LEARNING ROAD MAP SYSTEM

A PROJECT REPORT

Submitted by

KARAN SOOD [RA2211026010016]
PRATHAM SHARMA [RA2211026010022]
ADITYA NAIR[RA2211026010027]
SANKET DHUMAL[RA2211026010020]

Under the Guidance of

Dr. Shiny Angel T S

Assistant Professor, Department of Computational Intelligence

in partial fulfillment of the requirementsfor the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in Artificial Intelligence and Machine Learning

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE ANDTECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2025

Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **B. tech / CSE-AIML**

Student Name : **Karan Sood, Pratham Sharma, Aditya Nair, Sanket Dhumal**

Registration Number : **RA2211026010016, RA2211026010022, RA2211026010027, RA2211026010020**

Title of Work : **LEARNING ROAD MAP SYSTEM**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP107L - Minor Project [18CSP108L- Internship] report titled "**LEARNING ROAD MAP SYSTEM**" is the bonafide work of "**KARAN SOOD [RA2211026010016], PRATHAM SHARMA [RA2211026010022], ADITYA NAIR[RA2211026010027], SANKET DHUMAL[RA2211026010020]"** who carried out the project work[internship] under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Shiny Angel T S

SUPERVISOR
Assistant Professor
Computational
Intelligence

SIGNATURE

DR. R. ANNIE UTHRA

PROFESSOR &HEAD
DEPARTMENT OF
COMPUTATIONAL INTELLIGENCE

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M,** Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. Lakshmi,** Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department Dr. Annie Uthra R, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, Dr. C. Sherin Shibi, Department of Computational Intelligence, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide Dr. Shiny Angel T S, Department of Computational Intelligence, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computational Intelligence, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

Authors

ABSTRACT

In today's fast-paced academic environment, students often struggle with balancing various responsibilities, such as coursework, assignments, extracurricular activities, and personal commitments. This challenge can lead to stress, missed deadlines, and reduced productivity. As a result, effective time management and organization have become essential skills for academic success. This report presents the Learning Roadmap System, a digital platform designed to help students manage their academic tasks, set study goals, and track their progress in a structured yet flexible manner.

The system provides students with a personalized experience by allowing secure access to their data through a sign-in and login system. Central to the application is a to-do list feature, which enables users to add, organize, prioritize, and manage their academic tasks effectively. This feature helps students stay on top of their assignments, deadlines, and study schedules.

To further support students in managing their time, the system includes an AI-powered chatbot that offers reminders, motivational prompts, and helpful study tips. This interactive assistant helps users stay focused, improve their time management skills, and maintain consistent study habits.

The platform is designed to be accessible across various devices, ensuring that students can manage their tasks wherever they are. Additionally, it includes a personalized dashboard that provides real-time feedback on task completion and progress, helping students assess their productivity and make informed decisions to improve their study habits.

Ultimately, the Learning Roadmap System aims to reduce academic stress, enhance time management, and help students achieve their academic goals. By combining effective organization with motivational support, the platform provides a comprehensive solution for students striving to improve their study habits and academic performance.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction:

In today’s educational landscape, students face a variety of challenges when it comes to managing their academic responsibilities. With multiple assignments, projects, exams, and extracurricular commitments, it becomes increasingly difficult for students to balance all aspects of their academic life effectively. This often leads to feelings of stress, frustration, and burnout. The need for better time management and organizational skills has never been more critical to academic success. Without a clear structure, students are at risk of falling behind, missing deadlines, or not reaching their full academic potential.

As a response to this challenge, the Learning Roadmap System has been designed to provide students with a comprehensive and intuitive platform that helps them manage their academic tasks efficiently. The goal of the system is to offer a structured yet flexible solution for students to plan, track, and achieve their study goals in a way that reduces stress and enhances productivity.

This system includes several key features, such as a task management tool, a personalized dashboard, and an AI-powered virtual assistant that offers support, motivation, and timely reminders. The combination of these features allows students to stay organized, improve their time management skills, and maintain focus on their academic objectives.

The purpose of this chapter is to provide an overview of the challenges students face in managing their academic tasks, explain the need for a solution like the Learning Roadmap System, and highlight the goals and objectives of the project. This chapter will also outline the structure of the report, which will provide detailed information on the system's features, development process, and future enhancements.

1.2 Motivation

In today's fast-paced academic environment, students are often overwhelmed with the increasing demands placed upon them. From attending classes, completing assignments, and preparing for exams to participating in extracurricular activities and managing personal life, the balance can be difficult to maintain. This overwhelming load often leads to stress, missed deadlines, poor time management, and, in some cases, burnout.

Effective time management is a crucial skill that can significantly impact a student's academic success. However, many students struggle with organizing their time and staying focused amidst their numerous commitments. While some students may rely on traditional methods like paper planners or digital calendars, these tools often lack the flexibility and personalized support needed to help students stay on track and motivated.

The Learning Roadmap System was conceived to address these challenges. Its development is motivated by the desire to create a tool that not only helps students organize their tasks but also provides ongoing support to enhance their study habits, reduce stress, and improve productivity. By integrating features such as task management, personalized reminders, motivational prompts, and data-driven insights, the platform aims to empower students to take control of their academic schedules and optimize their performance.

This system seeks to bridge the gap between traditional time management tools and the evolving needs of students in the digital age. The motivation behind this project is to provide a solution that not only simplifies task organization but also provides personalized guidance to help students build healthier, more effective study routines and improve their overall academic experience.

1.3 Sustainable Development Goal of the Project

The Learning Roadmap System directly supports the United Nations' Sustainable Development Goal 4 (SDG 4): "Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all." This platform addresses key challenges faced by students in managing their academic responsibilities and enhances the overall learning experience through structured planning, personalized support, and accessible digital tools.

By incorporating AI-powered features such as a virtual assistant for time management and motivational support, the system ensures that students from diverse backgrounds can receive personalized guidance tailored to their individual learning styles and needs. This helps remove traditional barriers to effective education such as lack of structure, poor time management, or limited access to personal academic support.

The platform promotes inclusivity and lifelong learning by empowering students to take control of their study habits and academic progress. Features such as the to-do list, progress-tracking dashboard, and smart reminders make it easier for learners to stay consistent and focused, regardless of their environment or circumstances.

Furthermore, the application has the potential to evolve into a collaborative educational space, where peer support, resource sharing, and community engagement can further enrich the learning process. This fosters a culture of continuous improvement and shared knowledge, aligning with SDG 4's goals of inclusive and equitable education.

In essence, the Learning Roadmap System contributes to building a sustainable, tech-driven educational ecosystem. It equips students with the tools and insights necessary for academic success while promoting broader social goals of equity, accessibility, and community-centered growth.

1.4 Product Vision Statement

1.4.1 Audience:

Primary Audience:

Students at high school, college, and university levels who seek to improve their study habits, stay organized, and boost academic performance through structured digital tools.

Secondary Audience:

Educators, tutors, and academic professionals who aim to support students in organizing their workload, tracking progress, and enhancing learning efficiency.

1.4.2 Needs:

Primary Needs:

Effective management and organization of academic tasks and study schedules.

Personalized study planning based on course demands, learning style, and deadlines.

Smart reminders and motivational prompts to stay consistent with academic goals.

Secondary Needs:

Progress tracking for assignments, exams, and study goals.

Integration with document-sharing and note-taking platforms (e.g., Google Drive, OneNote).

Collaboration tools to support group studies and peer learning activities.

1.4.3 Products:

Core Product:

A comprehensive AI-powered learning roadmap application that helps students structure, schedule, and track their academic tasks and goals efficiently.

Additional Features:

Calendar synchronization with academic deadlines and events.

Intelligent notifications and reminders tailored to user behavior and study patterns.

Study resource sharing and platform integration for seamless content management.

Study session tools including Pomodoro timers and customizable focus sessions.

AI chatbot assistant offering time management tips and motivational support.

1.4.4 Values:

Core Values:

Productivity: Enabling users to manage their time and responsibilities effectively.

Personalization: Adapting the experience to suit each user's learning style, goals, and performance.

Support & Collaboration: Encouraging peer interaction and providing helpful feedback mechanisms to build a supportive learning environment.

Differentiators:

Smart Scheduling: AI-driven suggestions for study times based on user habits and energy levels.

Seamless Integration: Smooth compatibility with calendars, task managers, and productivity tools.

Motivational Tools: Gamified learning features such as challenges, rewards, and progress tracking to maintain engagement and drive.

1.5 Product Goal

The primary goal of the Learning Roadmap System is to empower students with a digital platform that simplifies academic planning, enhances productivity, and supports effective time management. The application is designed to help users stay organized, motivated, and focused on their academic objectives through a personalized, intelligent, and accessible interface.

By integrating AI-powered features such as smart scheduling and a virtual assistant, the system aims to reduce academic stress and improve learning outcomes. It supports students in managing their coursework, tracking progress, and maintaining consistency in their study routines.

In the long term, the platform seeks to evolve into a collaborative learning environment where students can also interact with peers and educators, share study resources, and engage in group projects. Ultimately, the product aspires to become an essential academic companion that promotes lifelong learning, self-discipline, and academic excellence.

1.6 Product Backlog

Table 1.1 Product Backlog

S.No	User Story
#US 1	As a new user, I want to easily register and log in to the platform so that I can securely access personalized study tools and features.
#US 2	As a user, I want to create and customize my personal study profile so that I can manage my academic goals, courses, and learning preferences.
#US 3	As a user, I want to add and organize academic tasks in a to-do list so that I can effectively manage my study schedule and meet deadlines.
#US 4	As a user, I want to receive smart reminders and motivational prompts from the AI chatbot so that I can stay consistent and encouraged in my studies.
#US 5	As a user, I want to track my progress on assignments and study sessions so that I can monitor my improvement and stay on top of my goals.
#US 6	As a user, I want to sync my academic calendar with the platform so that I never miss important deadlines for exams, assignments, or classes.
#US 7	As a user, I want to set study goals and break them into smaller tasks so that I can manage my workload more effectively.
#US 8	As a user, I want to use customizable study session templates (like Pomodoro timers) so that I can maximize my focus and productivity during study time.
#US 9	As a user, I want to share study resources and notes with peers so that we can collaborate and support each other's learning.
#US 10	As a user, I want the platform to provide study tips and personalized suggestions based on my performance data so that I can improve my learning strategy.
#US 11	As a user, I want to give feedback on platform features and study tools so that I can contribute to improving the overall experience for myself and others.

The product backlog of the Learning Roadmap System was configured using the Microsoft Planner Agile Board, as illustrated in Figure 1.1. The backlog includes all user stories relevant to the application's development. Each user story is defined with essential parameters such as MoSCoW prioritization, clearly outlined functional and non-functional requirements, detailed acceptance criteria, and linked tasks to guide implementation and track progress effectively.

