# Campus Connect: Where Students Meet Opportunity

A PROJECT REPORT

*Submitted by*

OMPREET CHOUDHURY [RA2211026010396]

ABHIGYAAN CHAUBEY [RA2211026010398]

AMRITESHKUMAR YADAV [RA2211026010403]

ANSHIKA SINGH [RA2211026010428]

*Under the Guidance of*

DR. ANTONY SOPHIA N

Assistant Professor, Department of Computational Intelligence

*in partial fulfillment of the requirements for 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 AND TECHNOLOGY

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/ 21CSC303J

**Student Name** : Ompreet Choudhury, Abhigyaan Chaubey, Amriteshkumar Yadav, Anshika Singh

**Registration Number** : RA2211026010396, RA2211026010398, RA2211026010403, RA2211026010428

**Title of Work** : Campus Connect: Where Students Meet Opportunity

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 21CSC303J - Software Engineering and Project Management report titled "**CAMPUS CONNECT: WHERE STUDENTS MEET OPPORTUNITY**" is the bonafide work of "**OMPREET CHOUDHURY [RA2211026010396], ABHIGYAAN CHAUBEY [RA2211026010398], AMRITESHKUMAR YADAV [RA2211026010403], ANSHIKA SINGH [RA2211026010428]"** who carried out the project work 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. ANTONY SOPHIA N**
**SUPERVISOR**
ASSISTANT PROFESSOR
DEPARTMENT OF
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,** Professor and Head, Department of Computational Intelligence, 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. Salomi M**, 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. Antony Sophia N**, Department of Computational Intelligence, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of the Department 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

Campus Connect is a complete full-stack web application built to facilitate communication, interaction, and event management within a college or university setting. The platform is particularly built to serve students, faculty, and outside organizations through a centralized virtual space where information is readily available, activities are coordinated, and participation is facilitated. Developed on contemporary technologies like Next.js for the frontend and Supabase for backend functions, Campus Connect provides a seamless, scalable, and secure user experience on any device or platform.

One of the most notable features of Campus Connect is its secure authentication process that enables users to sign up for accounts, login, and access customized dashboards. With these dashboards, the students are able to sign up for events, control their involvement, check upcoming events, and access specially recommended events based on their interest. The organizers and club administrators are also offered robust features to design, edit, and control events, monitor attendance, and interact with the target group. This two-way functionality allows both organizers and students to have special areas to deal with the platform based on their role.

Campus Connect stresses real-time data processing, utilizing Supabase's cloud database services for efficient event management and user tracking. Security is bolstered by the utilization of JWT-based authentication and guarded API routes, guaranteeing the privacy and secure management of user data. The app also includes an appealing, responsive user interface built using Tailwind CSS, providing a contemporary, intuitive experience that optimizes usability and accessibility.

Aside from its technical characteristics, Campus Connect tackles real-world problems commonly encountered by universities, including siloed communication pathways, ineffective event promotion, and low participation. By merging these capabilities into one unified platform, the project encourages a more integrated and dynamic campus experience. It encourages cooperation, enhances information flow, and boosts student participation.

This project reflects hands-on proficiency in full-stack development, cloud integration, secure authentication methodologies, and user-focused interface design. Campus Connect is a production-ready, scalable solution for bettering the academic life as well as social life in educational institutes, which ultimately leads to a more vibrant and collaborative campus society.

# TABLE OF CONTENTS

**ABSTRACT** **4**

**TABLE OF CONTENTS** **5**

**LIST OF FIGURES** **7**

**LIST OF TABLES** **9**

**ABBREVIATIONS** **10**

**CHAPTER NO. TITLE PAGE NO.**

**1 INTRODUCTION** **11**

1.1 Introduction to Project 11

1.2 Motivation 11

1.3 Sustainable Development Goal of the Project 12

1.4 Product Vision Statement 12

1.5 Product Goal 14

1.6 Product Backlog 14

1.7 Product Release Plan 15

**2 SPRINT PLANNING AND EXECUTION** **16**

2.1 Sprint 1 16

2.1.1 Sprint Goal with User Stories of Sprint 1 16

2.1.2 Functional Test Cases 18

2.1.3 Committed vs Completed User Stories 18

2.1.4 Sprint Retrospective 18

2.2 Sprint 2 19

2.2.1 Sprint Goal with User Stories of Sprint 1 19

2.2.2 Functional Test Cases 20

2.2.3 Committed vs Completed User Stories 20

2.2.4 Sprint Retrospective 21

2.3 Execution 22

2.3.1 Functional Document 22

2.3.2 Architecture Document 26

2.3.3 UI Design 30

**3 RESULTS AND DISCUSSIONS** **32**

3.1 Project Outcomes 32

3.2 Committed vs Completed User Stories 32

**4 CONCLUSIONS & FUTURE ENHANCEMENT 33**

4.1 Conclusion 33

4.2 Future Enhancement 33

**APPENDIX 34**

**A. SAMPLE CODING 34**

**B. PLAGIARISM REPORT 36**

# LIST OF FIGURES


| FIGURE NO. | TITLE | PAGE NO. |
|---|---|---|
| 1.1 | MS Planner Board of CampusConnect | 15 |
| 1.2 | Release plan of CampusConnect | 15 |
| 2.1 | User story1 | 16 |
| 2.2 | User story 2 | 16 |
| 2.3 | User story 3 | 17 |
| 2.4 | User story 5 | 17 |
| 2.5 | Functional test cases for sprint 1 | 18 |
| 2.6 | Bar graph for Committed Vs Completed User Stories for sprint 1 | 18 |
| 2.7 | Sprint Retrospective for the Sprint 1 | 18 |
| 2.8 | User story 4 | 19 |
| 2.9 | User story 6 | 19 |
| 2.10 | User story 9 | 20 |
| 2.11 | Functional test cases for sprint 2 | 20 |
| 2.12 | Bar graph for Committed Vs Completed User Stories for sprint 2 | 20 |
| 2.13 | Sprint Retrospective for the Sprint 2 | 21 |
| 2.14 | Architecture Diagram | 30 |
| 2.15 | Landing Page | 30 |
| 2.16 | Login Page | 30 |
| 2.17 | Dashboard Page | 30 |
| 2.18 | Sign Up Page | 30 |
| 2.19 | Events Details Page | 31 |
| 2.20 | Events Search Page | 31 |
| 2.21 | Admin Authentication Page | 31 |
| 2.22 | About Page | 31 |
| 3.1 | Committed vs Completed Stories | 32 |
| A.1 | Sample Coding 1 | 34 |
| A.2 | Sample Coding 2 | 34 |
| A.3 | Sample Coding 3 | 35 |

A.4 Sample Coding 4 35

B.1 Plagiarism Report 36

# LIST OF TABLES


| TABLE NO. | TITLE | PAGE NO. |
|---|---|---|
| 2.1 | Detailed User Stories of sprint 1 | 16 |
| 2.2 | Detailed User Stories of sprint 2 | 19 |
| 2.4 | Authorization Matrix | 25 |

# ABBREVIATIONS

| Abbreviation | Full Form |
|---|---|
| UI | User Interface |
| UX | User Experience |
| API | Application Programming Interface |
| REST | Representational State Transfer |
| DB | Database |
| JWT | JSON Web Token |
| ORM | Object Relational Mapping |
| CRUD | Create, Read, Update, Delete |
| HTTPS | HyperText Transfer Protocol Secure |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |
| DBMS | Database Management System |
| IDE | Integrated Development Environment |
| UI/UX | User Interface/User Experience |
| MFA | Multi-Factor Authentication |
| DB Schema | Database Schema |
| SQL | Structured Query Language |
| OTP | One-Time Password |
| SPA | Single Page Application |
| AI | Artificial Intelligence |

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction to Campus Connect: Where Students Meet Opportunity

In today's changing higher education environment, effective communication, collaboration, and engagement are important factors that aid students as well as institutions in their success. Old modes of communicating information through physical notice boards, mass emails, and offline event registrations tend to create disjoined communication, lost opportunities, and low levels of student engagement. To overcome such challenges, Campus Connect has been designed as a centralized online platform that consolidates announcements, event management, and student engagement within the context of a university campus. Through the adoption of cutting-edge full-stack development methodologies, the platform will simplify the transmission of information and establish an inclusive, dynamic ecosystem in which students and organizations may engage effectively.

