

**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 26, 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 fulfillment 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

The author has agreed that the library, Khwopa College of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for the extensive copying of this project report for scholarly purpose may be granted by supervisor who supervised the project work recorded here in or, in absence the Head of The Department where in the project report was done. It is understood that the recognition will be given to the author of the report and to Department of Computer Engineering, KhCE in any use of the material of this project report. Copying or publication or other use of this report for financial gain without approval of the department and author's written permission is prohibited. Request for the permission to copy or to make any other use of material in this report in whole or in part should be addressed to:

Head of Department
Department of Computer Engineering
Khwopa College of Engineering
Libali,
Bhaktapur, Nepal

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

Certificate of Approval	i
Copyright	i
Acknowledgement	ii
Abstract	iii
Contents	v
List of Figures	viii
List of Symbols and Abbreviation	ix
1 Introduction	1
1.1 Background Introduction	1
1.2 Motivation	1
1.3 Problem Definition	2
1.4 Goals and Objectives	2
1.5 Scope and Applications	2
1.6 Report Organization	2
1.6.1 Introduction	2
1.6.2 Literature Review	2
1.6.3 Feasibility Study	2
1.6.4 Methodology	3
1.6.5 Requirement Analysis	3
1.6.6 System Design and Architecture	3
1.6.7 Block Diagram and Description of Proposed System	3
1.6.8 Expected Outcome	3
1.6.9 Actual Outcome	3
1.6.10 Conclusion and Future enhancements	3
2 Literature Review	4
2.1 Assessing the efficacy of the Pomodoro technique in enhancing anatomy lesson retention during study sessions: a scoping review	4
2.2 Analyzing the Impact of AI Tools on Student Study Habits and Academic Performance	4
2.3 A Study of Mobile App Use for Teaching and Research in Higher Education	5
3 Feasibility Study	6
3.1 Technical Feasibility	6
3.2 Operational Feasibility	6
3.2.1 Focus Timer and Study Session Module	6
3.2.2 Study Scheduling and Planner Module	6

3.2.3	Aesthetic UI, Themes, and Personalization	7
3.3	Economic Feasibility	7
3.4	Scheduling Feasibility	7
4	Methodology	8
4.1	Agile Method as Software Development Model	8
4.2	Figma for Ideation and Prototyping	9
4.2.1	Project Plan	9
4.2.2	Iteration1	9
4.2.3	Iteration2	9
4.3	Overall Phase Followed	10
4.3.1	Planning Phase	10
4.3.2	Development Phase	10
4.3.3	Integration	10
4.4	Task Workflow	11
5	Requirement Analysis	12
5.1	Software Requirement	12
5.2	Hardware Requirement	12
5.3	Functional Requirement	12
5.4	Non-Functional Requirement	13
6	System (or Project) Design and Architecture	14
6.1	Use Case Diagram	14
6.2	Context Diagram	15
6.3	Data Flow Diagram	15
6.4	Workflow Diagram	16
7	Block Diagram and Description of Prepared System	18
7.1	System Block Diagram	18
7.2	Frontend Interface Module	19
7.3	Authentication Module	19
7.4	Backend Server Module	19
7.5	Task Management Module	19
7.6	Study Scheduling Module	19
7.7	Focus Timer Module	19
7.8	Progress Tracking Module	20
7.9	Database Module	20
7.10	System Workflow	20
7.11	Advantages of the Prepared System	20
8	Expected Outcome	21
8.1	Desktop Outcome Expected	21
9	Actual Outcome	23
9.1	Desktop Outcome	23

10 Conclusion and Future Enhancements	24
Appendix	25
Bibliography	27

List of Figures

4.1	Agile Model as Software Development Model	8
4.2	Figma Workflow	9
4.3	Create a shared github repository	11
6.1	Use case Diagram	14
6.2	Context Diagram of Inference System	15
6.3	Data Flow Diagram of Inference System	16
6.4	Sequence Diagram of Inference System	17
7.1	System Block Diagram of StudyBug	18
8.1	Mockup Design main window for desktop App	21
8.2	Login UI for desktop	22
8.3	Study Room Expected	22
9.1	Actual outcome of desktop website	23

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 visually impaired people and the system is able to:

- study scheduling and task planning
- 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 our readers so that they can know about the problems that visually impaired people have to go through and how our purposed system can help them.

1.6.2 Literature Review

The literature review is there to familiarize ourselves with the current state of knowledge on assistance for visually impaired people and ensure us to not repeat what others have already done.