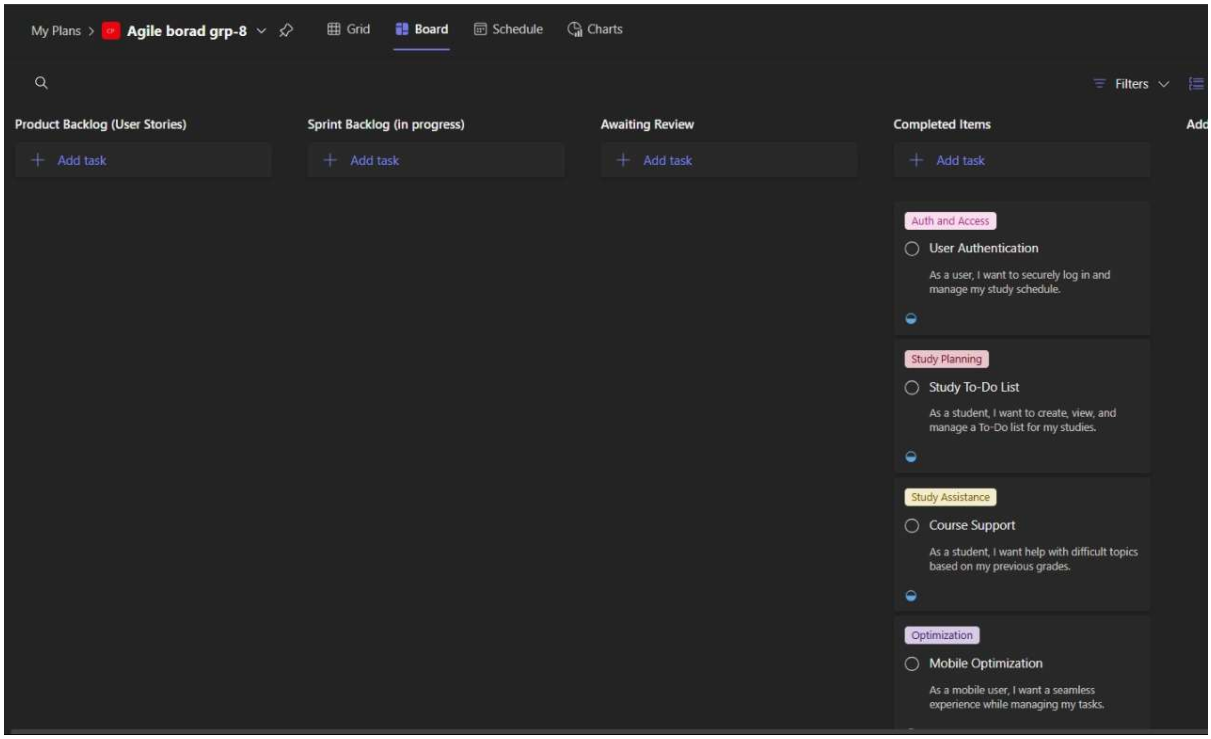


Figure 1.1 MS Planner Board of Ai E-learning Application

1.7 Product Release Plan

The following Figure 1.2 depicts the release plan of the project

Figure 1.2 Release plan of Learning Roadmap System

CHAPTER 2

SPRINT PLANNING AND EXECUTION

2.1 Sprint 1

2.1.1 Sprint Goal with User Stories of Sprint 1

The goal of Sprint 1 is to develop the core foundational elements of the application, including the user authentication system, profile creation, and task management interface to help students organize their academic workload effectively.

The following Table 2.1 outlines the detailed user stories included in Sprint 1:

Table 2.1 Detailed User Stories of sprint 1

S.No	Detailed User Stories
US #1	As a new user, I want to register and log in to the platform so that I can securely access my personalized learning dashboard.
US #2	As a user, I want to create and update my profile so that the application can provide personalized task and study recommendations.
US #3	As a user, I want to add, edit, and delete tasks in my to-do list so that I can organize my academic schedule and manage my workload effectively.

2.1.2 Functional Document

2.1.2.1. Introduction

The Learning Roadmap System is an AI-assisted educational platform designed to enhance academic success by organizing students' study tasks, schedules, and goals in a personalized and collaborative environment. It empowers learners through smart scheduling, real-time progress tracking, and integration with collaboration tools, while providing educators with insights to support student learning.

2.1.2.2. Product Goal

The primary goal of this project is to build a digital productivity assistant tailored for students that enables effective time management, academic tracking, and community-based study support. The platform is designed to:

Provide smart, personalized study plans based on user input and course requirements.

Track progress on academic tasks such as assignments and exams.

Enable collaboration through shared study sessions and resource sharing.

Integrate with calendar, note-taking, and document-sharing tools.

2.1.2.3. Demography (Users, Location)

Users:

Target Users: High school and college students, tutors, academic mentors, and peer collaborators.

User Characteristics: Varying academic goals, time-management challenges, and collaboration needs.

Location:

Target Location: Global, with a focus on digitally-connected students in academic institutions using e-learning and productivity tools.

2.1.2.4. Business Processes

User Registration and Authentication:

Users can sign up using email or institutional login.

Authentication ensures personalized and secure access to learning dashboards.

Study Plan & Task Management:

Users input course-related deadlines and learning preferences.

The system generates a personalized roadmap with milestones, reminders, and time slots for focused study.

Integration with platforms like Google Calendar, OneNote, and Drive enables seamless task handling and note sharing.

2.1.2.5. Features

Feature 1: User Registration

Description:

Allows secure onboarding of users to access personalized study dashboards.

User Story:

As a new user, I want to register so that I can start building my study schedule and tracking academic goals.

Feature 2: Profile Creation

Description:

Users can build profiles reflecting academic strengths, study habits, and areas of focus to tailor their learning journey.

User Story:

As a user, I want to set up my profile with courses and interests to receive relevant study recommendations.

Feature 3: Enhanced Search Functionality

Description:

Enables users to search tasks, shared study sessions, and notes using filters like subject, priority, or deadline.

User Story:

As a user, I want to quickly find my tasks and relevant resources using search and filtering options.

2.1.2.6. Authorization Matrix

Table 2.2 Access level Authorization Matrix

Role	Access Level
Administrator	Full access to user account management, academic data control, and system settings.
Mentor/Tutor	Access to monitor learner progress, provide feedback, and manage shared resources.
Student/Learner	Access to personalized dashboards, task management tools, and study collaboration features.
Guest User	Limited access to explore the platform overview, demo features, and public help guides.

2.1.2.7. Assumptions

- The AI scheduling and progress-tracking modules will rely on academic input data provided by users (e.g., course deadlines, study goals).
- The development team will maintain continuous access to Firebase and other cloud services for development and deployment.
- Feedback from end-users (students and educators) will be collected regularly during each sprint review.
- The platform will comply with international data privacy standards (e.g., GDPR), ensuring encrypted data storage and secure authentication mechanisms.
- Platform usage will occur predominantly through modern browsers and internet-enabled devices, assuming internet connectivity.

2.1.3 Architecture Document

2.1.3.1. Application

Microservices:

The Learning Roadmap System is designed using a microservices-based architecture to ensure scalability, modularity, and independent service management. Each service is loosely coupled and communicates through APIs, which enables efficient deployment and maintenance. The major microservices include:

- Authentication Service:
  Responsible for managing secure user login, registration, session management, two-factor authentication, and password recovery. It ensures users can access their personalized dashboards safely.
- Task and Study Planner Service:
  Handles the creation, scheduling, and tracking of study tasks and goals. Integrates AI to recommend optimal study plans and reminders based on user activity and academic deadlines.
- User Profile & Role Management Service:
  Manages user profile data and role-based access (e.g., Student, Mentor, Administrator). Ensures features and content are tailored to the user's role and permissions.
- Notification Service:
  Sends real-time alerts and reminders regarding upcoming tasks, deadlines, study goals, and motivational prompts. Integrates with email and in-app messaging.
- Dashboard and Analytics Service:
  Provides users with insights into their study habits, progress tracking, and performance metrics. Visualizes data using charts and graphs for actionable feedback.
- Chatbot and AI Assistance Service:
  This service offers intelligent virtual assistant features including study tips, motivational quotes, reminders, and responses to user queries.

  This modular structure ensures flexibility, easy maintenance, and the potential for future expansion, such as integration with third-party educational platforms or tools.

Architecture Diagram

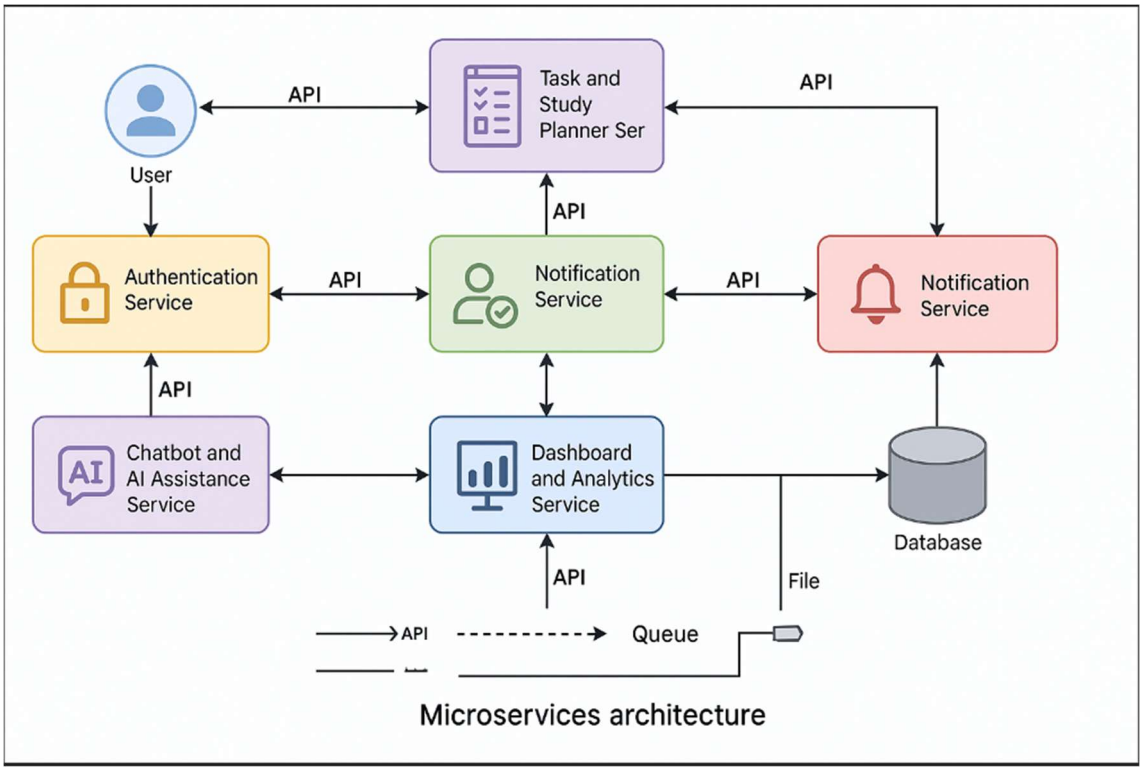


Figure 2.1 Architecture Diagram

Architecture Diagram Description

The Learning Roadmap System uses a microservices architecture to ensure scalability, flexibility, and easier maintenance. Each service operates independently and communicates primarily through REST APIs, while asynchronous tasks use message queues. File-based exports are also supported.