Campus Connect is a web application developed with Next.js for the frontend and Supabase for backend operations, providing real-time data management, secure authentication, and smooth user experiences on all devices. Students can simply sign up, log in, access personalized dashboards, register for events, and receive personalized notifications and recommendations. Organizers and club administrators have specific tools to manage events, monitor participation, and communicate updates effectively. The system ensures security via JWT-based authentication and role-based access controls, whereas the responsive and contemporary UI, developed using Tailwind CSS, ensures ease of use and interaction. Campus Connect not just addresses the communication and coordination problems prevalent in educational institutions but also creates a greater sense of community and engagement by getting students, teachers, and external stakeholders on a common, interactive platform.

## 1.2 Motivation

The Campus Connect platform was conceptualized with the intent of bridging the gap among students, campus events, and event organizers. In the fast-paced academic world, students tend to miss out on significant events because they lack access to centralized information, while event organizers lack the means to reach and connect with their intended audience. This project shall enable students to control their academic and extracurricular calendars effectively, keeping them in touch with the events of utmost importance. With the implementation of a secure login system, event registration, and real-time alerts, Campus Connect provides an optimized experience to users. The application also has filters and categories to enable students to easily locate events of concern and register accordingly. Moreover, the platform also supports sharing events to make sure that even people who are not part of the system are included. As a full-stack solution, this platform not only enhances communication but also makes event organizers benefit from valuable insights, hence making them better planners and coordinators of future events. In the end, Campus Connect is not merely a site—it's a community-building tool that enables students to get the most out of their campus life while creating a more structured and inclusive setting.

## 1.3 Sustainable Development Goal of the Project

The Campus Connect initiative is highly related to a number of United Nations Sustainable Development Goals (SDGs), i.e., SDG 4 (Quality Education), SDG 9 (Industry, Innovation, and Infrastructure), and SDG 11 (Sustainable Cities and Communities). By creating a digital platform that provides secure user authentication, event discovery, registration management, personalized dashboards, and real-time communication, Campus Connect can increase student participation and learning opportunities in the campus ecosystem. By extending digital access to campus events, the project invigorates the active involvement of students, advances innovation within campus systems, and promotes an extended, more robust, and more sustainable student body. The project helps foster the culture of a smart campus wherein students become better informed, engaged, and enabled to become actively involved in their academic and extra-curricular learning.

## 1.4 Product Vision Statement

### 1.4.1. Audience:

• **Primary Audience:** Students, event organizers, campus administrators, club members seeking to engage with campus activities, events, and announcements.

• **Secondary Audience:** Academic institutions, student unions, event sponsors.

### 1.4.2. Needs:

- **Primary Needs:**
  - A centralized platform to discover, explore, and register for campus events.
  - Secure user authentication and profile management.
  - Personalized dashboards showing registered events and notifications.
  - Timely announcements and event updates from organizers and admins.
  - Efficient communication between students and event organizers.
- **Secondary Needs:**
  - Track event participation and engagement metrics.
  - Event promotion and targeted announcements.
  - Simplified event registration and tracking system for organizers.
  - Seamless campus-wide notification and reminder system.

### 1.4.3. Products:

- **Core Product:**
  - Secure login/signup system with email authentication.
  - Centralized event discovery and registration platform.
  - Personalized student dashboard showcasing registered events and notifications.
  - Admin panel for announcements, event management, and communication.
- **Additional Features:**

- **Smart event categorization:** "Featured", "Happening Now", and "Upcoming" events.
- **Search and Filter Options:** Filter events by category, date, type, or organizer.
- **Event Recommendations:** Personalized based on interests and registration history.
- **Event Sharing:** Share event links externally with non-registered students.
- **Notification System:** Alerts for upcoming events and important updates.
- **Responsive and Mobile-Friendly Interface:** Ensures easy access across devices.
- **Data Security:** Password hashing and session management for user safety.
- **Analytics:** Event organizers can see engagement and registration insights.

### 1.4.4. Values:

- **Core Values:**
  1. **User-Centricity:** Focused on providing a smooth and engaging campus event experience.
  2. • **Security:** Prioritize user data privacy with encrypted authentication and secure backend.
  3. • **Accessibility:** Fully responsive design, ensuring access anytime, from any device.
  4. • **Efficiency:** Make event discovery, registration, and management seamless and quick.
  5. • **Community Building:** Foster a connected and active campus community.
- **Differentiators:**
  1. **Centralized Event Platform:** All campus activities available in one place.
  2. **Smart Dashboards:** Personalized views of upcoming and registered events.
  3. **Notification and Reminders:** Keeps users updated without missing any events.
  4. **Admin Announcements:** Direct communication from organizers to students instantly.
  5. **Secure Authentication:** Reliable session handling, database integration for real users.

## 1.5 Product Goal

The primary goal of the Campus Connect project is to design and develop a comprehensive, secure, and scalable digital platform that bridges the gap between students, clubs, and event organizers within a college or university environment. The platform is aimed at providing a centralized space where students can seamlessly discover, explore, register for, and participate in various campus activities, events, and initiatives.

Campus Connect seeks to simplify the traditionally fragmented process of campus event management by offering an intuitive user experience with secure email-based authentication, personalized dashboards, event filtering, and smart notifications. It empowers students by offering real-time updates about events, personalized event recommendations based on their interests, and easy event registration workflows. For organizers and club administrators, the platform provides tools to create, manage, and promote events while tracking student engagement and registrations through insightful analytics.

Furthermore, Campus Connect promotes community building and enhances student engagement by keeping students informed, involved, and motivated to participate in extracurricular activities beyond academics. It fosters a collaborative and connected campus atmosphere, where communication between students and organizers is streamlined, opportunities are more accessible, and participation is encouraged.

In essence, the product goal is to create a secure, reliable, and innovative event management solution that not only improves student life on campus but also supports the institution's mission to provide holistic education through enhanced extracurricular involvement and digital innovation.

## 1.6 Product Backlog

| User Story Number | User Story (Short Description) |
|---|---|
| User Story 1 | Secure email authentication for user data protection and safe event participation. |
| User Story 2 | Admin needs a secure authentication and database to track student registrations. |
| User Story 3 | Students should easily browse events, apply filters, and register for events seamlessly. |
| User Story 4 | Announcements and updates should be visible to all users on the homepage regularly. |
| User Story 5 | Students can view their registered events in a personal dashboard to manage their schedule. |
| User Story 6 | Students should be able to share event links externally with friends outside Campus Connect. |
| User Story 7 | Events should be categorized into Featured, Happening Now, and Upcoming for easy filtering. |
| User Story 8 | Notification alerts to remind students about upcoming registered events automatically. |
| User Story 9 | An "About You" page sharing Campus Connect’s vision, mission, and features to users. |

Table 1.1 Product backlog of Campus Connect

Using the MS planning Agile Board, Campus Connect set out their product backlog; this is shown in Figure 1.1. The whole user tales of Campus Connect make up the Product Backlog.

Every user narrative has functional and nonfunctional criteria, MoSCoW prioritization, and thorough acceptance criteria including connected tasks.