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 voice assistance.

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 ensure that our set goal is achieved and not to repeat what others have done already in this topic.

1.6.9 Actual Outcome

It provides clear view about the accomplishments we have achieved and comparison can be made with expected outcome to note down deviations.

1.6.10 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, Vue, or Angular. 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 Firebase or MongoDB for persistent, login-based access. Secure authentication and deployment using platforms like GitHub Pages, Netlify, or Vercel further support technical feasibility.

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.

nation. 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.

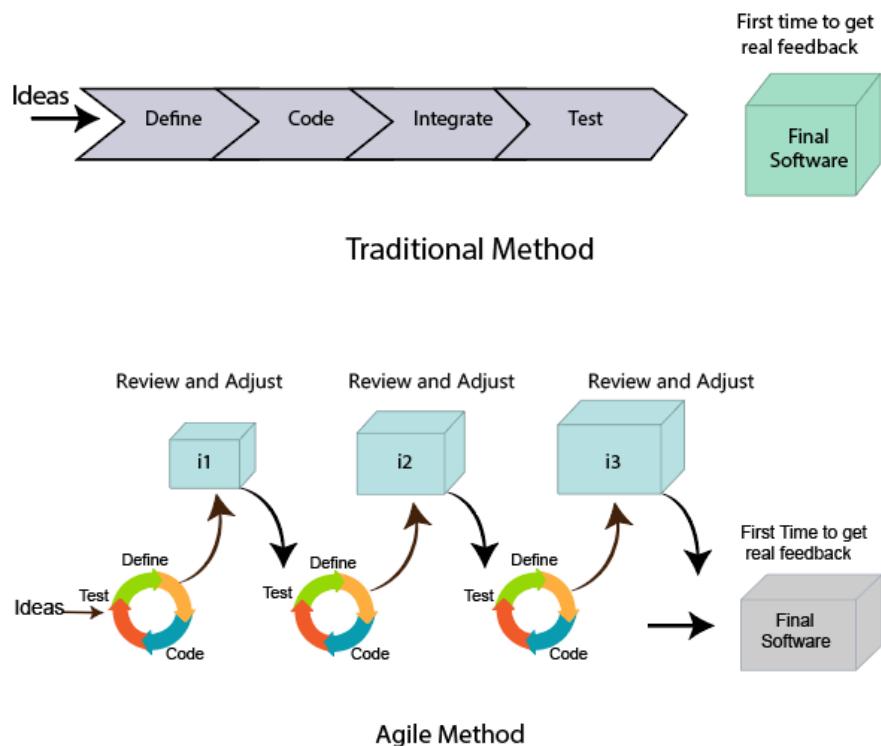


Figure 4.1: Agile Model as Software Development Model

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 was managed in asana in five sessions:

