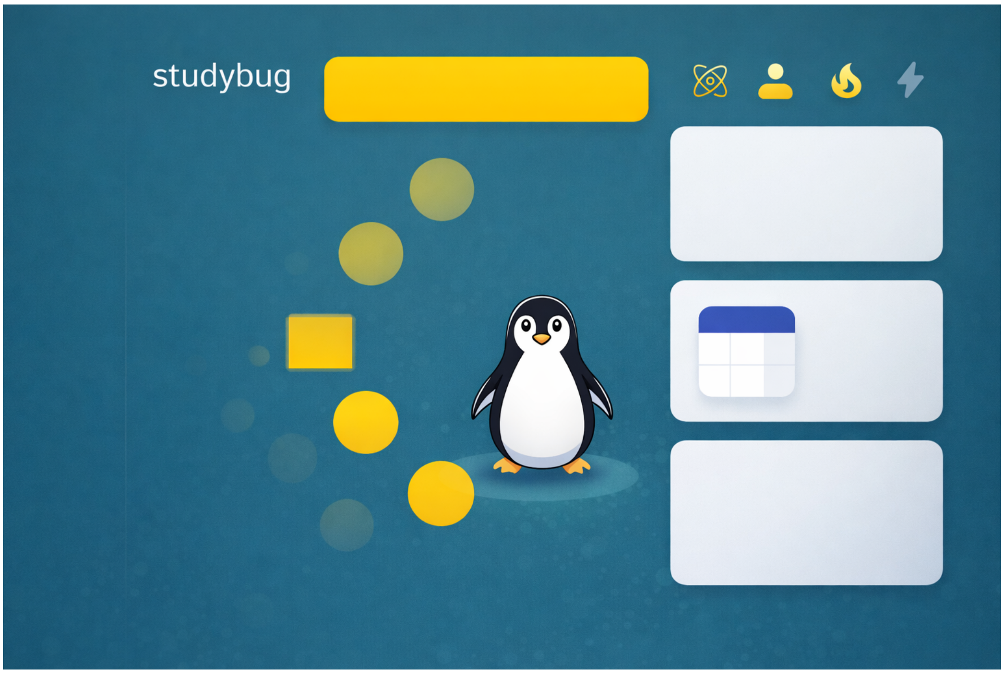

Figure 8.1: Mockup Design main window for desktop App

As per our expectations in our desktop app we intended to develop good UI with smooth selection of desired objects to be detected which unfortunately proved to be inefficient interms of performance so the extra tabs were removed afterwards. Also with some logs and statistics tabs we thought of keeping each and every generated texts and detected objects throughout the program.