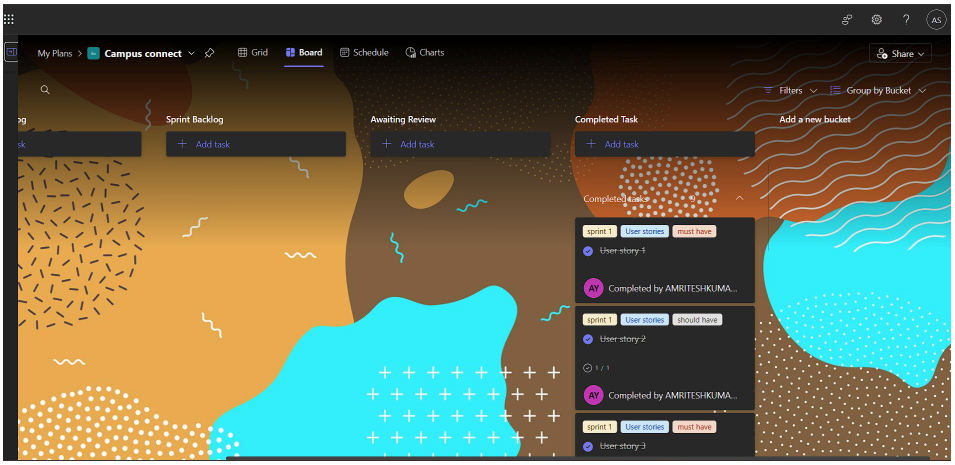


Figure 1.1 MS Planner Board of Campus Connect

# 1.6 Product Release Plan

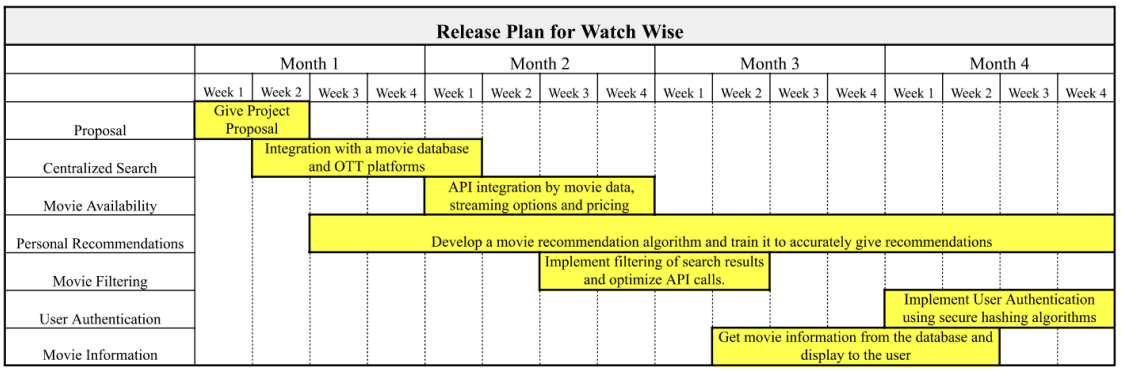


Figure 1.2 Release plan of Campus Connect

# CHAPTER 2

# SPRINT PLANNING AND EXECUTION

## 2.1 Sprint 1

### 2.1.1 Sprint Goal with User Stories of Sprint 1

The Goal of the first sprint is to construct the user landing page and to enable the search functionalities such as skills and courses. The following table 2.1 represents the detailed user stories of the sprint

| User Story Number | User Story (Short Description) |
|---|---|
| User Story 1 | Secure email authentication for user data protection and safe event participation. |
| User Story 2 | Admin needs a secure authentication and database to track student registrations. |
| User Story 3 | Students should easily browse events, apply filters, and register for events seamlessly. |
| User Story 5 | Students can view their registered events in a personal dashboard to manage their schedule. |

Table 2.1 Detailed User Stories of sprint 1

Planner Board representation of user stories are mentioned below figures 2.1, 2.2, 2.3 and 2.4