Main Components:

User Interface:
The entry point for all users (students, mentors, admins), interacting with backend services via APIs.

Authentication Service:
Manages user login, registration, sessions, two-factor authentication, and secure access control.

Task and Study Planner Service:
Schedules and tracks study tasks and goals, using AI to recommend personalized study plans.

Notification Service:
Sends alerts and reminders through API-based real-time messaging and optional integration with external platforms.

Dashboard and Analytics Service:
Visualizes progress and study metrics using charts and graphs, aiding in self-assessment and planning.

Chatbot and AI Assistance Service:
Provides a virtual assistant for study tips, reminders, and responses to user queries, powered by AI.

Database:
Stores user profiles, tasks, grades, chatbot logs, and performance data, accessed by relevant services.

Data Exchange Modes:

- API-Based: For real-time tasks like chatbot replies and planner updates.
- Queue-Based: For asynchronous processing, such as grade analysis or bulk notifications.
- File-Based: For exporting reports or study plans in PDF/CSV format.

This design ensures modularity and smooth integration of future services like third-party educational tools.

2.1.3.2 ER Diagram

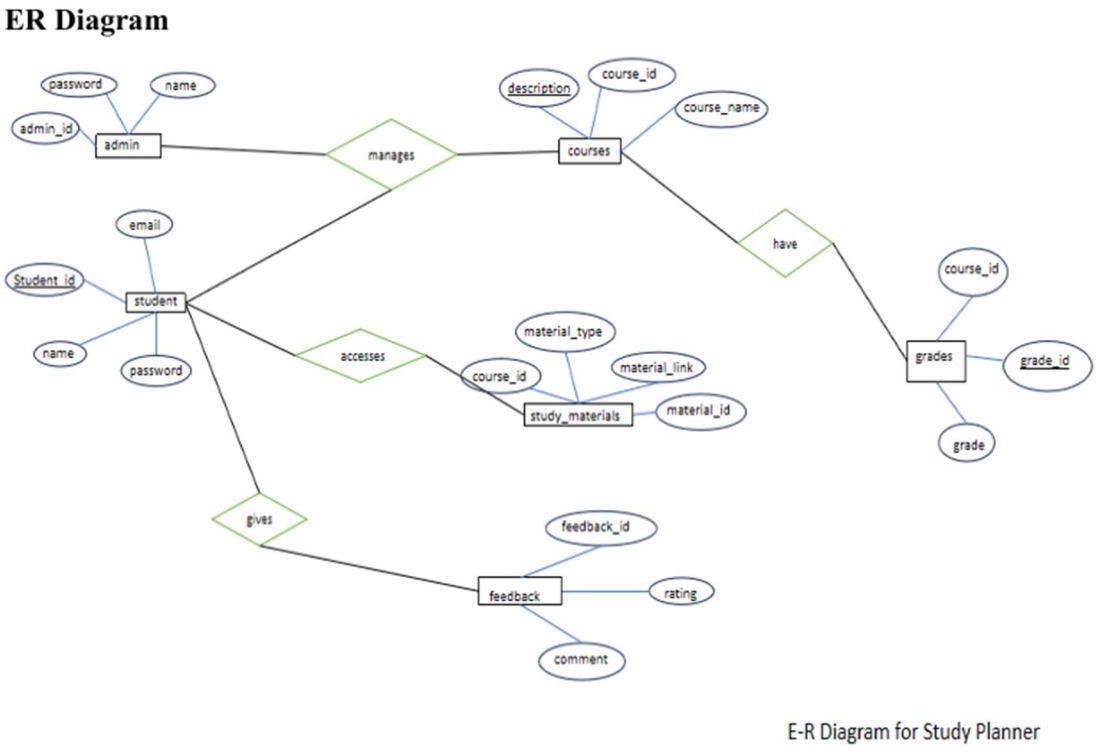


Figure 2.4 System ER Diagram

2.1.3.3. Data Exchange Contract:

Frequency of Data Exchanges:

The data exchange between various services (microservices, database, external systems) occurs at different intervals:

Real-time exchanges for tasks like chatbot responses, grade analysis, and to-do list updates.

Daily updates for student performance analysis and progress reporting.

Data Sets

Student Data: Includes student profiles, to-do lists, and grade records.

Course Data: Includes study materials, course details, and subject breakdowns.

Grade Data: Includes historical and recent grades for each course the student is enrolled in, used for focus area analysis.

Chatbot Interaction Data: Includes the queries students ask and the chatbot's responses, stored for future reference and improvement.

Mode of Exchanges (API, File, Queue, etc.)

API Exchanges:

REST APIs are used for data exchange between frontend and backend services

Queue-based Exchanges:

Event-driven components (e.g., grade analysis) use message queues to handle asynchronous processing of data.

File-based Exchanges:

Exporting progress reports, study plans, and grades can be done via downloadable CSV or PDF files.

2.1.4 UI DESIGN

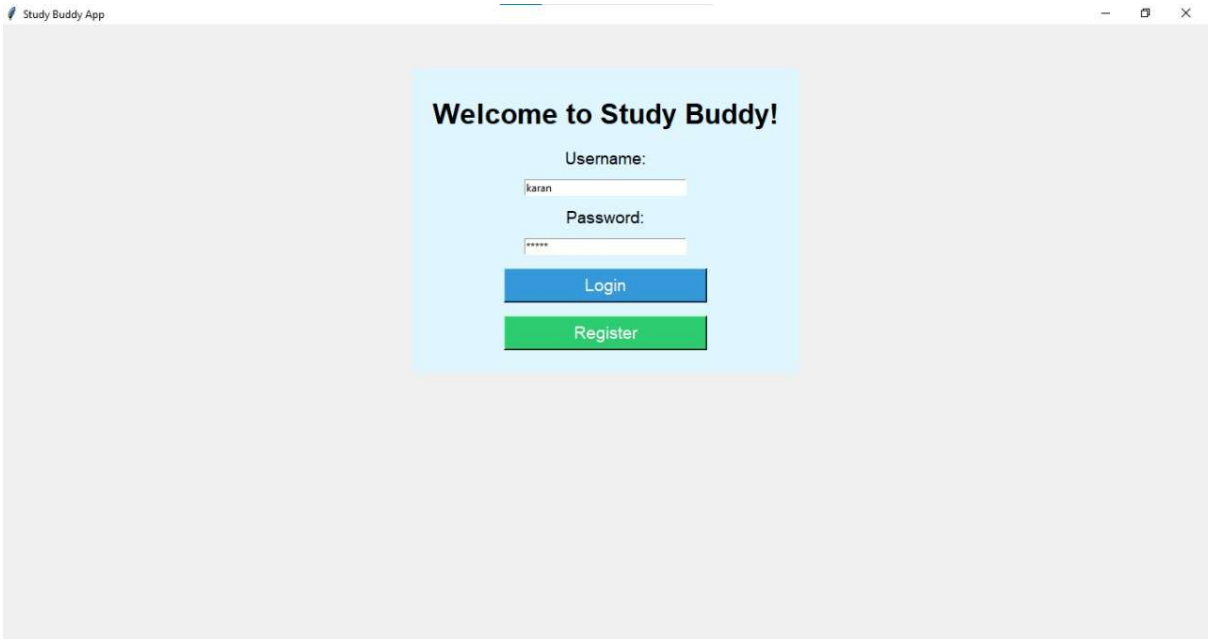
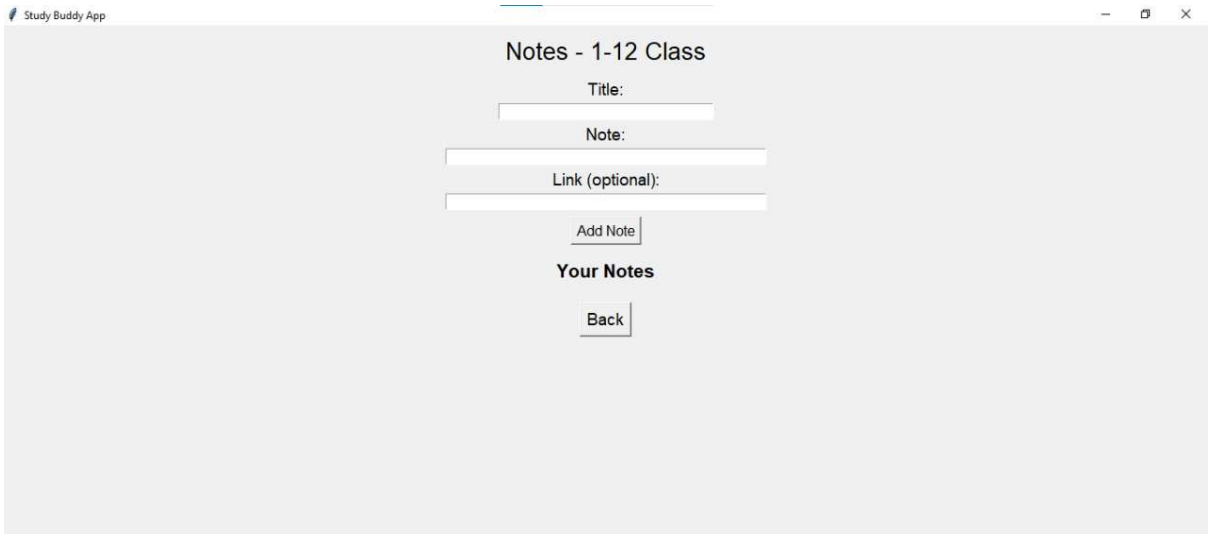


Figure 2.5 UI design for login page

Figure 2.6 UI design for selection page



2.1.5 Functional Test Cases

Table 2.3 Detailed Functional Test Case

	Functional Test Case Template					
Fecture	Test Case	Steps to execute test case	Expected Output	Actual O	Status	More Information
er Registration	Valid User Proiistquizon	1. Open the registration page 2. Enter username. email 3. Click on "Register"	User should be succesfuly registered and redirected to the welcome page	Pass	Pass	Verify email confirmation is shown for invalid input atsplayed
rofile Creation	Varnoi-l Profile Update	1. Log in to rhe aplication 2. Navigate to profile page 3. Enter skills, interests, acilidev-ments, aa.vapofile	Profile is updated and saved correctly	Pass	Pass	Ensure validation messages are shown for invalid inputs
ourse Search	Create 5 kill Listing	1. Go to search bar 2. Enter skill in flo (et."Pytthon") 3. Clich "Post"	Listing is posted and visible to others	Pass	Pass	Results should reflect keyword relevance
Real-Time Interaction	Join Peer Session	1. Join on a seheduled peer session 2. Confirm join prompt 3. Enter live interactioion window	User joinsa live session with peers	Pass	Pass	Verify audio/videco rsedsplayed feedback
Sentiment Analysis	Emotion Detection Active	1. Start video call 2. Speak and observe reactions 3. Monitor system feedback	Real-time emotion analysis shows visual/actie	Pass	Pass	Ensure privacy notices are displayed
lback Mechanism	Submit Feedback	1. Navigate to Feedback page 2. Select area (course, UI, content)	Feedback submitted and confirmation shown	Pass	Pass	System acknowledge and store feedback