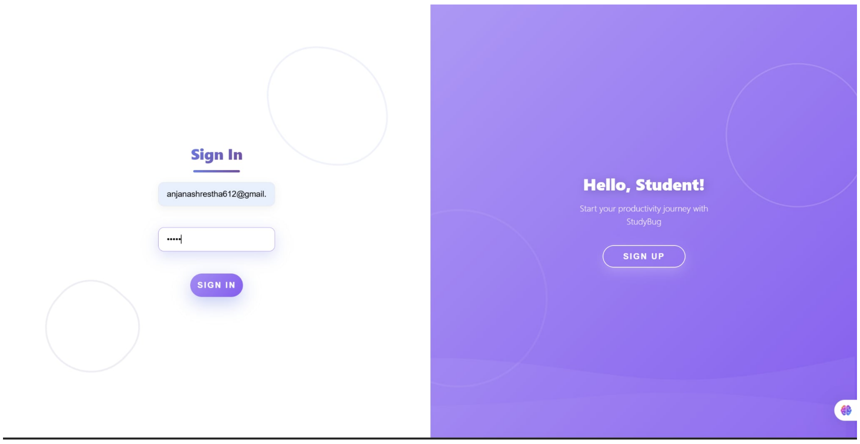

Figure 8.2: Login UI for desktop

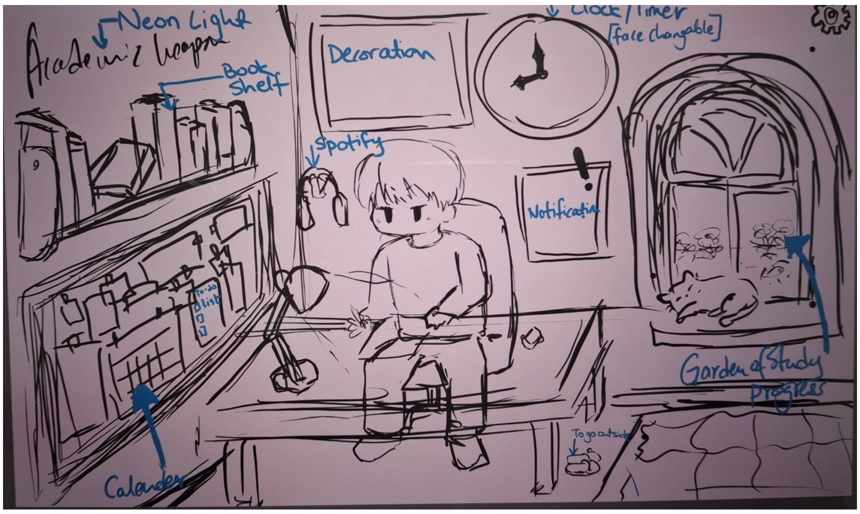

Figure 8.3: Study Room Expected

Chapter 9

Conclusion and Future Enhancements

The proposed aesthetic study website aims to address common challenges faced by students, such as lack of focus, poor scheduling, and low motivation. By combining productivity tools with a visually appealing interface, the system is expected to create a supportive study environment that encourages consistency and effective time management. The project intends to demonstrate that thoughtful UI/UX design can significantly enhance user engagement in educational tools.
As for future enhancements, after the initial implementation, the following features could be added:

- Expansion of theme options, layouts, and visual customization for better user engagement
- Accessibility improvements including screen reader support, keyboard navigation, and high-contrast modes

Appendix

Appendix

Appendix A: Project Overview

This appendix provides additional supporting information related to the proposed development and implementation of the *StudyBug* study website. StudyBug is designed as a student-centric productivity platform aimed at improving focus, time management, and study consistency through an integrated and aesthetically pleasing interface. The appendix complements the main chapters by offering further clarification on design decisions, planned development practices, and intended system usage.

Appendix B: Tools and Technologies to be Used

The StudyBug project will be developed using standard web technologies to ensure accessibility and ease of deployment. HTML, CSS, and JavaScript (React) will be used for frontend development, enabling responsive design across desktop and mobile browsers. Figma will be used during the ideation and prototyping phase to design wireframes and interactive layouts before implementation. GitHub will be used for version control and collaborative development, allowing team members to work on assigned tasks efficiently and maintain code integrity.

Appendix C: Planned Development Workflow

The development process will follow a structured yet flexible approach. Initially, project requirements will be analyzed and tasks will be divided among team members. A shared GitHub repository will be created, and development will be carried out using feature-based contributions. Each team member will implement their assigned components locally and push updates to the repository. Changes will be reviewed and merged into the main branch to ensure smooth integration and consistency across the system.

Appendix D: Intended System Usage

StudyBug will allow users to create and manage study tasks, plan daily or weekly study schedules, and conduct focused study sessions using a built-in timer. The system will record completed tasks and study sessions to provide basic progress tracking. Users will also be able to customize the visual appearance of the website to create a comfortable study environment. Data will be stored locally within the browser, enabling continued usage without mandatory login or backend dependency.

Appendix E: Limitations and Assumptions

The proposed implementation of StudyBug will focus on core study-support features and will not include advanced analytics, cloud synchronization, or collaborative study features in the initial version. The system assumes that users will access the website through a modern web browser with JavaScript enabled. Future versions of the project may address these limitations by incorporating additional features and backend support.

Bibliography

[1] S. Oğut, "Assessing the efficacy of the pomodoro technique in enhancing study focus and productivity: A scoping review," *Journal of Educational Psychology and Learning*, vol. 42, no. 1, pp. 45–62, 2025.

[2] A. Hinze, N. Vanderschantz, C. Timpany, S. J. Cunningham, S.-J. Saravani, and C. Wilkinson, "A study of mobile app use for teaching and research in higher education," *Technology, Knowledge and Learning*, vol. 28, pp. 1271–1299, 2023.

[3] J. Gikas and M. M. Grant, "Mobile computing devices in higher education: Student perspectives on learning with cellphones, smartphones and social media," *The Internet and Higher Education*, vol. 19, pp. 18–26, 2013.

[4] H. Compton and D. Burke, "The use of mobile learning in higher education: A systematic review," *Computers & Education*, vol. 123, pp. 53–64, 2018.