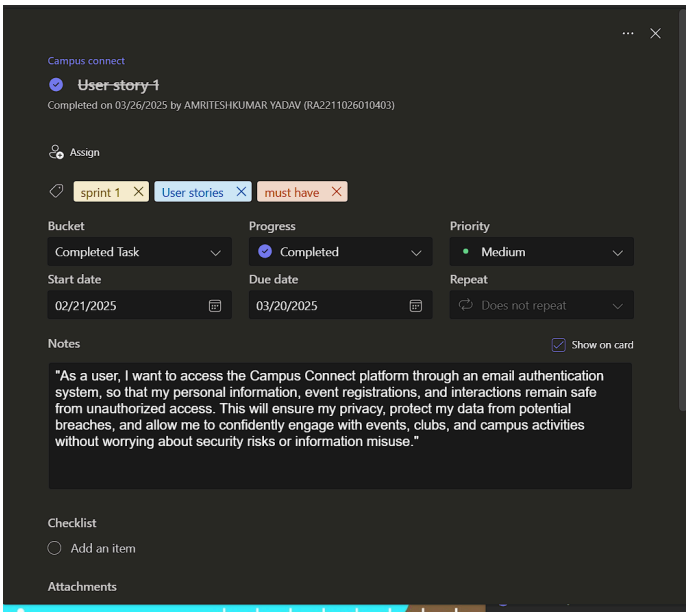


Figure 2.1 User story1

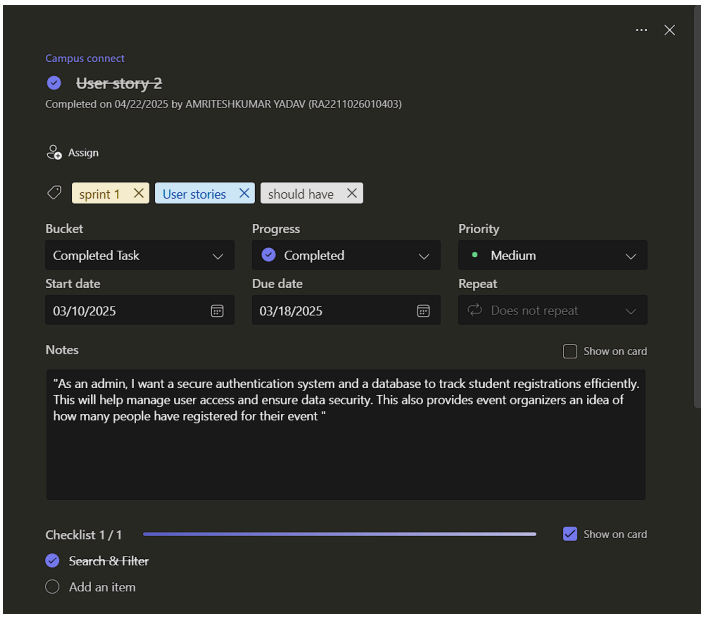


Figure 2.2 User story 2

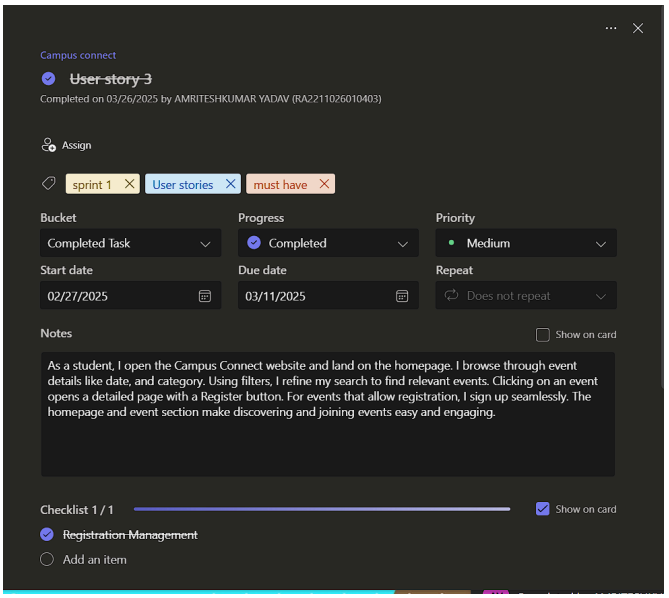


Figure 2.3 User story 3

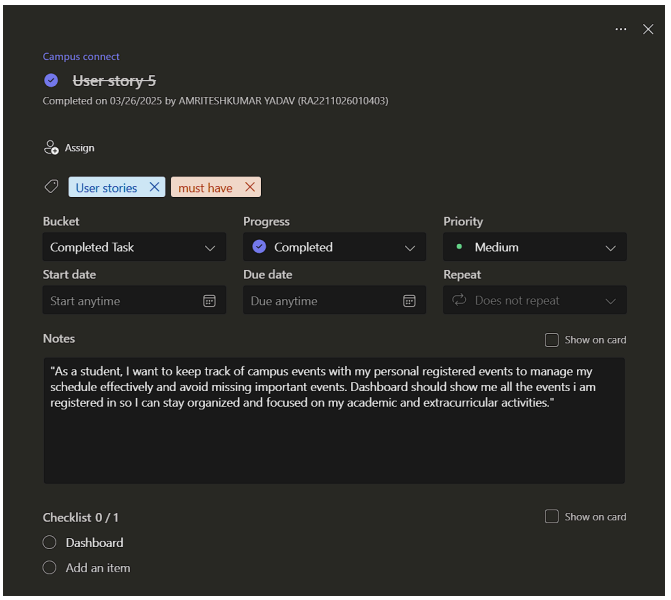


Figure 2.4 User story 5

## 2.1.2 Functional Test Cases

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| | | | Functional Test Case Template | | | | |
| | **Feature** | **Test Case** | **Steps to execute test case** | **Expected Output** | **Actual Output** | **Status** | **More Information** |
| | User Authentication | Validate user signup with email and password | 1. Open Campus Connect website<br>2. Click "Sign Up"<br>3. Enter valid email and password<br>4. Submit form | New user account created successfully and redirected to login page | New user account created successfully and redirected to login page | Pass | Check if confirmation email (optional) is sent |
| | User Authentication | Validate user login with valid credentials | 1. Open login page<br>2. Enter registered email and password<br>3. Click "Login" | Redirected to dashboard or homepage after successful login | Redirected to dashboard or homepage after successful login | Pass | Ensure JWT/session is created |
| | User Authentication | Login with invalid email/password | 1. Open login page<br>2. Enter wrong credentials<br>3. Click "Login" | Error message "Invalid email or password" displayed | Error message "Invalid email or password" displayed | Pass | Error message should be user-friendly |
| | Event Browsing | View list of events on homepage | 1. Login<br>2. Navigate to homepage<br>3. Scroll to Events section | All upcoming, featured events are displayed correctly | All upcoming, featured events are displayed correctly | Pass | Data fetched dynamically |
| | Event Registration | Register for an event | 1. Login<br>2. Open Events page<br>3. Click on an event<br>4. Click "Register" button | User successfully registered for the event and confirmation shown | User successfully registered for the event and confirmation shown | Pass | Event ID linked to user in database |
| | Dashboard Access | View registered events in dashboard | 1. Login<br>2. Navigate to dashboard<br>3. Check "My Events" section | List of registered events displayed correctly | List of registered events displayed correctly | Pass | Dashboard should load instantly |
| | Session Management | Session persists after login | 1. Login<br>2. Refresh the page<br>3. Try accessing dashboard without logging in again | User remains logged in (session not lost) | User remains logged in (session not lost) | Pass | Use secure HTTP-only cookies or JWT |
| | Session Management | Logout functionality | 1. Login<br>2. Click "Logout" button<br>3. Try accessing dashboard again | User is redirected to login page and session terminated | User is redirected to login page and session terminated | Pass | Session should be cleared completely |

Figure 2.5 Functional test cases for sprint 1

### 2.1.3 Committed Vs Completed User Stories

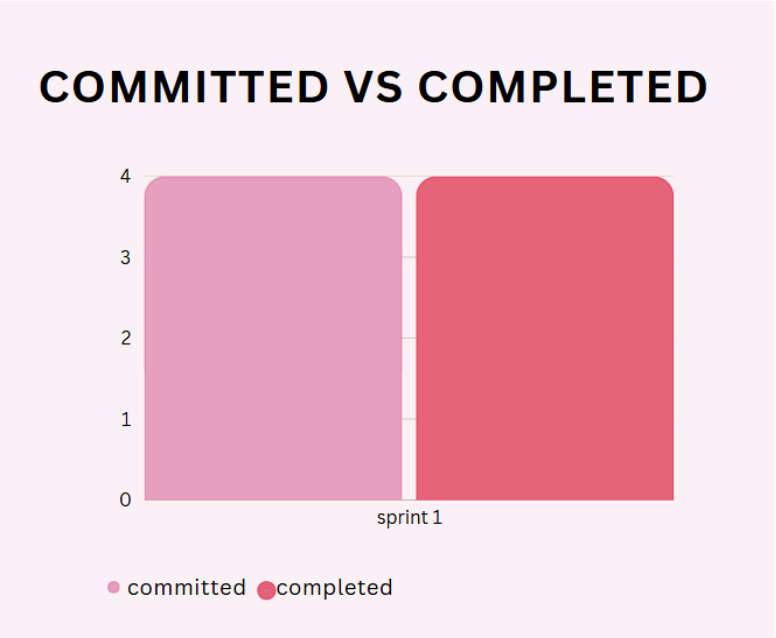


Figure 2.6 Bar graph for Committed Vs Completed User Stories for sprint 1

### 2.1.4 Sprint Retrospective

| | Sprint Retrospective | | |
|---|---|---|---|
| **What went well** | **What went poorly** | **What ideas do you have** | **How should we take action** |
| Successful basic authentication (Signup/Login) implemented | Initial setup of database integration took longer than expected | Setup environment variables and database connections at the very beginning | Create a checklist for environment setup before coding starts |
| Homepage and Events section designed and developed with good UI | Event registration feature had bugs during first implementation | Test each feature independently before combining them | Apply Test-Driven Development (TDD) for core features |
| User registration and event registration processes were completed smoothly | Session management (keeping users logged in) had inconsistencies initially | Research better session handling methods early (JWT, secure cookies) | Plan session management and authentication flow properly in sprint planning |
| Team collaboration was efficient and tasks were divided well | Lack of initial proper testing led to more rework later | Implement continuous testing after every major feature | Set a small dedicated time for QA/testing after each feature |
| Dashboard basic layout created after login | Dashboard loading after login was not very smooth | Improve dashboard data fetching and optimize API responses | Add lightweight loading indicators and prefetch required data |
| | | | |

Figure 2.7 Sprint Retrospective for the Sprint 1

## 2.2 Sprint 2

### 2.2.1 Sprint Goal with User Stories of Sprint 2

The Goal of the second sprint is to add functionality such as movie recommendation ,watchlist creation and movie filtering based on user requirement.

The following table 2.4 represents the detailed user stories of the sprint 2

| **User Story Number** | **User Story (Short Description)** |
|---|---|
| User Story 4 | Announcements and updates should be visible to all users on the homepage regularly. |
| User Story 6 | Students should be able to share event links externally with friends outside Campus Connect. |
| User Story 7 | Events should be categorized into Featured, Happening Now, and Upcoming for easy filtering. |

| | |
|---|---|
| User Story 8 | Notification alerts to remind students about upcoming registered events automatically. |
| User Story 9 | An "About You" page sharing Campus Connect's vision, mission, and features to users. |