2.1.6 Daily Call Progress

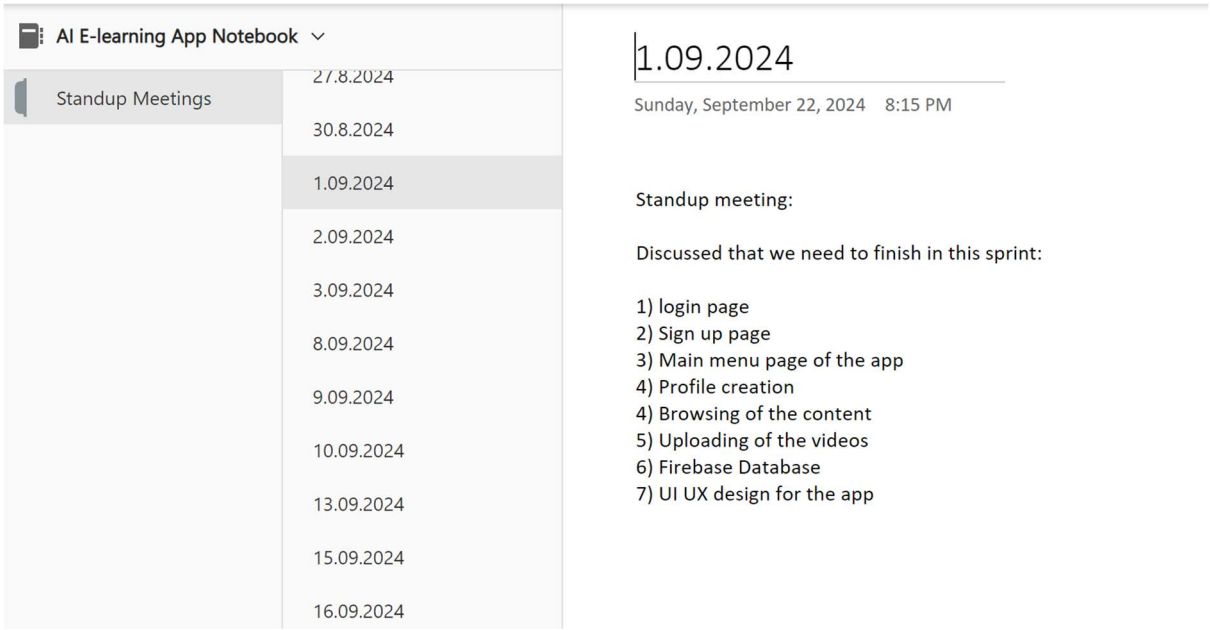


Figure 2.7 Standup meetings

2.1.7 Committed Vs Completed User Stories

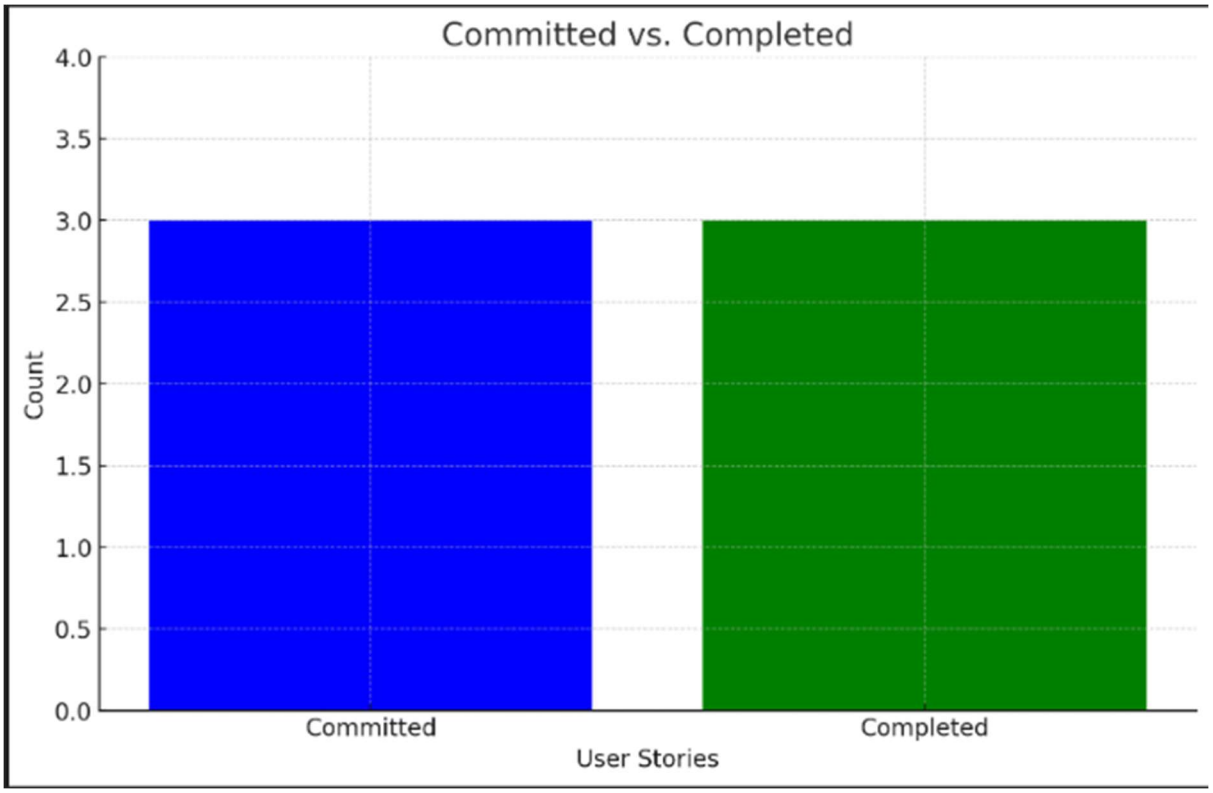


Figure 2.8 Bar graph for Committed Vs Completed User Stories

2.1.8 Sprint Retrospective

Sprint retrospective

Liked	Learned	Lacked	Longed For
Positive collaboration among team members, especially during feature development.	Ensured the website remains responsive across devices after discovering early mobile optimization challenges.	Clarity on the feedback mechanism and its integration with personalized study recommendations.	More detailed sprint planning discussions to address multi-language integration earlier.
Timely completion of user authentication and study to-do list functionality.	Learned the importance of prioritizing multi-language support to reach a broader audience.	More detailed test cases for user study tasks were lacking, resulting in rework after QA.	Better performance monitoring tools to ensure smooth experience across different network conditions.
The feature that highlights areas needing more focus based on student grades worked well.	Discovered how to use data analytics to provide personalized study tips and resources efficiently.	Proper documentation for the implementationteps, which led to delays in certain features.	Enhanced communication between the front-end and back-end teams for smoother integration of authentication features.
Smooth communication with stakeholders about the business case.	Realized the need for more automated testing for feature stability.	Had limited availability of performance monitoring tools for testing under real-world conditions.	More regular feedback sessions to gauge the progress of features like the to-do list management.
Effective implementation of the course support system.	Improved understanding of front-end performance optimization for better UX.	Integration issues between the study planner and grade tracking systems.	Additional resources to test and improve mobile responsiveness across different devices.
Clear delegation of tasks which helped speed up progress.	Gained insights into simplifying the course support module.	Lack of early feedback from students regarding user interface changes.	More user feedback during the sprint to improve feature accuracy.
Quick turnaround on mobile optimization.	Learned that weekly feedback from students helped improve the user interface.	Insufficient time allocated for user testing before feature releases.	More frequent collaboration with the stakeholders.
Detailed feedback from stakeholders was incorporated seamlessly.	Understood the importance of balancing between new features and performance improvements.	Absence of a detailed checklist for cross-browser testing.	Increased focus on automation to speed up manual tasks.

Figure 2.9 Sprint Retrospective for the Sprint 1

2.2 SPRINT 2

2.2.1 Sprint Goal with User Stories of Sprint 2

Sprint 2 Goal:

To enhance user engagement and productivity by integrating AI-powered assistance and personalized study planning tools. This sprint focuses on implementing the chatbot assistant, enabling smart reminders, and introducing progress tracking features to support effective study habits.

Table 2.1 Detailed User Stories for Sprint 2

S.No	Detailed User Stories
US #4	As a user, I want to interact with an AI-powered chatbot so that I can receive time management tips, motivational prompts, and study guidance.
US #5	As a user, I want to receive intelligent reminders for tasks and deadlines so that I stay on track with my academic schedule.
US #6	As a user, I want to view a progress dashboard so that I can monitor my study habits, completed tasks, and upcoming deadlines.
US #7	As a user, I want to synchronize my calendar with academic deadlines and events so that I have a unified view of my responsibilities.

2.2.2 Functional Document

2.2.1 Introduction

The Learning Roadmap System continues development in Sprint 2 with the goal of expanding its core functionality. After establishing user authentication, profile creation, and basic task management in Sprint 1, this sprint focuses on enhancing productivity and personalization through AI-powered features, smart reminders, calendar integration, and progress tracking. These additions aim to support students in developing consistent study habits and maintaining motivation throughout their academic journey.

2.2.2 Sprint Goal

To implement intelligent features that assist students with real-time guidance, automated task reminders, integrated scheduling, and actionable performance insights, enhancing their academic efficiency and organization.

2.2.3 Functional Requirements

Feature 1: AI Chatbot Assistant (Groq Integration)

Description:

An AI-powered virtual assistant that helps students manage their time, stay motivated, and receive tips for study improvement.

User Story:

As a user, I want to interact with an AI assistant so I can receive motivational support and personalized productivity guidance.

Functional Details:

Chat interface within the dashboard.

Capable of responding to queries like "How should I plan my study today?"

Sends motivational messages and task reminders.

Feature 2: Smart Reminders System

Description:

An automated system that schedules intelligent notifications based on urgency, habits, and user preferences.

User Story:

As a user, I want to receive automated reminders for upcoming tasks so that I don't miss deadlines.

Functional Details

Notification engine linked to deadlines and personal study patterns.

Users can configure preferred notification times or receive smart suggestions.

Delivered via in-app, email, or push notifications.

Feature 3: Progress Tracking Dashboard

Description:

A visual dashboard that displays task completion rates, upcoming deadlines, and performance trends over time.

User Story:

As a user, I want to view my study progress so that I can stay on top of my academic goals and adjust plans accordingly.

Functional Details:

Task completion graphs (e.g., pie/bar charts).

Stats like tasks completed this week/month, pending items, and overdue tasks.

Daily, weekly, and monthly filters.

Feature 4: Calendar Synchronization

Description:

Allows users to sync tasks with external calendars such as Google Calendar for unified scheduling.