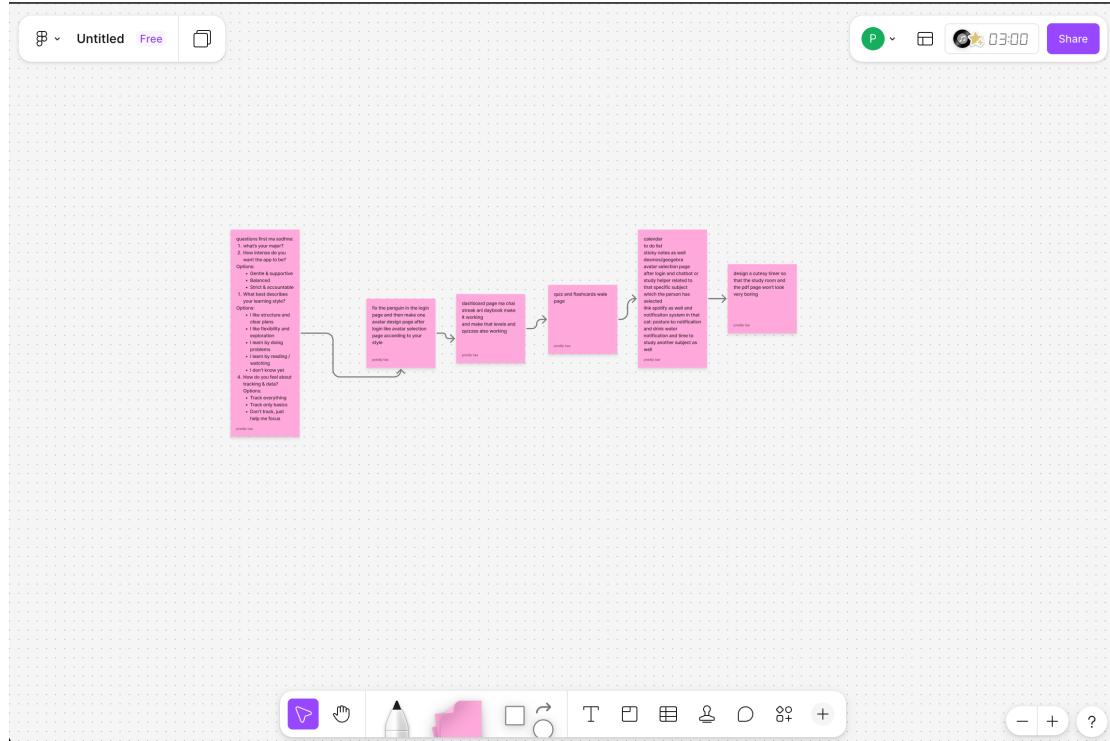


Figure 4.2: Figma Workflow

4.2.1 Project Plan

Planning for further steps were done in detail in this session. All the tasks to be done, their sub tasks, deadline of the tasks and subtasks were recorded here. Then the task was 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 were recorded here including meeting minutes and reports.

4.2.3 Iteration2

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

4.3 Overall Phase Followed

The overall project has been completed in three main phase 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 were well studied and developed. Then each of those developed parts were tested separately.

4.3.3 Integration

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

4.4 Task Workflow

Each tasks of every session were done and recorded in a procedural manner and the programmes were recorded and stored in gitlab.
The workflow of the tasks follow following steps:

1. Get task assigned on the basis of agreement.
2. A shared GitHub repository was created for the StudyBug project.

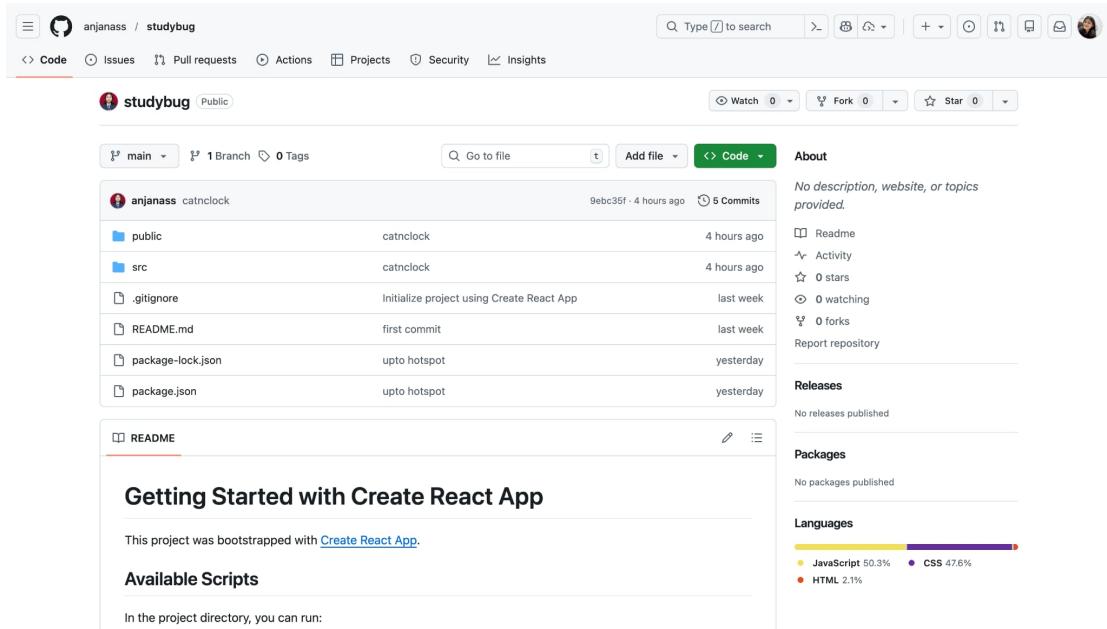


Figure 4.3: Create a shared github repository

3. Each team member worked on their assigned tasks using separate branches.
4. Changes were implemented locally and pushed to the GitHub repository.
5. Completed work was 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 prepared inference system.