Table 2.2 Detailed User Stories of sprint 2

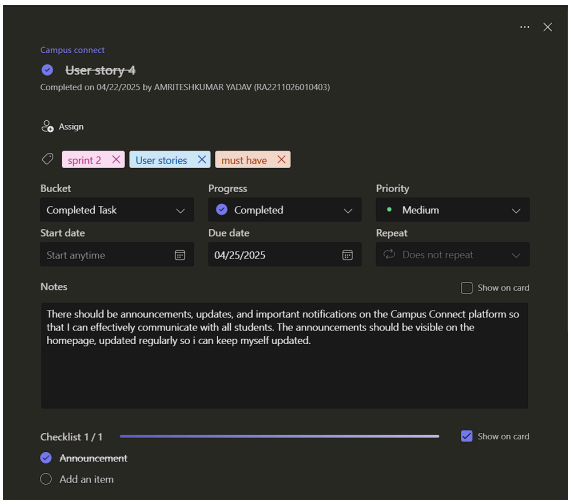


Figure 2.8 User story 4

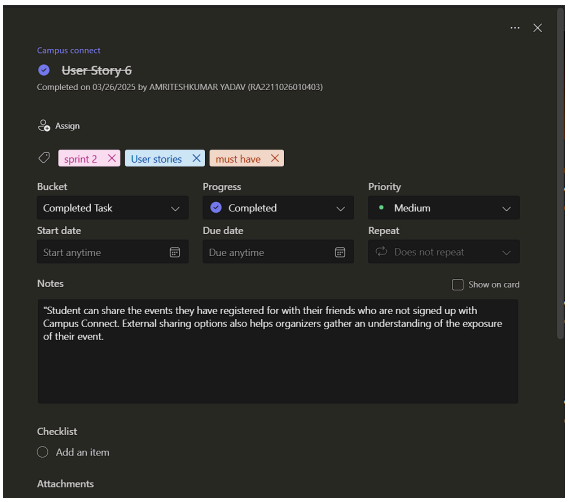


Figure 2.9 User story 6

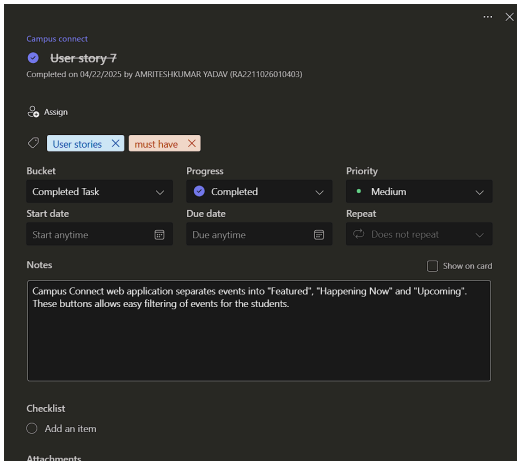


Figure 2.10 User Story 7

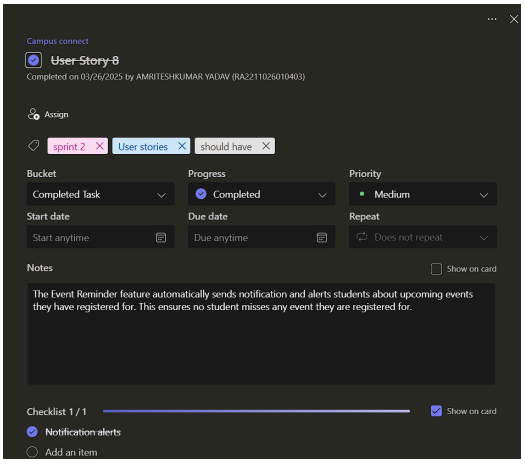


Figure 2.11 User Story 8

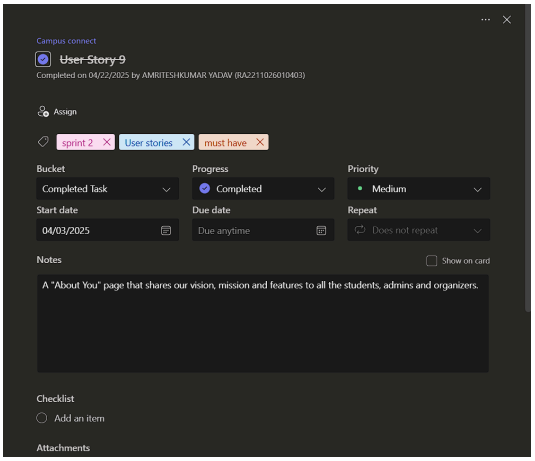


Figure 2.10 User Story 9

## 2.2.2 Functional Test Cases

| Feature | Test Case | Steps to execute test case | Expected Output | Actual Output | Status | More Information |
|---|---|---|---|---|---|---|
| **Feature** | **Test Case** | **Steps to Execute Test Case** | **Expected Output** | **Actual Output** | **Status** | **More Information** |
| Announcement Posting | Verify if admin can create and post an announcement | 1. Login as Admin<br>2. Go to Announcement section<br>3. Create and post a new announcement | Announcement appears on homepage successfully | (To be filled after test) | (Pass/Fail) | Test different announcement types (text/image) |
| Event Filtering | Verify event filtering by Featured, Upcoming, and Happening Now | 1. Navigate to Events page<br>2. Apply different filters<br>3. Observe event display | Correct filtered events should appear as per selection | (To be filled after test) | (Pass/Fail) | Check event status accuracy |
| Notifications and Reminders | Verify students receive reminders for registered events | 1. Register for an event<br>2. Wait for reminder notification<br>3. Check notification received | Notification/alert received before event starts | (To be filled after test) | (Pass/Fail) | Test reminder timing accuracy |
| About Page Content | Verify content on About page (Mission, Vision, Features) | 1. Navigate to About page<br>2. Verify the sections for Mission, Vision, and Features | Correct and complete content displayed without formatting errors | (To be filled after test) | (Pass/Fail) | Confirm responsiveness on different devices |
| External Event Sharing | Verify users can share event links externally | 1. Open an event detail page<br>2. Click Share button<br>3. Copy and share event link | Event link copied and working for external users | (To be filled after test) | (Pass/Fail) | Validate link works for both logged-in and external users |
| Registration Data Saving | Verify that event registration data is saved in database | 1. Register for an event<br>2. Check database backend for new record | New record created with correct user and event ID | (To be filled after test) | (Pass/Fail) | Check database consistency |

Figure 2.11 Functional test cases for sprint 2

## 2.2.3 Committed Vs Completed User Stories

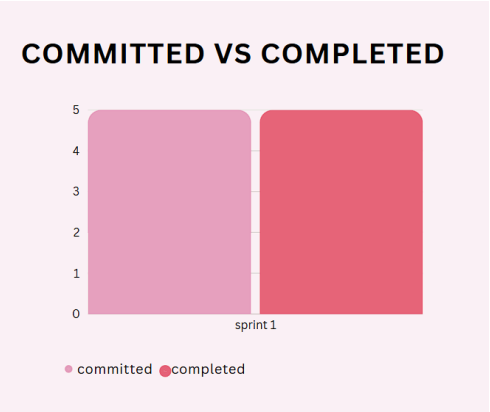


Figure 2.12 Bar graph for Committed Vs Completed User Stories for sprint 2

## 2.2.4 Sprint Retrospective

| | Sprint Retrospective | | |
|---|---|---|---|
| What went well | What went poorly | What ideas do you have | How should we take action |
| **What Went Well** | **What Went Poorly** | **What Ideas Do You Have** | **How Should We Take Action** |
| Event filtering and categorization (Featured, Upcoming, Happening Now) worked successfully | Notifications were delayed and sometimes not triggered as expected | Implement more reliable real-time notifications using better scheduling or third-party APIs | Integrate a robust reminder service and test notifications in staging before release |
| Announcements and About page content was properly structured and deployed | Some external sharing links were broken or expired early | Use dynamic link generation for sharing instead of static URLs | Add dynamic URL generation logic and verify link validity |
| Event registration saving and database updates became more stable | Minor bugs were found when users tried to register for the same event twice | Add checks to prevent duplicate registrations | Implement database constraints or frontend validations |
| UI/UX design improvements made the application more visually appealing | Dashboard responsiveness on mobile devices was not optimized initially | Design with mobile-first approach and test on different screen sizes earlier | Include mobile responsiveness as a fixed checklist item in every sprint |
| Team communication improved with clearer division of work | Last-minute rush for deploying features caused stress | Improve sprint planning and have internal deadlines before the final date | Finish main features 2-3 days before actual sprint end for safe buffer |

Figure 2.13 Sprint Retrospective for the Sprint 2

# 2.3 Execution

## 2.3.1 Functional Document

### 2.3.1.1 Introduction

Campus Connect is a full-stack web application designed to bridge the gap between college students and the various events, clubs, and opportunities available on their campuses. This platform streamlines event discovery, registration, and management for students, club organizers, and administrators. The first release aims to offer a centralized, user-friendly interface for all stakeholders to interact, register, and communicate about events, with a secure login and authentication mechanism backed by a robust database.

### 2.3.1.2 Product Goal

Campus Connect's primary objective is to make campus life more connected and organized by allowing students to discover, register, and track events seamlessly. It also equips club organizers with tools to manage registrations and announcements, while enabling administrators to monitor engagement and participation trends. The platform strives to enhance student involvement, minimize information gaps, and create a unified digital campus experience.

### 2.3.1.3 Demography (Users, Location)

**Users**

- Target Users: College students, event organizers, and campus administrators
- User Characteristics:
  - Age range: 18–28
  - Active participants in college events
  - Moderate to high digital proficiency
  - Need for organized event participation
  - Interest in student engagement and opportunities

**Location**

- Target Location: University campuses across India
- Initial Focus: Major academic institutions and student-led clubs
- Language Support: English (first release), with plans to expand to other languages

### 2.3.1.4 Business Processes

The key business processes include:

**User Registration and Authentication:**

- Secure student sign-up/login
- Email verification and password hashing
- Session/token-based authentication
- Role-based access (Student, Admin, Organizer)

**Event Discovery & Registration:**

- Filterable events by date, category, and tags
- “Featured,” “Upcoming,” and “Happening Now” classifications
- One-click event registration and RSVP tracking

**Admin Announcements**

- Admins can post updates, reminders, and opportunities
- Students receive alerts and notifications

**Dashboard & Personalization:**

- student dashboard: registered events, recommended events, alerts
- Admin/Organizer dashboard: Event creation, registration stats