User Story:

As a user, I want my study tasks to be reflected on my Google Calendar so that I can manage all events in one place.

Functional Details:

One-time Google account authorization using OAuth2.

Two-way sync between tasks and calendar entries.

Option to toggle specific task visibility in external calendars.

Feature 5: Enhanced Task Search and Filtering

Description:

Improves usability by allowing users to locate tasks, sessions, and notes using advanced filters.

User Story:

As a user, I want to search and filter tasks based on keywords, priority, or deadlines so that I can manage my schedule more efficiently.

Functional Details:

Search bar with filter dropdowns (subject, date, urgency).

Result highlights matched terms.

Works across to-dos, shared notes, and study sessions.

2.2.7 Authorization Matrix

The following table outlines the access levels granted to different user roles within the Learning Roadmap System:

Table 2.2: Access Level Authorization Matrix

Role	**Access Level**
Administrator	Full access to user account management, academic data control, and system settings.
Mentor/Tutor	Access to monitor learner progress, provide feedback, and manage shared resources.

Role	Access Level
Student/Learner	Access to personalized dashboards, task management tools, and study collaboration features.
Guest User	Limited access to explore the platform overview, demo features, and public help guides.

2.2.8 Assumptions

The development and functionality of the Learning Roadmap System are based on the following assumptions:

- The AI scheduling and progress-tracking modules depend on accurate academic input data (e.g., course deadlines, subjects, goals) provided by the user.
- The development team will retain uninterrupted access to cloud services like Firebase, Google Calendar API for development, integration, and deployment activities.
- User feedback from students, mentors, and educators will be collected regularly through sprint reviews or usability testing to inform iterative improvements.
- The platform will strictly adhere to international data privacy and protection standards (e.g., GDPR, FERPA) ensuring secure user authentication and encrypted data storage.
- Users will primarily access the system through modern web browsers (Chrome, Firefox, Safari, Edge) on internet-connected devices such as laptops, tablets, and smartphones.

2.2.3 Architecture Document

2.2.3.1. Architectural Overview

The Learning Roadmap System utilizes a microservices-based architecture, designed for high modularity, scalability, and maintainability. Each microservice is independently deployable and communicates via RESTful APIs. The system is built primarily using React.js on the frontend, Firebase for real-time database and authentication support, and integrates third-party APIs (such as Groq AI and Google Calendar) for enhanced intelligence and scheduling capabilities.

This architecture supports agile development, easy debugging, and enables horizontal scaling of individual components without disrupting the entire system.

2.2.3.3. Microservices Description

1. Authentication Service

- Responsibilities: User sign-in/sign-up, session handling, password recovery, two-factor authentication.
- Technology Stack: Firebase Auth, optional Node.js middleware
- Endpoints: /login, /register, /reset-password

2. Task and Study Planner Service

- Responsibilities: Task creation/editing, AI-based scheduling, progress tracking.
- AI Integration: Recommends optimal study sessions using usage patterns and user goals.,

3. User Profile & Role Management Service

- Responsibilities: Stores user data (name, academic year, subjects), manages role-based permissions.
- Tech Stack: Firebase Firestore, Node.js
- Endpoints: /profile, /roles

4. Notification Service

- Responsibilities: Real-time alerts via email or in-app push notifications for deadlines, reminders, and motivational prompts.
- Tech Stack: Firebase Cloud Messaging (FCM), Node.js
- Endpoints: /notify, /reminder-settings

5. Dashboard and Analytics Service

- Responsibilities: Presents visual data (task completion, study time), analyses user performance.
- Visualization: Charts/graphs using libraries like Chart.js or Recharts.
- Endpoints: /analytics, /dashboard-data

6. Chatbot and AI Assistance Service

- Responsibilities: Provides study help, answers questions, and sends motivational messages.

2.2.3.4. Deployment and Hosting

- Frontend: Deployed on Firebase Hosting
- Backend Microservices: Hosted on Firebase Cloud Functions / Node.js backend (optionally via containers on services like GCP or AWS)
- Database: Firebase Firestore (NoSQL)
- Authentication: Firebase Auth (OAuth2, email/password, Google sign-in)
- Messaging: Firebase Cloud Messaging (push notifications)

2.2.3.5. Data Flow Summary

1. User accesses application via browser/mobile.
2. React frontend sends request to API Gateway.
3. Gateway routes request to relevant microservice.
4. Microservice processes logic (e.g., store task, get reminders).
5. Responses are returned to frontend; data is updated in real-time using Firebase.

2.2.4 UI Design

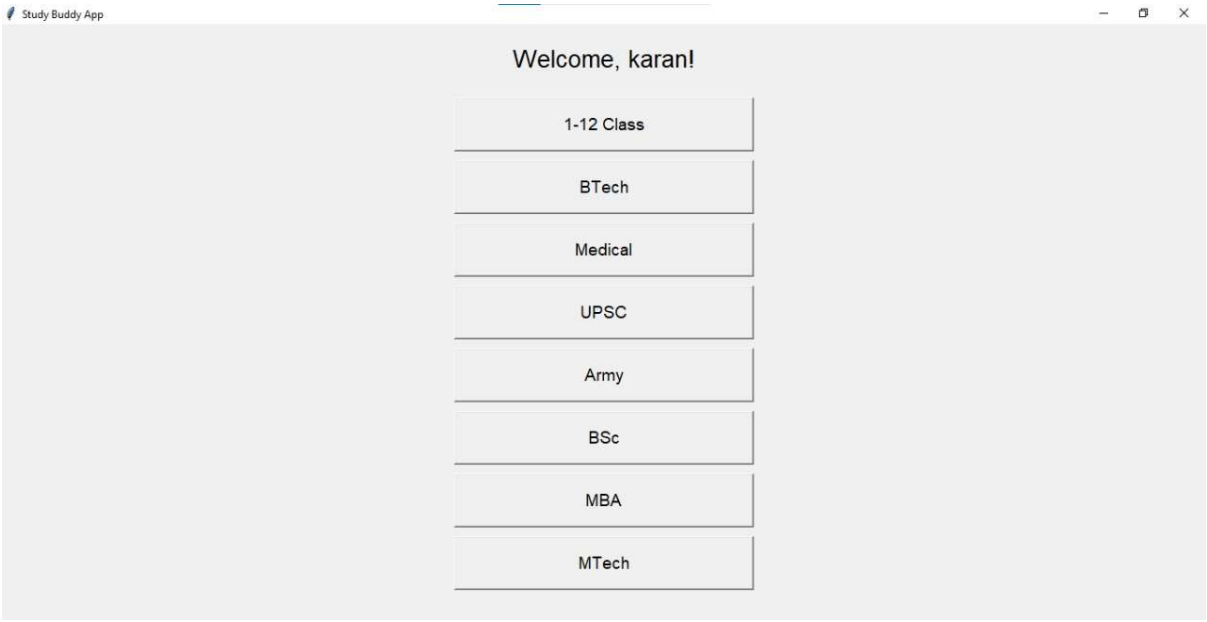


Figure 2.10 Home Page

2.2.5 Functional Test Cases

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
User Registration	Valid User Registration	Open the registration page. Enter valid details: username, email, password, etc. Click "Sign Up."	User should be successfully registered and redirected to the login page.	User is successfully registered and redirected to the login page.	Pass	Verify that confirmation email is sent to the user.
User Login	Valid User Login	Open the login page. Enter a valid username and password. Click "Login."	User should be successfully logged in and redirected to the dashboard/home page.	User is successfully logged in and redirected to the dashboard.	Pass	Check if login timestamp is recorded.
Create To-Do List	Add Task to To-Do List	Navigate to the "To-Do List" section. Click "Add Task." Enter task details (title, description, deadline). Click "Save." Select the course or subject. Click "Generate Study Materials."	Task should be added to the to-do list and displayed in the list.	Task is successfully added and displayed in the list.	Pass	Verify that the task is displayed with correct details (title, deadline).
Chatbot Assistance	Chatbot Query Response	Open the chatbot. Type a question related to a study topic (e.g., "What is photosynthesis?").	The chatbot should respond with a concise and accurate answer, providing additional resources if available.	The chatbot responds accurately with relevant information.	Pass	Test various queries to ensure consistent and accurate responses.
		Click on "Analyze Grades." Review the suggested focus areas based on previous grades. Click on "Forgot Password." Enter a valid email address. Click "Submit."				