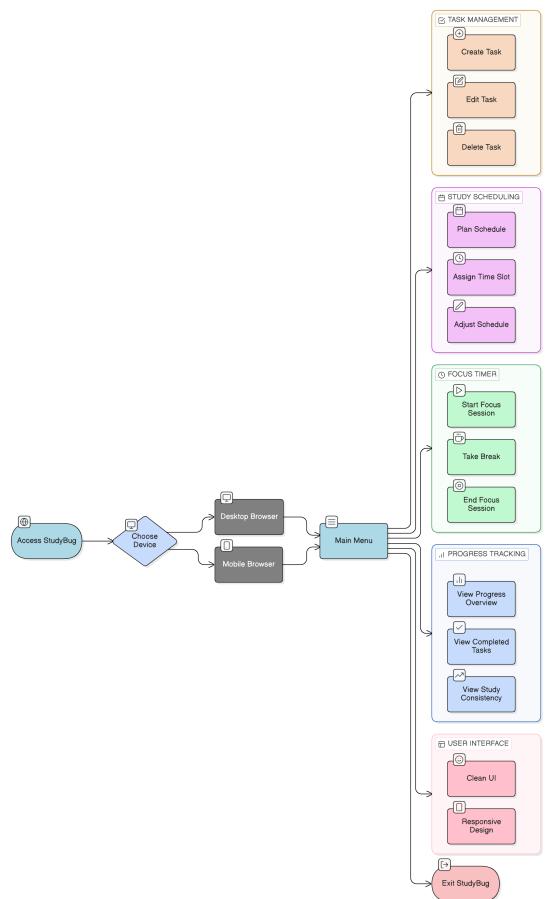


Figure 6.1: Use case Diagram

6.2 Context Diagram

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

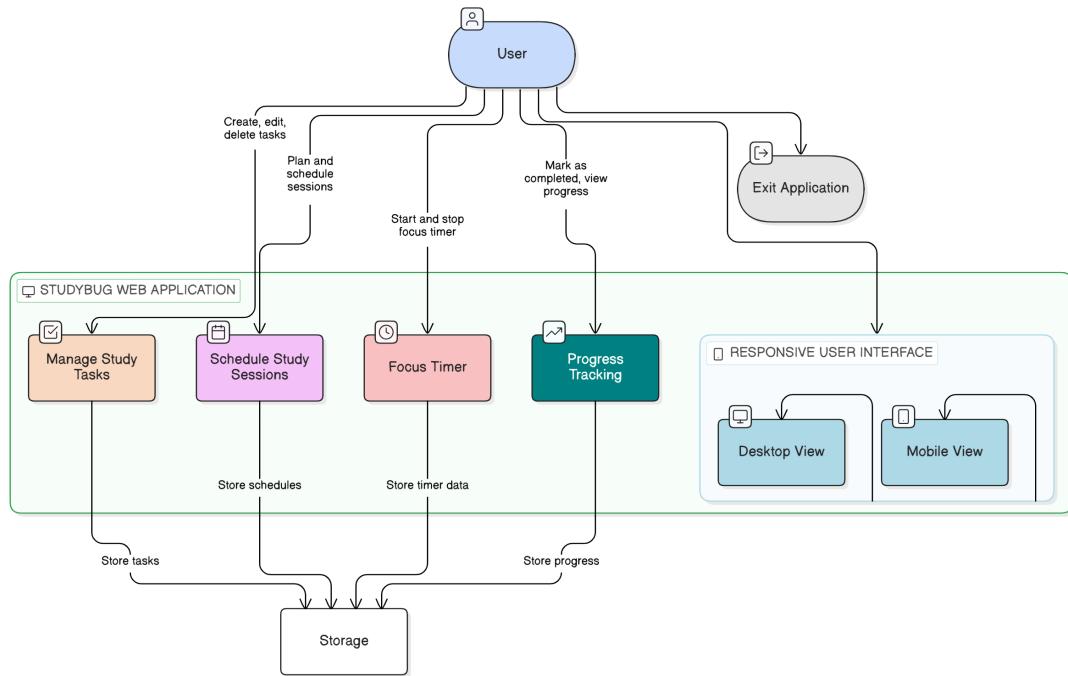


Figure 6.2: Context Diagram of Inference System

6.3 Data Flow Diagram

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

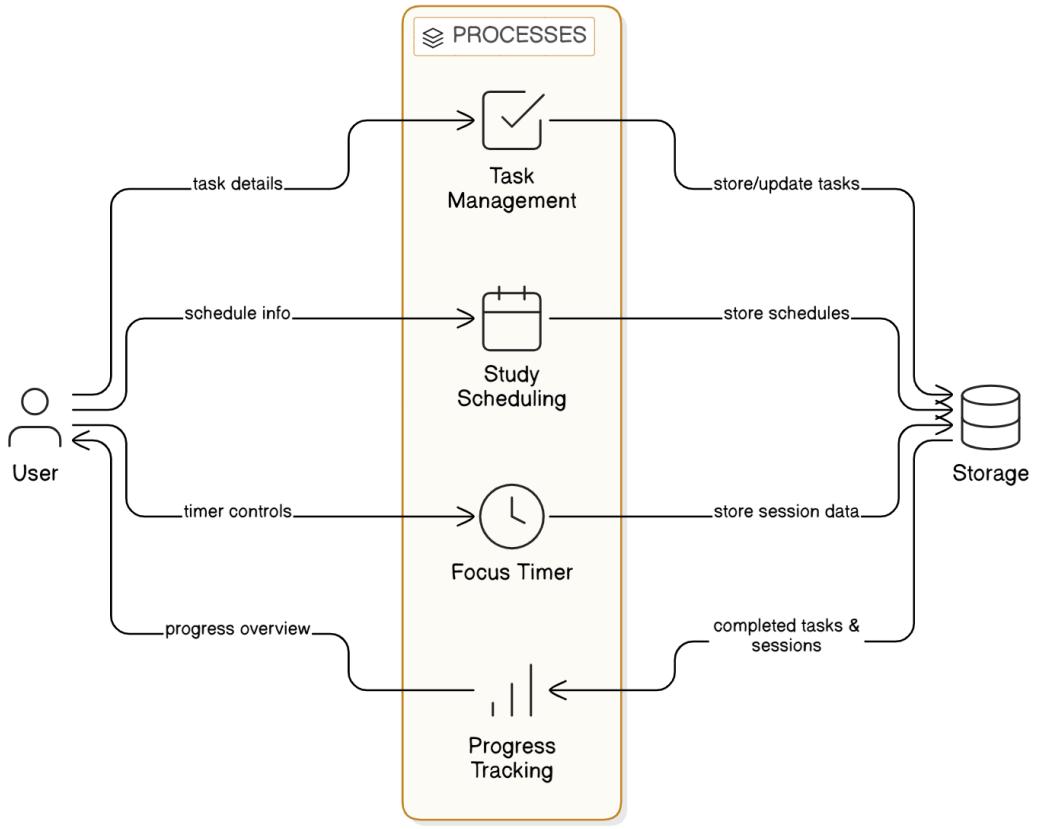


Figure 6.3: Data Flow Diagram of Inference System

6.4 Workflow Diagram

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

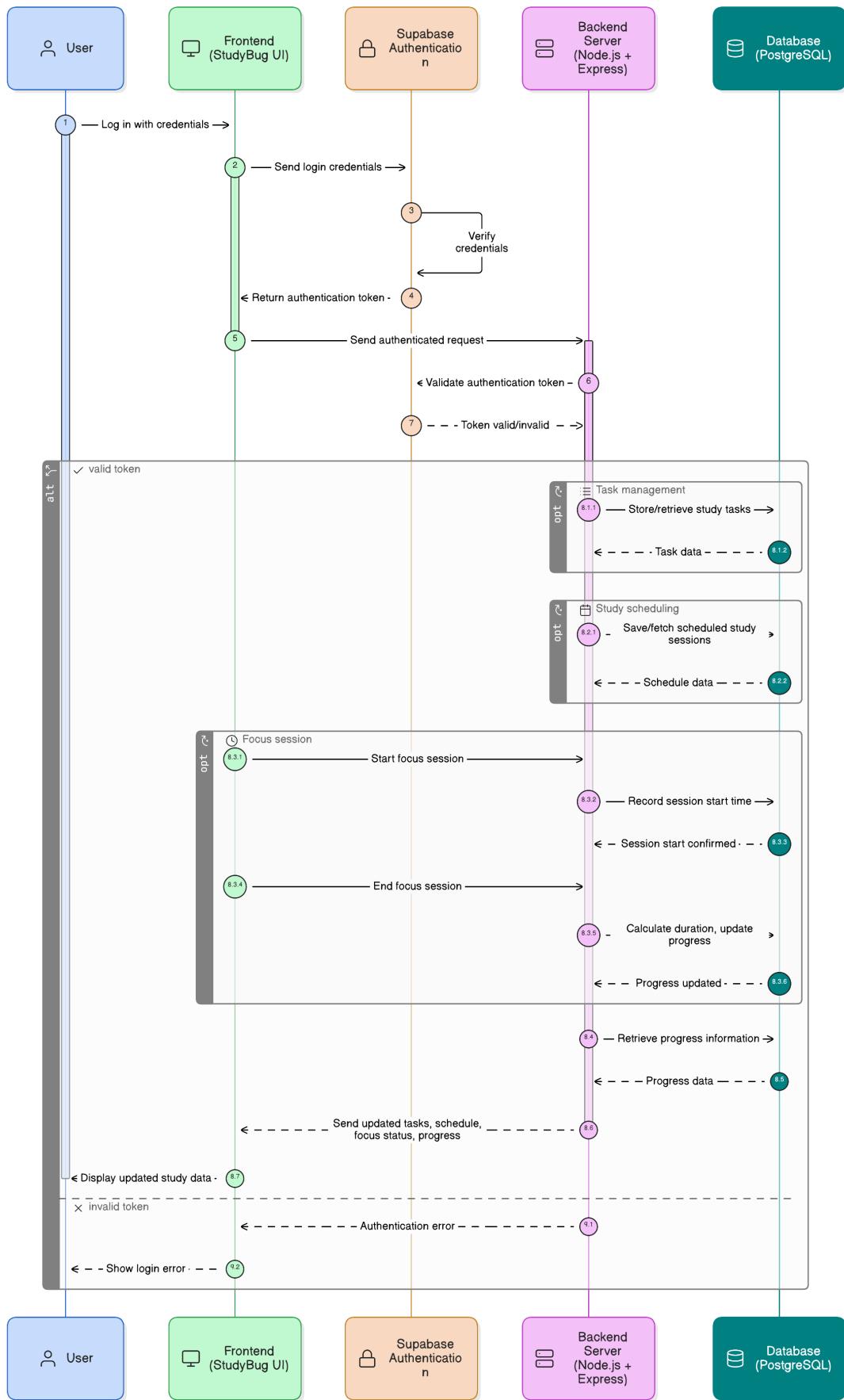


Figure 6.4: Sequence Diagram of Inference System

Chapter 7

Block Diagram and Description of Prepared System

7.1 System Block Diagram

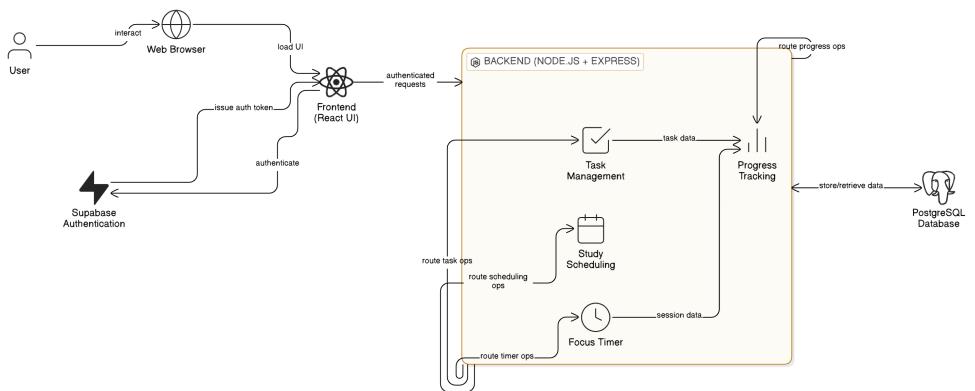


Figure 7.1: System Block Diagram of StudyBug

The system block diagram represents the overall architecture of the *StudyBug* web application and illustrates how different system modules interact to provide an integrated study-support platform. The system follows a client–server architecture and is accessed using a standard web browser.

When a user opens the StudyBug website, the frontend interface loads in the browser and provides access to various study-support features. User actions such as task creation, study scheduling, and focus session initiation are sent from the frontend to the backend server through secure API requests. User authentication is handled through an external authentication service, which ensures secure access to personalized data. All study-related information is stored and managed in a centralized database. The processed data are then 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 is responsible for user interaction and visual presentation. It is developed using modern web technologies and follows responsive design principles to support both desktop and mobile devices. The frontend provides interfaces for task management, study scheduling, focus timer execution, progress tracking, and aesthetic customization. Its primary role is to capture user input, display system output, and communicate with backend services.