**Notification System:**

- Reminders for upcoming events
- Email or in-app notifications

**2.3.1.5 Features**

### 1: Secure Authentication System

**Description**: User sign-up/login with backend integration and role-based session management

**User Story**: "I want to securely log in so my data and participation is safe and personalized."

### 2: Filterable Event Listing

**Description**: View detailed info (plot, cast, ratings, trailers, related movies).

**User Story**: "I want all movie info in one place, so I don’t have to check multiple sites."

### 3: Event Registration & Management

**Description**: Create and share editable watchlists with friends or family.

**User Story**: "I want to share and organize watchlists with friends for movie nights."

### 4: Personalized Dashboard

**Description**: Personalized suggestions based on ratings, watch history, and preferences.

**User Story**: "I want recommendations that match my tastes so I discover movies faster."

### 5: Admin & Organizer Interface

**Description**: Show streaming availability by region, with optional region switching.

**User Story**: "I want to see what’s available in my country, so I don’t get frustrated.

### 6: Notifications & Announcements

**Description:** System alerts for important updates and reminders

**User Story:** "I want alerts so I never miss an event or opportunity I care about."

### 7: Role-Based Authorization System

**Description**: Differentiated access and permissions for students, admins, and organizers

**User Story**: "I want access to features based on my role so the system stays secure."

#### 2.3.1.6 Authorization Matrix

| Feature/Module | Student | Admin | Organizer |
|---|---|---|---|
| View Events | ✅ | ✅ | ✅ |
| Register for Events | ✅ | ❌ | ❌ |
| Create Events | ❌ | ✅ | ✅ |
| Update Events | ❌ | ✅ | ✅ |
| Delete Events | ❌ | ✅ | ✅ |
| View Announcements | ✅ | ✅ | ✅ |
| Post Announcements | ❌ | ✅ | ✅ |
| Manage User Registrations | ❌ | ✅ | ✅ |
| Approve Event Registrations | ❌ | ✅ | ✅ |
| Access Dashboard | ✅ (My Events & Notifications) | ✅ (Admin Dashboard) | ✅ (Organizer Dashboard) |
| Authentication (Login/Signup) | ✅ | ✅ | ✅ |
| Manage Users | ❌ | ✅ | ❌ |

Table 2.3 Authorization Matrix

#### 2.3.1.7 Assumptions

- Users are verified students, organizers, or admins.
- Email-based login is used for authentication.
- Event data and registrations are securely stored.
- Admins and organizers manage content regularly.
- Platform is accessible on both desktop and mobile.
- Notifications and emails are reliably delivered.
- System can handle moderate user traffic.

- Role-based access is enforced.
- Data privacy and security standards are maintained.

## 2.3.2 Architecture Document

### 2.3.2.1 Architecture Overview

Watchwise is built on a microservice architecture with a layered approach, enabling modularity, scalability, and maintainability. The system consists of four primary layers:

1. Application Layer (Frontend) - User interface and client-side logic
2. Service Layer - Core business logic implemented as independent microservices
3. Data Layer - Firebase for data storage and management
4. External API Layer - Integration with TMDB and other third-party services

This architecture allows components to be developed, tested, deployed, and scaled independently while maintaining clear separation of concerns.

### 2.3.2.2 Architecture Layers

#### 2.3.2.2.1 Application Layer (Frontend)

The frontend layer provides the interface for users to interact with Campus Connect, offering tailored views for students, admins, and organizers..

**Components:**

- Web Application (Next.js with React)
- Responsive UI using Tailwind CSS
- Client-side Routing (Next Router)
- Axios for REST API calls
- Form validation (Zod)
- Dynamic components for events, registration, and announcements

**Responsibilities:**

- Render user interface based on role (Student/Admin/Organizer)
- Display events, announcements, and registration status
- Handle form submissions and client-side validation
- Interact with backend API
- Manage client-side session and route control

#### 2.3.2.2.2 Service Layer (Microservices)

This layer provides business logic and manages data processing, user auth, event operations, and announcement handling via REST APIs.

**User Service:**

- JJWT-based authentication
- Session and role management (student/admin/organizer)
- Profile handling
- Password hashing (bcrypt)
- Tech: Node.js, Express.js, Supabase/Auth

**Event Service:**

- CRUD operations for events
- Filtering by category, date, and status
- Featured and recommended events
- Tech: Node.js, Express.js, Prisma ORM

**Registration Service:**

- Register/unregister for events
- Track user-event mappings
- View registered events
- Tech: Node.js, Express.js

**Announcement Service:**

- Create, update, view announcements
- Visibility controls for different user types
- Tech: Node.js, Express.js

**API Gateway :**

- Request routing and rate-limiting
- Authentication middleware
- Logging
- Tech: Node.js, Express Middleware

**2.3.2.2.3 Data Layer**

This layer is responsible for persistent storage and data management using PostgreSQL via Supabase and Prisma ORM.

**Database (PostgreSQL via Supabase):**

- Users Table: stores user credentials, roles, and profile info
- Events Table: event details, categories, timestamps
- Registrations Table: user-event mapping
- Announcements Table: system-wide messages

**Data Models:**

- User: id, name, email, role, createdAt
- Event: id, title, date, venue, category, createdBy
- Registration: userId, eventId, timestamp
- Announcement: id, message, createdBy, visibleTo

**2.3.2.2.4 External API Layer**

While not core to the MVP, Campus Connect can be enhanced using external tools/APIs:

- **Email Services** (Resend API or SendGrid): for email notifications and OTP
- **Location API**: to filter events by location (future scope)
- **Analytics API**: to track event popularity

**2.3.2.3 System Interactions**

**2.3.2.3.1 Request Flow**

1. User performs an action (e.g., registers for event) via frontend
2. Frontend sends REST API request to backend
3. Middleware checks authentication (JWT/session)
4. Service executes logic and queries DB via Prisma/Supabase
5. Response sent back to frontend to update UI

**2.3.2.3.2 Common Scenarios**

- **Login & Auth:**
  Frontend → Auth API → Supabase (Auth) → JWT → Session maintained
- **Event Registration:**
  Student → Register Button → Registration API → Registrations Table