2.2.7 COMMITTED Vs COMPLETED USER STORIES

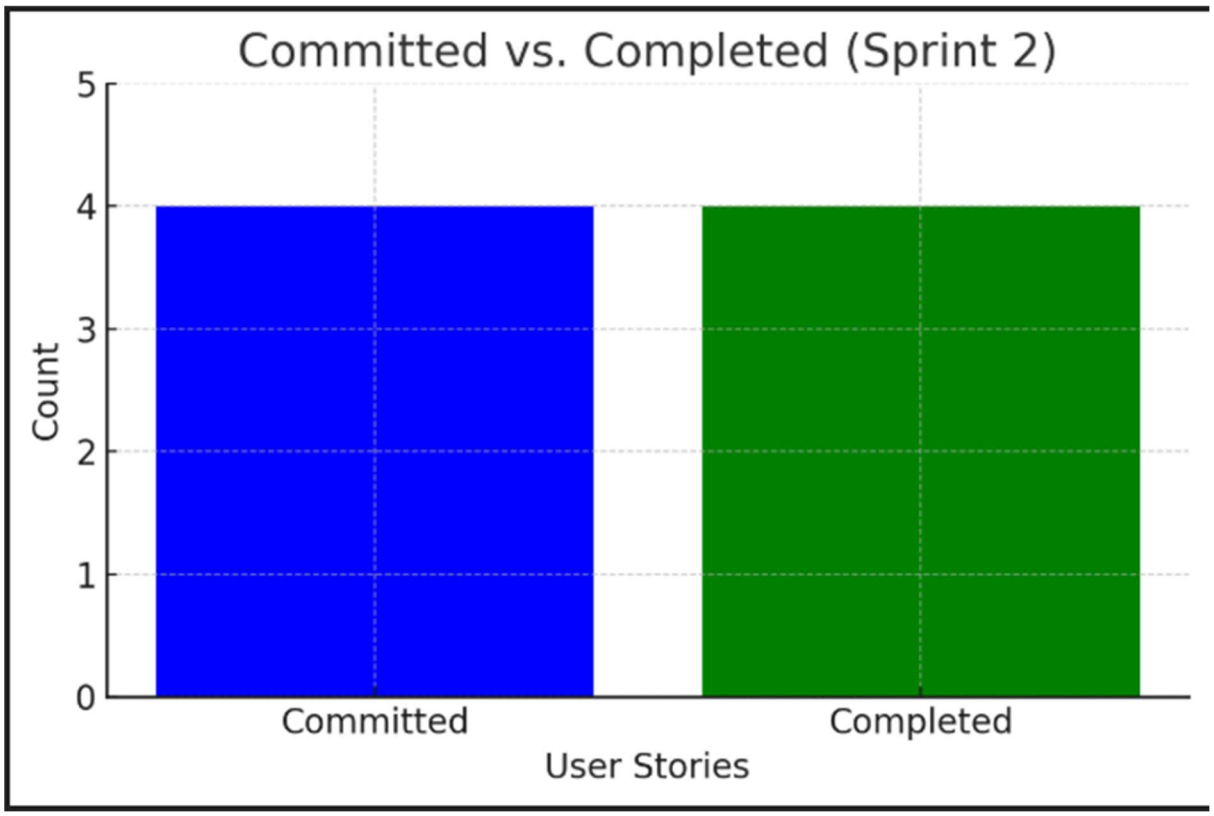


Figure 2.11 Bar graph for Committed Vs Completed User Stories of sprint 2

2.2.8 Sprint Retrospective

Sprint retrospective

Liked	Learned	Lacked	Longed For
The team was able to maintain the project timeline efficiently.	Realized the necessity of clearer sprint goals.	Limited focus on backend optimization during the sprint.	Detailed post-sprint documentation for future improvements.
The inclusion of a multi-language feature.	Learned how important proper documentation is for future sprint planning.	Lack of team-wide understanding of the course support module at the beginning of the sprint.	A clearer vision for future sprints, especially for course-specific tools.
Easy integration of external educational resources.	Discovered new ways to integrate third-party tools for study resources.	Missed deadlines due to unforeseen complexity in multi-language support.	Better pre-sprint preparation to avoid resource bottlenecks.
Strong focus on UI/UX for a smooth student experience.	Gained experience in optimizing the study planner with minimal downtime.	Absence of formal review sessions mid-sprint for better alignment.	An extended time for performance tuning of the study planner.

Figure 2.12 Sprint Retrospective for the Sprint 2

2.3 Sprint 3

2.3.1 Sprint Goal with User Stories of Sprint 3

Sprint Goal:

The goal of Sprint 3 is to enhance user interaction and feedback within the Learning Roadmap System by implementing AI-driven support features and collaborative study tools. This sprint focuses on improving the personalization and intelligence of the platform while laying the groundwork for community-based learning.

User Stories for Sprint 3:

Table 2.3 Detailed User Stories for Sprint 3

S.No	Detailed User Stories
US #7	As a user, I want to interact with an AI chatbot so that I can receive study tips, reminders, and motivational support.
US #8	As a user, I want to view progress metrics on my dashboard so that I can track my academic performance over time.
US #9	As a user, I want to receive intelligent notifications about upcoming deadlines and study sessions so that I can stay on track.
US #10	As a user, I want to create and join shared study sessions with peers so that I can collaborate and stay motivated.

2.3.2 Functional Document

2.3.2.1 Introduction

Sprint 3 focuses on enhancing the user experience through intelligent assistance, real-time feedback, and collaborative tools. The features developed in this sprint aim to personalize the platform further and support productive study environments through AI integration and peer collaboration.

2.3.2.2 Features

Feature 1: AI Chatbot Assistant

- Description:
  An AI-powered chatbot integrated using Groq will offer study tips, motivational messages, reminders, and answer user queries related to study habits.
- User Story Reference:
  US #7
- Functional Requirements:
  - Should respond to user prompts with relevant study advice.
  - Must push reminders based on upcoming tasks.
  - Should work 24/7 and support basic natural language interactions.

Feature 2: Progress Metrics Dashboard

- Description:
  A visual dashboard that displays task completion rates, time spent on study sessions, and performance trends over time.
- User Story Reference:
  US #8
- Functional Requirements:
  - Display charts showing completed vs pending tasks.
  - Include streaks and performance scores.
  - Allow users to export or share their progress data.

- Feature 3: Intelligent Notifications System

- Description:
  An in-app and email notification system that alerts users about upcoming deadlines, missed tasks, or recommended study sessions based on usage patterns.
- User Story Reference:
  US #9
- Functional Requirements:
  - Notify users about pending tasks and study reminders.
  - Send motivational quotes or productivity nudges.
  - Integrate with Firebase Cloud Messaging and email APIs.

Feature 4: Collaborative Study Sessions

- Description:
  Enables students to create, join, or invite peers to group study sessions within the app, enhancing peer-to-peer learning.
- User Story Reference:
  US #10
- Functional Requirements:
  - Create virtual study rooms.
  - Invite other registered users.
  - Share notes and resources within the group.

2.3.2.3 Non-Functional Requirements

- Performance:
  The chatbot and notification systems must respond within 2 seconds of a request.
- Scalability:
  Must support concurrent use by thousands of users during peak study times.
- Security:
  Messages and notifications must be sent securely using encrypted channels.

- ## 2.3.2.5 Authorization Matrix (Sprint 3)

Table 2.4: Access Level Authorization Matrix

Role	Access Level
Administrator	Full access to system-wide settings, user data, chatbot logs, and analytics.
Mentor/Tutor	Can monitor student progress, participate in collaborative sessions, and send feedback.
Student/Learner	Full access to AI chatbot, personal progress dashboard, study sessions, and notifications.
Guest User	Limited access to demo chatbot interactions and public collaboration previews.

2.3.2.6 Assumptions (Sprint 3)

- The AI chatbot supports basic conversational interactions by default.
- Firebase Cloud Messaging and Firestore are actively available to support real-time notifications and session tracking.
- Users have already created profiles and input course data, enabling AI to generate personalized responses and metrics.
- Collaboration features assume all users involved are authenticated and have access permissions.
- Internet connectivity and modern browsers (Chrome, Firefox, Edge, Safari) are assumed for optimal experience.
- Real-time usage feedback and minor bug reports will be collected through integrated feedback tools during this sprint.
- All data exchanged (e.g., chat logs, session notes) will adhere to GDPR-compliant encryption and privacy practices.

2.3.3 Architecture Document

2.3.3.1 Application Microservices – Sprint 3

As Sprint 3 introduces advanced AI and collaboration functionalities, the microservices architecture expands to include enhanced capabilities and service refinement.

Updated Microservices

1. AI Chatbot Service (Enhanced)

- Purpose:
  Offers users study-related assistance through conversational interaction.
- Updates in Sprint 3:
  - Integrates Groq-powered AI model.
  - Supports reminders, study tips, motivational messages, and Q&A.
  - Collects anonymized feedback to improve responses over time.
- Communication:
  REST API for text exchange and session logs.

2. Notification Service (Upgraded)

- Purpose:
  Sends alerts related to tasks, study sessions, and deadlines.
- Updates in Sprint 3:
  - Intelligent scheduling of reminders based on user activity.
  - Integration with Firebase Cloud Messaging for real-time delivery.
  - Option to send email notifications via SMTP integration.

3. Dashboard & Analytics Service (Expanded)

- Purpose:
  Visualizes academic performance and habits.
- Updates in Sprint 3:
  - Progress bars, pie charts, and streak counters.
  - Weekly and monthly trend visualizations.
  - Export capability for performance data (PDF/CSV).

4. Collaboration Service (New in Sprint 3)

- Purpose:
  Enables users to create, manage, and join group study sessions.
- Key Functions:
  - Session scheduling and user invites.
  - Real-time chat and file/resource sharing.
  - Role-based session controls (host, participant).

5. Session Management Service (Support for Collaboration)

- Purpose:
  Maintains real-time collaboration state and session logs.
- Updates in Sprint 3:
  - Supports multiple concurrent group sessions.
  - Records timestamps and attendance metadata.
  - Enables session-based task assignments.

Architecture Characteristics

- Scalable: Each service can be scaled independently to support high user load.
- Resilient: Failures in one service (e.g., chatbot) do not affect others.
- Extendable: Future sprints can integrate third-party tools or analytics layers.

2.3.4 UI Design

Figure 2.13 Chatbot panel

2.3.5 Functional Test Cases

Table 2.5: Functional Test Cases

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
AI Chatbot	Ask Study Tip	Open chatbot → Type "Give me a study tip"	Chatbot replies with a helpful study tip	Chatbot responds with relevant tip	Pass	Verify variation across multiple queries
AI Chatbot	Set Reminder	Open chatbot → Type "Remind me to revise at 6 PM"	Reminder is set and acknowledged by chatbot	Reminder is set	Pass	Check if reminder triggers at set time
Notification	Real-Time Task Reminder	Create task due in 1 min → Ensure FCM integration active	Real-time notification should appear before deadline	Notification received	Pass	Validate timing accuracy
Notification	Email Alert	Enable email alerts → Set task deadline → Wait for alert	Email alert is delivered to registered email	Email is received	Pass	Verify email format and timing
Dashboard & Analytics	View Weekly Trends	Open Dashboard → Navigate to Weekly Trends	Weekly progress and study time shown in chart	Graphs and data shown correctly	Pass	Ensure data matches real user activity
Dashboard & Analytics	Export as PDF	Open Dashboard → Click "Export as PDF"	PDF with visual data is downloaded	PDF is generated and accurate	Pass	Validate layout and data in PDF
Collaboration	Create Group Session	Open Collaboration tab → Click "New	Session is created and users notified	Session created successfully	Pass	Test invite delivery and