7.3 Authentication Module

The authentication module manages user login and access control. It verifies user credentials and issues authentication tokens that allow secure communication between the frontend and backend. This module ensures that each user can access only their own study data while maintaining system security.

7.4 Backend Server Module

The backend server acts as the central control unit of the system. It receives requests from the frontend, processes application logic, and coordinates interactions between different modules. The backend manages task operations, scheduling logic, focus session handling, and progress computation. It also handles data validation and communicates with the database for data storage and retrieval.

7.5 Task Management Module

The task management module enables users to create, update, delete, and mark study tasks as completed. Each task represents a specific academic activity such as assignments, revision topics, or project milestones. Task-related requests are 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 allows users to plan daily or weekly study routines by assigning time slots to specific tasks or subjects. This module retrieves task information from the database and organizes it into structured schedules. The generated schedules are then sent to the frontend for visualization, helping users manage time effectively.

7.7 Focus Timer Module

The focus timer module supports structured study sessions inspired by time-management techniques such as the Pomodoro method. Users can start, pause, and reset focus sessions through the frontend interface. The backend records focus session data, which contributes to overall progress tracking and study analysis.

7.8 Progress Tracking Module

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

7.9 Database Module

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

7.10 System Workflow

The overall system workflow begins when the user accesses the StudyBug website through a web browser. After authentication, the user interacts with the frontend to manage tasks, plan schedules, or initiate focus sessions. The frontend sends these requests to the backend server, which processes the logic and updates the database accordingly. The updated information is then returned to the frontend and displayed to the user, completing the interaction cycle.

7.11 Advantages of the Prepared System

- Provides a centralized and secure platform for study planning and tracking
- Ensures persistent data storage through a backend database
- Supports access across multiple devices and sessions
- Encourages structured study habits using scheduling and focus timers
- Offers 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.

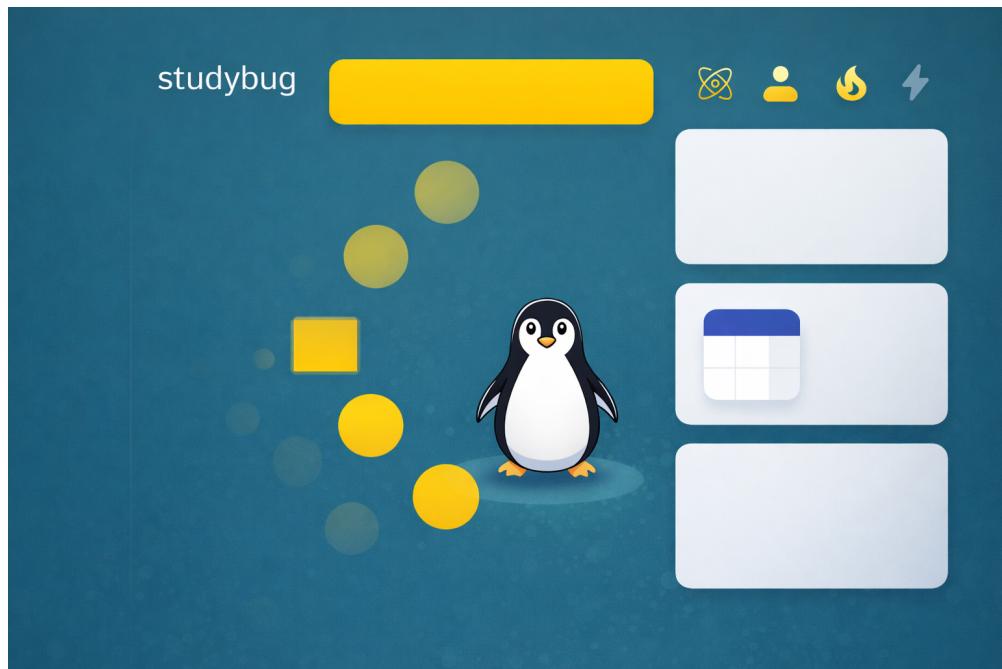


Figure 8.1: Mockup Design main window for desktop App

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 in terms 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.