- **Dashboard Display:**

  Dashboard Page → Auth Check → DB Queries → Render My Events & Notifications

### 2.3.2.4 Cross Cutting Concerns

- **Authentication & Authorization:**

  JWT, Supabase Auth, role-based guards
- **Security:**

  HTTPS, input validation, encrypted passwords
- **Error Handling:**

  Standard error messages, fallback UI
- **Session Management:**

  Secure cookies or token storage
- **Monitoring & Logging:**

  Console logs, Supabase logging, Vercel Analytics
- **Scalability:**

  Supabase scaling, stateless backend design, optimized API routing

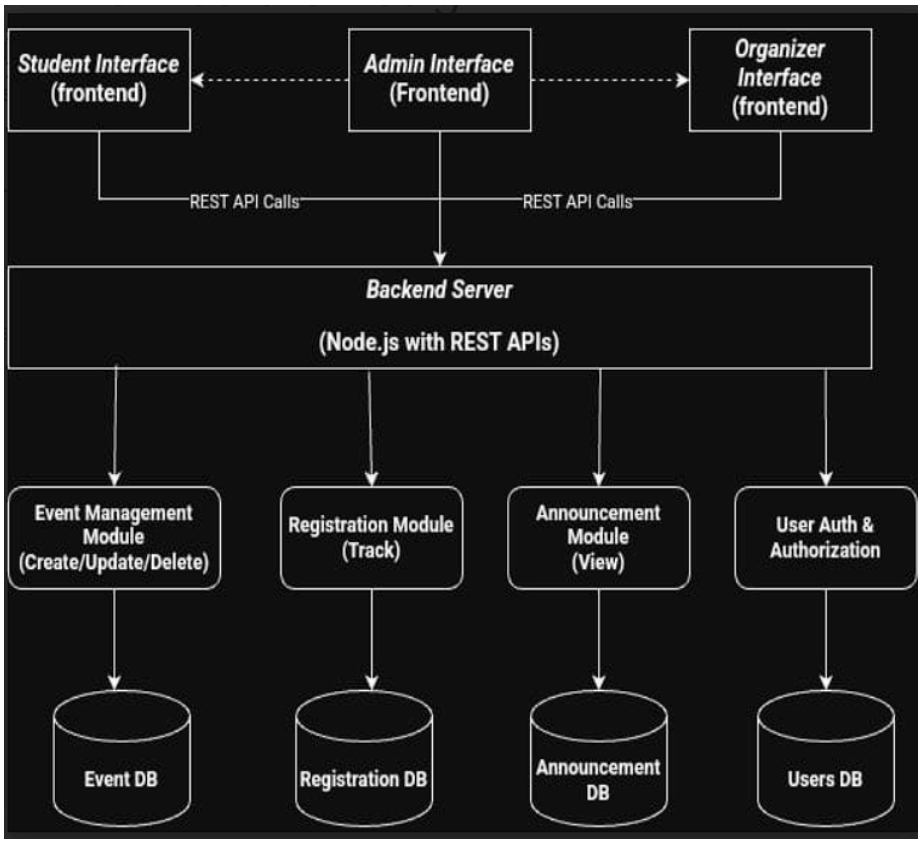


Figure 2.14 Architecture Diagram

## 2.3.3 UI DESIGN

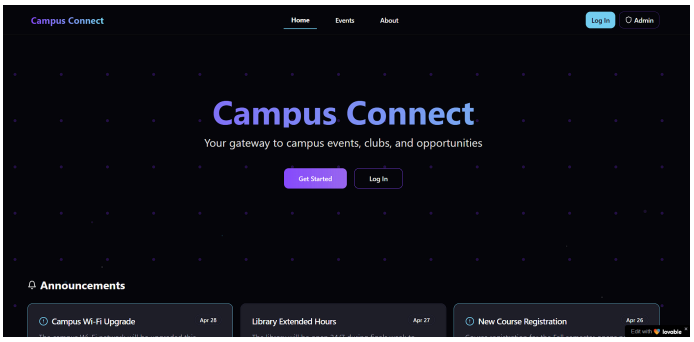


Figure 2.15 Landing Page

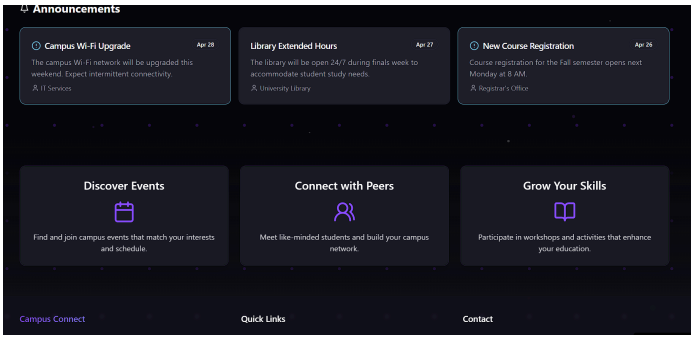


Figure 2.16 Landing Page

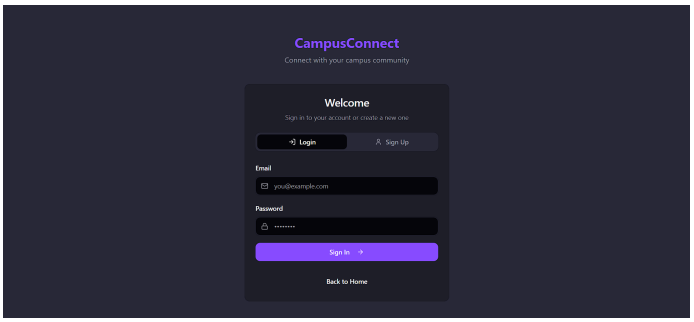


Figure 2.17 Login Page

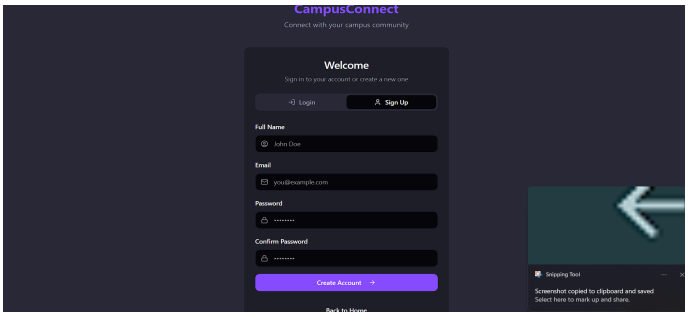


Figure 2.18 Sign Up Page

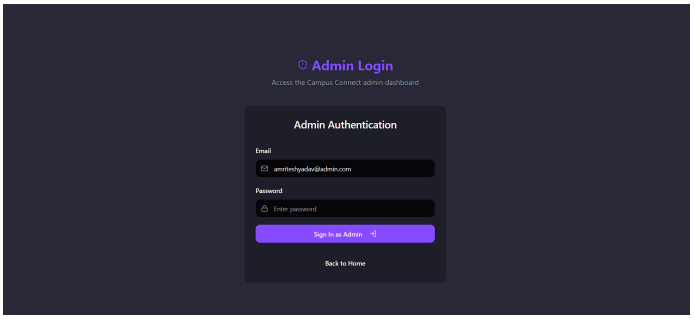


Figure 2.19 Admin Authentication

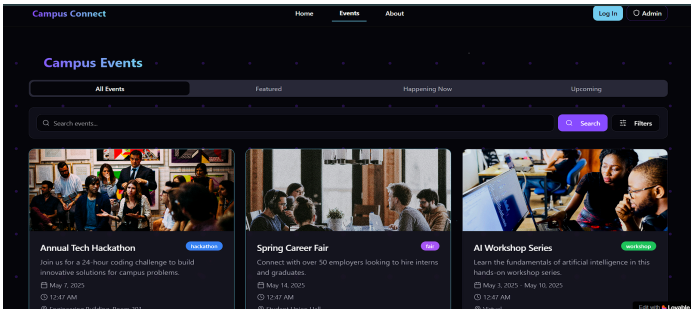


Figure 2.20 Event Search Page

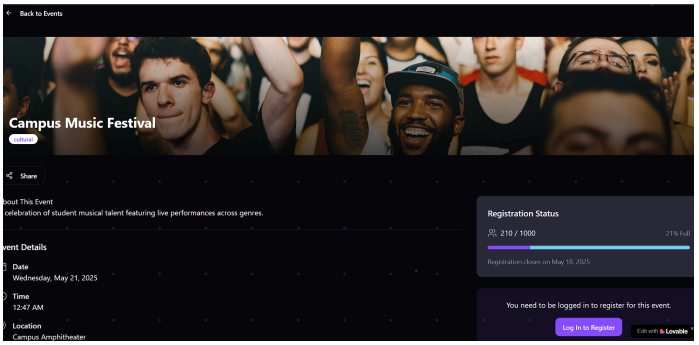


Figure 2.21 Event Booking Page

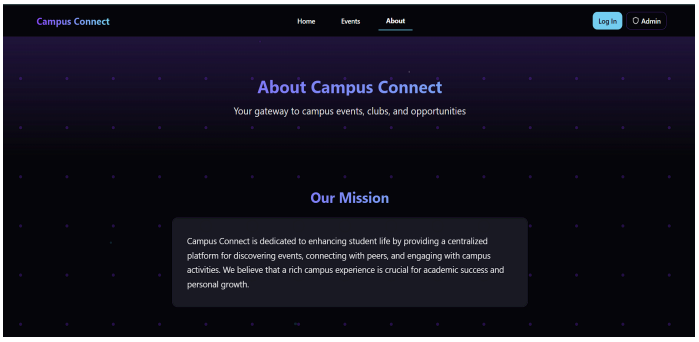


Figure 2.22 About Page

# CHAPTER 3

# RESULTS AND DISCUSSION

## 3.1 Project Outcomes

The development of Campus Connect resulted in a working full-stack web application that successfully addresses the challenges of communication and engagement in a college environment. The system offers an integrated and intuitive interface between students, staff, and event planners to interact with campus information and events in real time. With cutting-edge web technologies and secure database integration, the platform achieves a robust and scalable solution that meets academic requirements.

Key outcomes of the project include:

- Secure user authentication using email/password with JWT session handling
- Personalized dashboard for students showing events, notifications, and recommendations
- Real-time event registration and management backed by Supabase
- Role-based access for students, club admins, and external partners
- Responsive and modern UI built with Next.js and Tailwind CSS
- Fully functional deployment ready for production use in academic institutions

Overall, WatchWise fulfills its goal of streamlining the content discovery experience in the modern OTT ecosystem.

## 3.2 Committed Vs Completed User stories

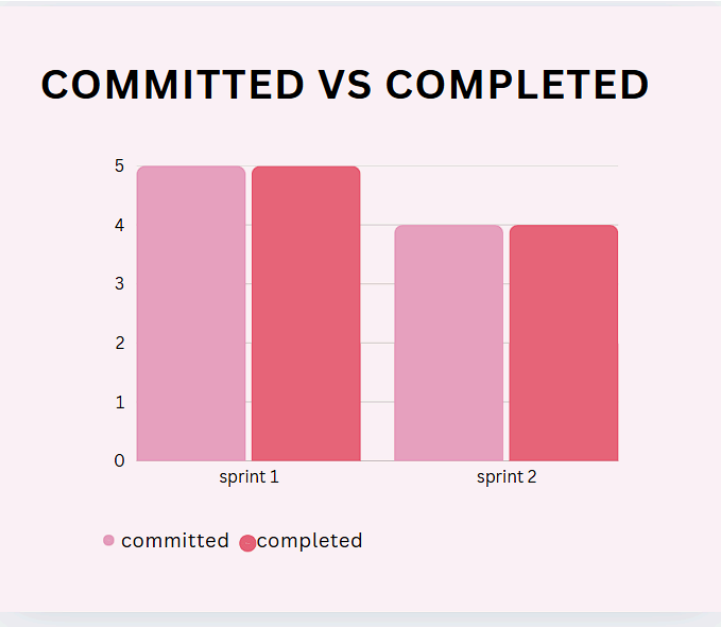


Figure 3.1 Committed vs Completed Stories

# CHAPTER 4

# CONCLUSION & FUTURE ENHANCEMENTS

## 4.1 Conclusion

Campus Connect was developed to address the need for a centralized digital platform that simplifies communication, event management, and engagement within a university setting. The project aimed to modernize how students, clubs, and faculty interact by replacing fragmented systems with a cohesive and intuitive web application.