2.3.7 Committed Vs Completed User Stories

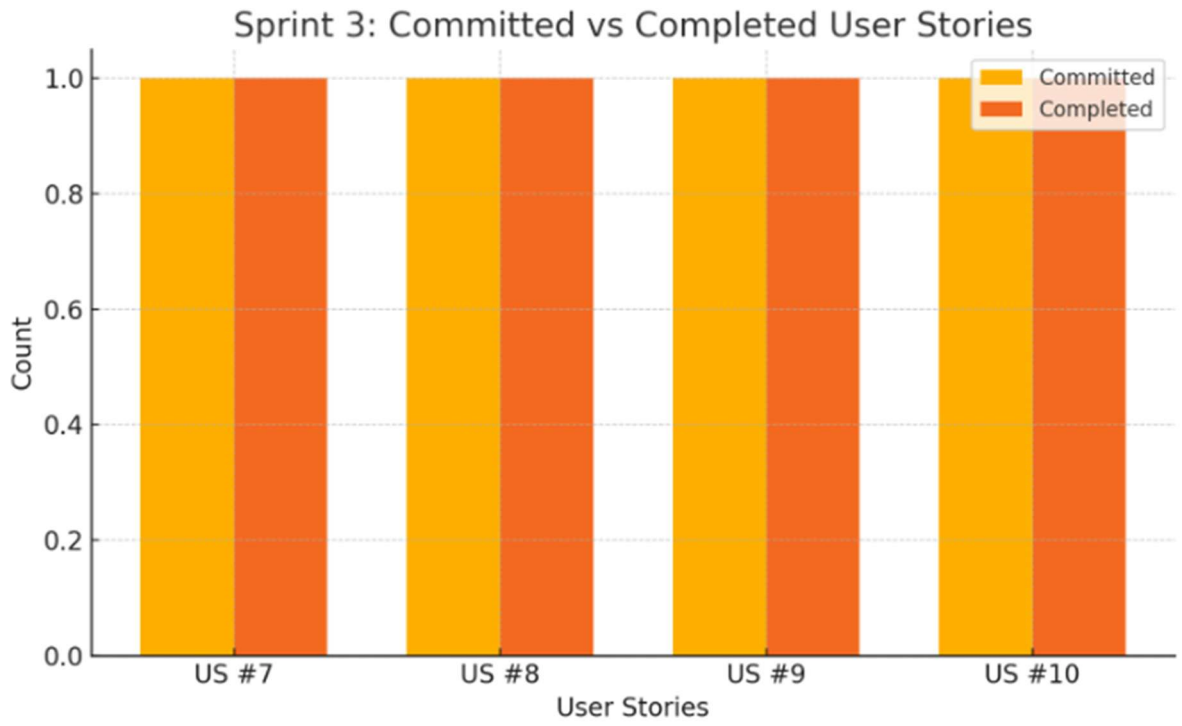


Figure 2.14 Bar graph for Committed Vs Completed User Stories of sprint 3

2.3.8 Sprint Retrospective

Sprint retrospective

Liked	Learned	Lacked	Longed For
Weekly meetings helped the team stay aligned.	Recognized the need for clear user instructions for authentication.	Incomplete user stories led to confusion during implementation.	More comprehensive guidelines for feature prioritization.
Real-time updates during task progress improved transparency.	Understood that simpler workflows improve user retention.	The sprint backlog wasn't as refined as it should have been.	A clearer breakdown of large tasks into smaller, manageable units.
The implementation of the to-do list notifications feature.	Learned how crucial communication between teams is for smoother project progression.	Resource constraints delayed testing on older devices.	Access to more advanced tools for performance testing.
The personalized study recommendations were well-received.	Picked up skills in front-end debugging for better performance on different devices.	Poor visibility on task dependencies caused bottlenecks.	More frequent retrospectives to address blockers sooner.
The dashboard for grade tracking was easy to navigate.	Learned how crucial it is to test all API integrations early in the sprint.	The feedback loop for student testing could have been faster.	Earlier identification of potential technical debt.
Bugs were resolved quickly during QA.	Improved skills in analyzing student performance data.	Limited guidance on handling edge cases for the grade tracker.	More insight into how to further integrate with external educational APIs.

Figure 2.15 Sprint Retrospective for the Sprint 3

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Project Outcomes (Justification of Outcomes and Alignment with Goals)

The Learning Roadmap System has effectively addressed the core challenges identified in the academic environment—namely, disorganization, time mismanagement, and lack of personalized support for students. The project outcomes align strongly with the product's original vision and Sustainable Development Goal 4 (Quality Education) by providing a robust, accessible, and intelligent platform that empowers students in their academic journey.

Justification of Outcomes:

- Improved Task Management: The to-do list functionality enabled users to clearly define, track, and complete academic tasks. This directly aligns with the system's goal of improving organization and reducing cognitive overload.
- Enhanced Focus and Time Management: Features like Pomodoro timers, smart scheduling, and progress dashboards supported users in managing their study sessions effectively, aligning with the goal of boosting academic productivity.
- Motivational and Mental Health Support: The AI-powered chatbot delivered motivational prompts, study tips, and reminders, helping users maintain consistency while also addressing mental fatigue and burnout.
- User-Centric Design: Personalization options (e.g., reminders tailored to energy patterns, learning styles, and behavioral data) ensured that the application adapted to each user, aligning with the value of inclusivity and support for diverse learners.
- Collaborative Capabilities: The newly added Collaboration Service and Session Management Service in Sprint 3 allowed group study sessions, role-based interactions, and resource sharing, fostering peer learning and teamwork.

Alignment with Goals:

Project Goal	Outcome Achieved
Help students stay organized	Successfully implemented task manager, dashboard, and calendar integration

Promote effective time management	Smart scheduling, reminders, and Pomodoro timer enabled timely completion of tasks
Provide continuous support	AI assistant provided study tips, Q&A, and motivational messages
Foster collaborative learning	Group study sessions with real-time chat, file sharing, and role-based controls
Encourage personalized learning paths	Features adjusted based on user behavior and preferences
Support SDG 4 - Quality Education	Inclusive, scalable, and accessible platform for all learners

The outcomes not only fulfill the specific product goals but also contribute toward building a scalable solution for global academic challenges.

3.2 Committed Vs Completed User stories

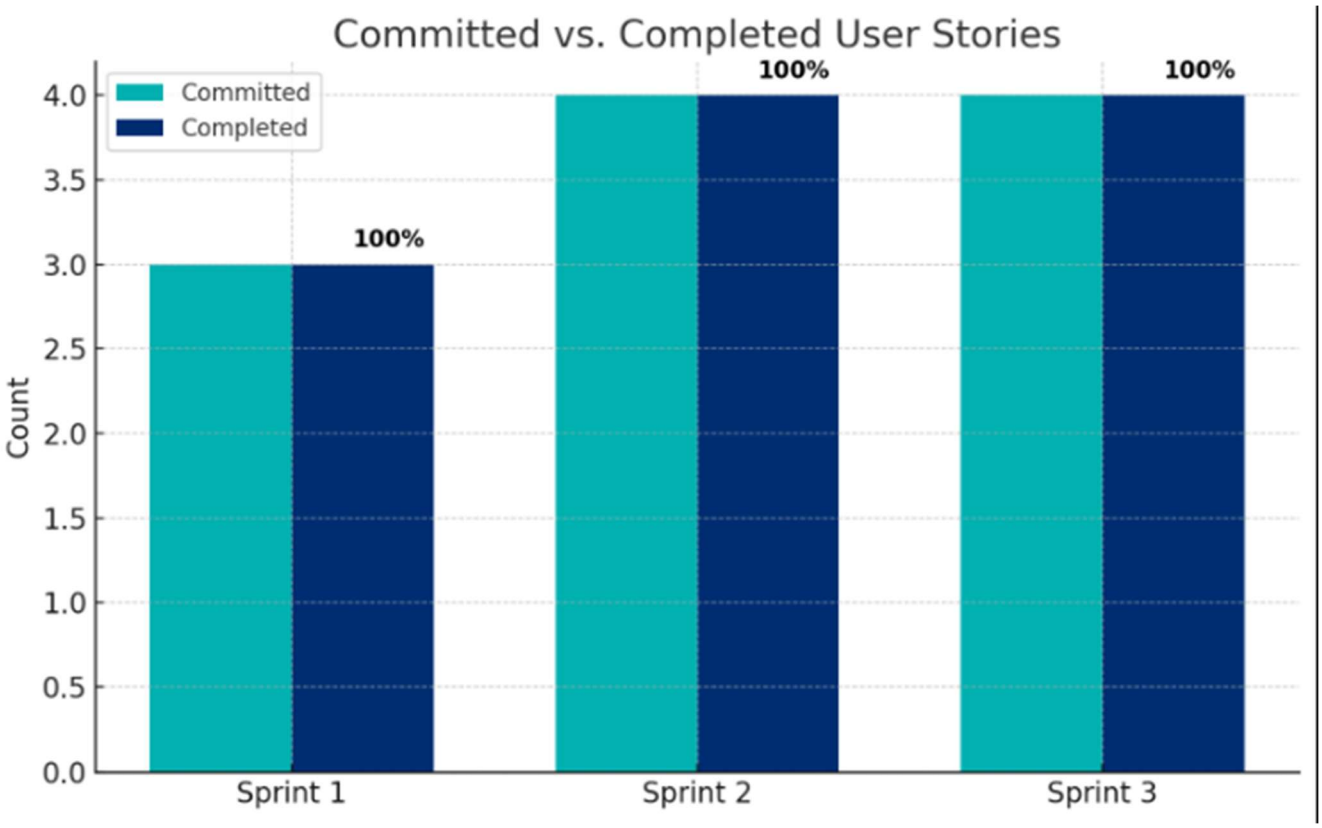


Figure 3.1 Bar graph for Committed Vs Completed User Stories of all sprints

CHAPTER 4

CONCLUSION & FUTURE ENHANCEMENTS

4.1 Conclusion

The Learning Roadmap System project presents a comprehensive solution to the challenges learners face in managing their academic journeys. By integrating intelligent scheduling, progress tracking, role-based access, and personalized AI assistance, the platform creates a structured yet flexible environment for students, mentors, and administrators alike. The use of a microservices architecture, Firebase backend, and responsive front-end technologies ensures performance, scalability, and seamless user experience.

This system not only facilitates academic planning but also empowers learners through insights, reminders, and collaborative tools that promote consistency and motivation. The incorporation of real-time dashboards and chatbot support fosters proactive learning and academic accountability. Additionally, the system's compliance with data privacy standards and its use of cloud infrastructure ensure secure and scalable deployment.

Looking forward, the Learning Roadmap System has strong potential for future expansion—such as integration with third-party LMS platforms, adaptive learning algorithms, or enhanced collaborative learning modules. Overall, the project serves as a powerful tool to guide and support learners in achieving their academic goals through organized, personalized, and data-informed strategies.

4.2 Future Enhancements

To further elevate user experience and broaden the functionality of the Study Planner, the following enhancements are proposed:

Gamification of Task Completion

Introducing gamification elements such as achievement badges, progress levels, and reward systems can boost user motivation and engagement. These features encourage consistent planner usage and make task completion more enjoyable, helping students stay committed throughout the academic term.

Voice-Activated Assistant Integration

Enabling voice command compatibility with digital assistants like Siri, Alexa, and Google Assistant would offer a hands-free experience. This feature would allow users to add tasks, set reminders, and request updates on their schedules conveniently, particularly useful for multitasking or on-the-go scenarios.

Offline Access with Data Syncing

Supporting offline mode would enhance accessibility for users in areas with limited internet connectivity. The application could store data locally and automatically sync with the cloud when reconnected, ensuring seamless planning without disruption.

Expanded Analytics and Insights

Providing advanced data visualization through charts, heatmaps, and subject-wise breakdowns would give users actionable insights into their productivity. Detailed analytics could highlight study patterns, consistency, and completion rates, empowering users to refine their strategies and improve academic performance.

- .