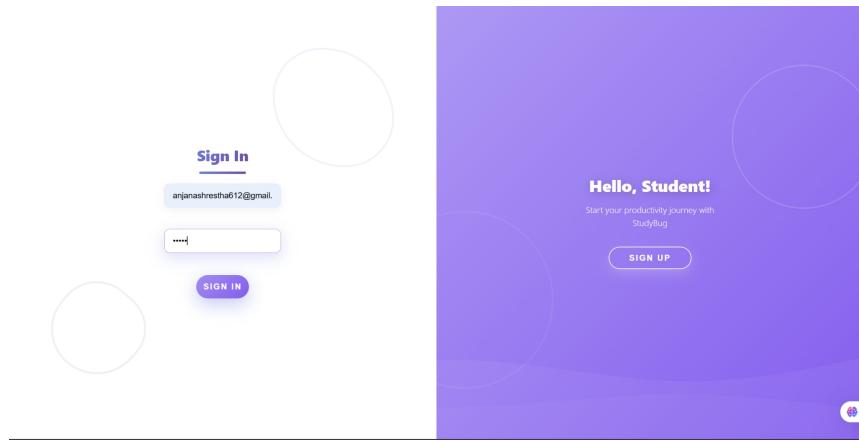


Figure 8.2: Login UI for desktop



Figure 8.3: Study Room Expected

Chapter 9

Actual Outcome

9.1 Desktop Outcome

On desktop platforms, the website performs as expected, providing a smooth and visually pleasing experience. All core features function correctly, and users can efficiently plan, track, and manage their study activities. The aesthetic design contributes to reduced distraction and improved focus during study sessions.

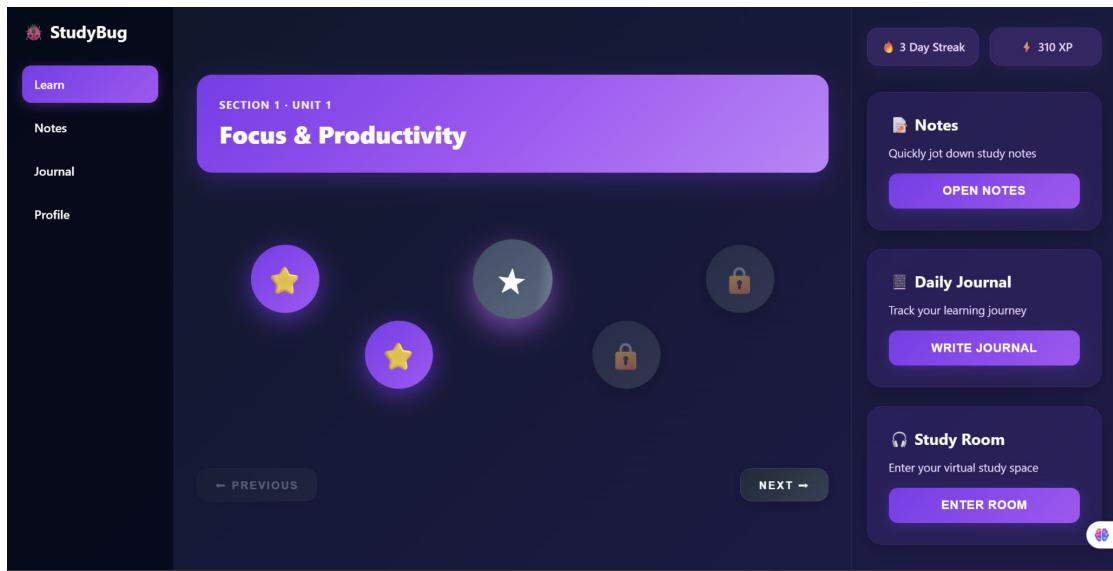


Figure 9.1: Actual outcome of desktop website

Chapter 10

Conclusion and Future Enhancements

The aesthetic study website successfully addresses 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 creates a supportive study environment that encourages consistency and effective time management. The project demonstrates that thoughtful UI/UX design can significantly enhance user engagement in educational tools.

As for Future Enhancements for our project we could implement these:

- 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 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, development practices, and system usage.

Appendix B: Tools and Technologies Used

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

Appendix C: Development Workflow

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

Appendix D: System Usage Description

StudyBug allows 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 records completed tasks and study sessions to provide basic progress tracking. Users can also customize the visual appearance of the website to create a comfortable study environment. Data is stored locally within the browser, enabling continued usage without mandatory login or backend dependency.

Appendix E: Limitations and Assumptions

The current implementation of StudyBug focuses on core study-support features and does not include advanced analytics, cloud synchronization, or collaborative study features. The system assumes that users 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.