The platform integrates secure authentication, role-based dashboards, real-time event registration, and a clean user interface, making it functional and easy to navigate. Technologies like Next.js and Supabase were effectively utilized to ensure responsive design, data security, and backend efficiency.

In summary, Campus Connect has proven to be a successful implementation of full-stack web development principles, offering a reliable, interactive, and user-friendly tool to enhance campus life and promote active participation among students and staff.

## 4.2 Future Enhancements

To further expand the platform's capabilities and user value, the following enhancements are proposed:

- Integration of real-time push notifications for event reminders and announcements
- Email verification and password reset functionality for better account security.
- In-app chat or discussion forums for student–club interaction.
- Multi-campus or department-specific filtering and personalization options.
- Integration with calendar apps (Google Calendar, Outlook) for event syncing.

These future enhancements aim to transform Campus Connect from a functional campus event portal into a smart, personalized engagement platform—one that not only informs but actively connects students with the opportunities, communities, and experiences most relevant to their academic and social journey.

# APPENDIX

## A. SAMPLE CODING

```
src > # App.css > #root
#root {
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
  text-align: center;
}

.logo {
  height: 6em;
  padding: 1.5em;
  will-change: filter;
  transition: filter 300ms;
}
.logo:hover {
  filter: drop-shadow(0 0 2em #646cffaa);
}
.logo.react:hover {
  filter: drop-shadow(0 0 2em #61dafbaa);
}

@keyframes logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

@media (prefers-reduced-motion: no-preference) {
  a:nth-of-type(2) .logo {
    animation: logo-spin infinite 20s linear;
  }
}

.card {
  padding: 2em;
```

Figure A.1 Sample Coding 1

```
src > # index.css > ...

@layer base {
  * {
    @apply border-border;
  }
  body {
    @apply bg-background text-foreground;
    background-image: url("data:image/svg+xml,%3Csvg width='30' height='30' viewBox='0 0 30 30' fill='none' xmlns='http://www.w3.org/2000/svg'%3E%3C
    background-size: 100px 100px;
  }
}

.event-card {
  @apply bg-card/70 backdrop-blur-sm transition-all duration-300 hover:bg-card hover:shadow-lg;
}

.logo-gradient {
  @apply bg-gradient-to-r from-primary to-secondary bg-clip-text text-transparent;
}

.space-card {
  @apply bg-card/80 backdrop-blur-sm border border-purple-800/30 shadow-lg rounded-xl text-white;
}

.btn-gradient {
  @apply bg-gradient-to-r from-primary to-accent text-white font-medium;
}

.bg-stars {
  position: relative;
}

.bg-stars::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
```

Figure A.2 Sample Coding 2

```
TS tailwind.config.ts > ...

import type { Config } from "tailwindcss";

export default {
    darkMode: ["class"],
    content: [
        "./pages/**/*.{ts,tsx}",
        "./components/**/*.{ts,tsx}",
        "./app/**/*.{ts,tsx}",
        "./src/**/*.{ts,tsx}",
    ],
    prefix: "",
    theme: {
        container: {
            center: true,
            padding: '2rem',
            screens: {
                '2xl': '1400px'
            }
        },
        extend: {
            colors: {
                border: 'hsl(var(--border))',
                input: 'hsl(var(--input))',
                ring: 'hsl(var(--ring))',
                background: 'hsl(var(--background))',
                foreground: 'hsl(var(--foreground))',
                primary: {
                    DEFAULT: 'hsl(var(--primary))',
                    foreground: 'hsl(var(--primary-foreground))'
                },
                secondary: {
                    DEFAULT: 'hsl(var(--secondary))',
                    foreground: 'hsl(var(--secondary-foreground))'
                },
                destructive: {
```

Figure A.3 Sample Coding 3

```
src > pages > DashboardPage.tsx > ...

import React from 'react';
import Layout from '@/components/Layout';
import { useAuth } from '@/contexts/AuthContext';
import { useNavigate } from 'react-router-dom';
import {
  Card,
  CardContent,
  CardDescription,
  CardHeader,
  CardTitle
} from '@/components/ui/card';
import { Button } from '@/components/ui/button';
import { Calendar, Users, Bell, LayoutDashboard } from 'lucide-react';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@/components/ui/tabs';
import {
  getAllEvents,
  getRegistrationsByUser,
  getEventById
} from '@/services/mockData';
import { useNotifications } from '@/contexts/NotificationContext';

const DashboardPage = () => {
  const { user, profile } = useAuth();
  const { unreadCount } = useNotifications();
  const navigate = useNavigate();

  React.useEffect(() => {
    if (!user) {
      navigate('/login');
    }
  }, [user, navigate]);

  const events = getAllEvents();
  const upcomingEvents = events
    .filter(event => new Date(event.date) > new Date())
    .sort((a, b) => new Date(a.date).getTime() - new Date(b.date).getTime())
```

Figure A.4 Sample Coding 4

# B. PLAGIARISM REPORT

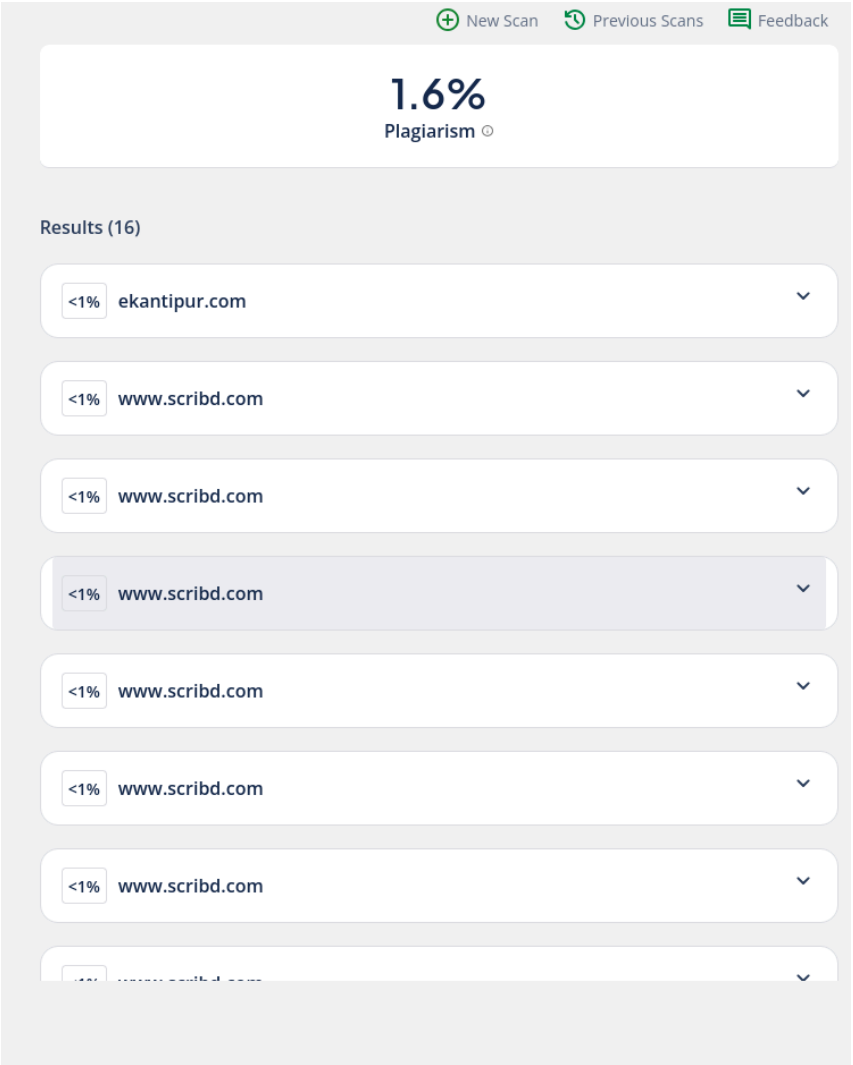


Figure B.1 Plagiarism Report