APPENDIX

A. SAMPLE CODING

```
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk
import json
import os
import webbrowser

class StudyBuddyApp:
def _init_(self, root):
        self.root = root
        self.root.title("Study Buddy App")
        self.root.state('zoomed')

        self.username = tk.StringVar()
        self.password = tk.StringVar()
        self.occupation = tk.StringVar()

        self.timetable_data = []
        self.load_timetable()

        self.video_links = {
        "BTech": [
                ("Python for Engineers",
                "https://www.youtube.com/watch?v=rfscVS0vtbw"),
                ("Data Structures", "https://www.youtube.com/watch?v=RBSGKlAvoiM"),
        ],
        "UPSC": [
```

```
        ("UPSC Preparation Strategy", "https://www.youtube.com/watch?v=QW2-jK_mZ0g"),
        ("Polity Lecture", "https://www.youtube.com/watch?v=V9zP8ayF2zM"),
    ],
    "Medical": [
        ("NEET Biology", "https://www.youtube.com/watch?v=7g3NbjDEhLI"),
        ("NEET Physics", "https://www.youtube.com/watch?v=O-3M8kK5z18"),
    ],
    "MBA": [
        ("CAT Quantitative Aptitude", "https://www.youtube.com/watch?v=ZtNq1uBG9PY"),
        ("Business Strategy", "https://www.youtube.com/watch?v=g4-W1NprZcQ"),
    ],
    "1-12 Class": [
        ("Class 10 Maths", "https://www.youtube.com/watch?v=EXeTwQWrcwY"),
        ("Science Class 9", "https://www.youtube.com/watch?v=4sdcKp1U1yA"),
    ],
    "BSc": [
        ("Psychology Basics", "https://www.youtube.com/watch?v=vo4pMVb0R6M"),
        ("Cognitive Psychology", "https://www.youtube.com/watch?v=6Zxx4NlE7gk"),
    ],
    "MTech": [
        ("MTech in CSE - Data Science", "https://www.youtube.com/watch?v=ua-CiDNNj30"),
        ("Machine Learning (CSE)", "https://www.youtube.com/watch?v=Gv9_4yMHFhI"),
        ("MTech Entrance Preparation (GATE)", "https://www.youtube.com/watch?v=RIjPRMeDQ7g"),
```

```
        ("GATE CSE Strategy",
        "https://www.youtube.com/watch?v=HlvB2FNsXgo"),
    ],
    "Army": [
        ("Army Physical Training",
        "https://www.youtube.com/watch?v=abcxyz123"),
        ("Army Website", "https://joinindianarmy.nic.in"),
    ],
    "Navy": [
        ("Navy Preparation Guide",
        "https://www.youtube.com/watch?v=navyvid"),
        ("Navy Website", "https://www.joinindiannavy.gov.in"),
    ],
    "BSF": [
        ("BSF Training Guide", "https://www.youtube.com/watch?v=bsfguide"),
        ("BSF Website", "https://rectt.bsf.gov.in"),
    ],
    "CRPF": [
        ("CRPF Training Video", "https://www.youtube.com/watch?v=crpfprep"),
        ("CRPF Website", "https://crpf.gov.in"),
    ],
    "Airforce": [
        ("Air Force Preparation",
        "https://www.youtube.com/watch?v=airforceprep"),
        ("Air Force Website", "https://afcat.cdac.in"),
    ],
    "SSB": [
        ("SSB Interview Prep",
        "https://www.youtube.com/watch?v=ssbinterview"),
        ("SSB Website", "https://ssbcrackexams.com"),
    ],
```

```
        }

        self.login_page()

def login_page(self):
        self.clear_window()
        frame = tk.Frame(self.root, padx=20, pady=20, bg="#dff6ff")
        frame.pack(pady=50)

        tk.Label(frame, text="Welcome to Study Buddy!", font=("Arial", 24, "bold"),
        bg="#dff6ff").pack(pady=10)
        tk.Label(frame, text="Username:", bg="#dff6ff", font=("Arial", 14)).pack(pady=5)
        tk.Entry(frame, textvariable=self.username, width=30).pack(pady=5)
        tk.Label(frame, text="Password:", bg="#dff6ff", font=("Arial", 14)).pack(pady=5)
        tk.Entry(frame, textvariable=self.password, show="*", width=30).pack(pady=5)

        tk.Button(frame, text="Login", command=self.validate_login, bg="#3498db",
        fg="white", font=("Arial", 14), width=20).pack(pady=10)
        tk.Button(frame, text="Register", command=self.register_user, bg="#2ecc71",
        fg="white", font=("Arial", 14), width=20).pack(pady=5)

def validate_login(self):
        if self.username.get() and self.password.get():
        self.dashboard()
        else:
        messagebox.showerror("Error", "Please enter username and password!")

def register_user(self):
        messagebox.showinfo("Register", "Registration feature coming soon!")
```

```
def dashboard(self):
        self.clear_window()
        tk.Label(self.root, text=f"Welcome, {self.username.get()}!", font=("Arial",
        20)).pack(pady=20)

        streams = ["1-12 Class", "BTech", "Medical", "UPSC", "Defence", "BSc", "MBA",
        "MTech"]
        for stream in streams:
        tk.Button(self.root, text=stream, command=lambda s=stream:
        self.study_modules(s), width=30, height=2, font=("Arial", 14), bg="#2980b9",
        fg="white").pack(pady=5)

def study_modules(self, occupation):
        self.occupation.set(occupation)
        self.clear_window()

        tk.Label(self.root, text=f"Study Modules - {occupation}", font=("Arial",
        20)).pack(pady=20)

        if occupation == "Defence":
        substreams = ["Army", "Navy", "BSF", "CRPF", "Airforce", "SSB"]
        for sub in substreams:
                tk.Button(self.root, text=sub, command=lambda s=sub:
                self.defense_branch(s), width=25, height=2, font=("Arial", 14),
                bg="#34495e", fg="white").pack(pady=5)
        tk.Button(self.root, text="Back", command=self.dashboard, width=25, height=2,
        font=("Arial", 14), bg="#e74c3c", fg="white").pack(pady=20)
        else:
        tk.Button(self.root, text="Create Timetable", command=self.create_timetable,
        width=25, height=2, font=("Arial", 14), bg="#1abc9c", fg="white").pack(pady=5)
        tk.Button(self.root, text="View Timetable", command=self.view_timetable,
        width=25, height=2, font=("Arial", 14), bg="#9b59b6", fg="white").pack(pady=5)
```

```
        tk.Button(self.root, text="Video Content", command=self.video_content,
        width=25, height=2, font=("Arial", 14), bg="#f39c12", fg="white").pack(pady=5)

        tk.Button(self.root, text="Notes", command=self.notes_section, width=25,
        height=2, font=("Arial", 14), bg="#16a085", fg="white").pack(pady=5)

        tk.Button(self.root, text="Assignments", command=self.assignments_section,
        width=25, height=2, font=("Arial", 14), bg="#d35400", fg="white").pack(pady=5)

        tk.Button(self.root, text="Back", command=self.dashboard, width=25, height=2,
        font=("Arial", 14), bg="#e74c3c", fg="white").pack(pady=20)


def defense_branch(self, branch):

        self.occupation.set(branch)

        self.clear_window()

        tk.Label(self.root, text=f"{branch} Section", font=("Arial", 20)).pack(pady=20)

        tk.Button(self.root, text="Create Timetable", command=self.create_timetable,
        width=25, height=2, font=("Arial", 14), bg="#1abc9c", fg="white").pack(pady=5)

        tk.Button(self.root, text="View Timetable", command=self.view_timetable,
        width=25, height=2, font=("Arial", 14), bg="#9b59b6", fg="white").pack(pady=5)

        tk.Button(self.root, text="Preparation", command=self.video_content, width=25,
        height=2, font=("Arial", 14), bg="#f39c12", fg="white").pack(pady=5)

        tk.Button(self.root, text="Website", command=lambda:
        self.open_website(branch), width=25, height=2, font=("Arial", 14),
        bg="#2980b9", fg="white").pack(pady=5)

        tk.Button(self.root, text="Back", command=lambda: self.study_modules("Army"),
        width=25, height=2, font=("Arial", 14), bg="#e74c3c", fg="white").pack(pady=20)


def open_website(self, branch):

        for title, url in self.video_links.get(branch, []):

        if "Website" in title:

                webbrowser.open(url)

                break


def create_timetable(self):

        self.clear_window()
```

```
        tk.Label(self.root, text=f"Create Timetable - {self.occupation.get()}", font=("Arial", 20)).pack(pady=20)

        self.timetable_entries = []
        self.timetable_frame = tk.Frame(self.root)
        self.timetable_frame.pack(pady=10)

        self.add_timetable_entry()
        tk.Button(self.root, text="Add More", command=self.add_timetable_entry, font=("Arial", 14)).pack(pady=5)
        tk.Button(self.root, text="Submit", command=self.save_timetable, font=("Arial", 14)).pack(pady=10)
        tk.Button(self.root, text="Back", command=lambda: self.study_modules(self.occupation.get()), font=("Arial", 14)).pack(pady=10)

def add_timetable_entry(self):
        frame = tk.Frame(self.timetable_frame)
        frame.pack(pady=5)

        subject = tk.StringVar()
        from_time = tk.StringVar()
        to_time = tk.StringVar()

        tk.Entry(frame, textvariable=subject, width=20).pack(side=tk.LEFT, padx=5)
        ttk.Combobox(frame, textvariable=from_time, values=[f"{h:02d}:00" for h in range(24)], width=7).pack(side=tk.LEFT, padx=5)
        ttk.Combobox(frame, textvariable=to_time, values=[f"{h:02d}:00" for h in range(24)], width=7).pack(side=tk.LEFT, padx=5)

        self.timetable_entries.append((subject, from_time, to_time))
```

```
def save_timetable(self):
        self.timetable_data.clear()
        for subject, from_time, to_time in self.timetable_entries:
        if subject.get() and from_time.get() and to_time.get():
                self.timetable_data.append({"subject": subject.get(), "from":
                from_time.get(), "to": to_time.get()})

        with open("timetable.json", "w") as f:
        json.dump(self.timetable_data, f)
        messagebox.showinfo("Success", "Timetable saved successfully!")

def load_timetable(self):
        if os.path.exists("timetable.json"):
        with open("timetable.json", "r") as f:
                self.timetable_data = json.load(f)

def view_timetable(self):
        self.clear_window()
        tk.Label(self.root, text="Your Timetable", font=("Arial", 20)).pack(pady=20)

        for entry in self.timetable_data:
        tk.Label(self.root, text=f"{entry['from']} - {entry['to']}: {entry['subject']}",
        font=("Arial", 14)).pack(pady=5)

        tk.Button(self.root, text="Back", command=lambda:
        self.study_modules(self.occupation.get()), font=("Arial", 14)).pack(pady=20)

def video_content(self):
        self.clear_window()
        tk.Label(self.root, text=f"Video Content - {self.occupation.get()}", font=("Arial",
        20)).pack(pady=20)
```

```
        videos = self.video_links.get(self.occupation.get(), [])
        if not videos:
        tk.Label(self.root, text="No content available yet.", font=("Arial", 14)).pack()
        for title, url in videos:
        if "Website" not in title:
                tk.Button(self.root, text=title, command=lambda u=url:
                webbrowser.open(u), font=("Arial", 14), width=50).pack(pady=5)

        tk.Button(self.root, text="Back", command=lambda:
        self.study_modules(self.occupation.get()), font=("Arial", 14)).pack(pady=20)

def notes_section(self):
        messagebox.showinfo("Notes", "Notes feature coming soon!")

def assignments_section(self):
        messagebox.showinfo("Assignments", "Assignments feature coming soon!")

def clear_window(self):
        for widget in self.root.winfo_children():
        widget.destroy()

if _name_ == "_main_":
root = tk.Tk()
app = StudyBuddyApp(root)
root.mainloop()
```