GFOSS / MyUni

Analysis and System Design Proposal For GSOC 2025

📋 Table of Contents

Introduction

- Name: Adil kadival
- Email: Adil Kadival (K-ADI)
- LinkedIn: adilkadival
- GitHub : adilkadivala
- X (twitter) : adil_kadivala
- Contact Number: (+91) 94080 18560
- First Language: English
- Institution: SMT B.K MEHTA IT CENTER PALANPUR
- Location & time zone: Palanpur, Gujarat (385210), india (GMT+5:30)

About me

I'm Adil Kadivala, a second-year computer application student with a strong background in MERN stack development and open-source contributions. I am excited to propose my participation in the Google Summer of Code (GSoC) 2025 for the MyUni project, under the mentorship of tsalmanastasios and eiosifidis, to ensure a comprehensive plan that leverages my skills in React.js, Express.js, MySQL, JavaScript, TypeScript, and Next.js.

Why am I Interested ?

- Excited to contribute to open-source.

I'm so excited to contribute to open-source, MyUni offers a chance to contribute to a platform that universities across Greece can use and enhance together. It's more than just coding—it's about creating a tool that simplifies academic life for students, faculty, and administrators. The idea that my work could positively impact thousands of people is a huge motivator.

- Opportunity for Learning and Growth.

  MyUni is a fantastic opportunity to expand my technical knowledge. From deepening my TypeScript expertise to exploring microservices and real-time data syncing, this project challenges me to grow. Plus, working with experienced mentors in the open-source world could elevate my skills to new heights.

- Solving Real-World Problems

  MyUni's mission is to build a unified, user-friendly platform. I'm excited to create something intuitive and efficient that actually works, cutting down on administrative hassles and modernizing how universities connect with their communities.

- This project can make impact to people or industry

  we're not just helping students and faculty stay connected—we're showing how technology can streamline education, cut costs, and inspire other regions to rethink their systems. I'm pumped to be part of that ripple effect, delivering something that leaves a lasting impact on how universities work.

- Perfect Fit for My Skills.

  I've worked on same stack in several projects where I've used React.js and Next.js with tailwind and TypeScript and Express.js

Here are some projects,

Postier Github (Next.js + Express.js) underdevelopment,
Finance Github (Next.js + Express.js) Completed
Ai-Content-generator Github (Next.js + Express.js) completed

Why am I fit for this project ?

This Project is Strongly align with Skill-set including authentication, real-time data, along with it's new functionalities, I'm so interested to working and contributing this project to bring it in real-life, and I got a point in this project that how data will communicate to each other, and what should be best practises for this project, I'm passionate about open-source and building solutions that matter, and I'm excited to bring that energy here to create a platform that meets its goals and wows its users. Let's do this together!

Why did I choose GFOSS ?

As a student There are many points which leaded me to choose the GFOSS,

- Mission alignment: This project Idea will surely help students in their academic life with university's professors, so I want to fill a gap which exists in academic management by reducing paperwork. outdated systems universities face a headache with it, so want to create or contribute to projects which solve this.

- Impact and Growth: I want to make such an impact ful contribution in open-source which lead myself to grow in technologies , and I got this chance in Gfoss,

Availability During the Coding Period

The only prior commitment I have is an important **exam from April 7th to April 17th**.

Outside of this period, I will be fully dedicated to the project, **contributing 6-7 hours per day.**

System Analysis

Current System Limitations

Limitation	Description	Impact
Inadequate Authentication	Lacks robust login mechanisms	Security vulnerabilities and limited user management
Limited Scalability	Architecture not designed for growth	Performance issues with increasing users
Manual Updates	No real-time data synchronization	Outdated information and administrative burden
Fragmented Implementation	Separate systems for each university	Duplicated development efforts and inconsistent experiences
Limited Customization	Rigid structure difficult to adapt	Cannot meet specific university requirements
Poor Integration	Minimal connection to university systems	Manual data entry and synchronization issues

Users Analysis

Stakeholder	Needs	Expectations
Students	Access to academic information, schedules, announcements	User-friendly interface, real-time updates, personalized content
Faculty	Course management, student communication	Administrative tools, content management capabilities
University Administrators	Platform customization, content management	Intuitive admin panel, branding options, analytics
IT Departments	System maintenance, integration	Well-documented code, scalable architecture, security features
Other Universities	Platform adoption and customization	Modular design, easy implementation, customization options

Feasibility Study

Technical Feasibility

The project is technically feasible using the specified technologies:

Frontend: React.js (vite) or Next.js, TypeScript, preferred Next.js with TypeScript for better SSR, and React.js (cra) is deprecated.
Backend: Express.js,
Database: MySql,
Authentication: JWT based authentication
Load balancing: NginX,

Caching: Redis
Notification: Resend
Multi-Domain Support: NginX Virtual Host

These technologies are well-established with robust communities and documentation. The TypeScript rewrite will enhance maintainability and scalability.

Operational Feasibility

The system will integrate with existing university data sources through standardized APIs. The modular architecture allows for:

- Phased implementation
- Customization flexibility
- Scalability for growing user bases
- Centralized updates with distributed customization

Economic Feasibility

As an open-source project, MyUni offers economic advantages:

- Shared development costs across universities
- Elimination of licensing fees
- Reduced maintenance burden through collaboration
- Resource optimization through standardization

Schedule Feasibility

The project can be completed within the GSoC timeframe of 350 hours over 12-14 weeks, with an average of 25-30 hours per week.

Requirements Specification

Functional Requirements

Authentication and User Management with Rolebase access system

- The system will authenticate users against university identity providers
- The system will support role-based access control
- The system will allow users to manage their profiles
- The system will provide secure password reset functionality

Content Management

- The system will provide tools for content creation and editing
- The system will support various content types (text, images, files)
- The system will allow scheduling of content publication
- The system will maintain version history of content

Data Integration

- The system will connect to university data sources via APIs
- The system will synchronize data in real-time or at scheduled intervals
- The system will transform external data to match internal schema
- The system will log synchronization activities

Customization

- The system will allow branding customization (logos, colors, fonts)
- The system will support custom homepage layouts with widgets
- The system will enable feature toggling for different universities
- The system will permit custom CSS and theming

Non-Functional Requirements

Performance

- The system will load pages within 2 seconds under normal conditions

- **The system will support at least 10,000 concurrent users**
- **The system will handle peak loads during registration periods**

Security

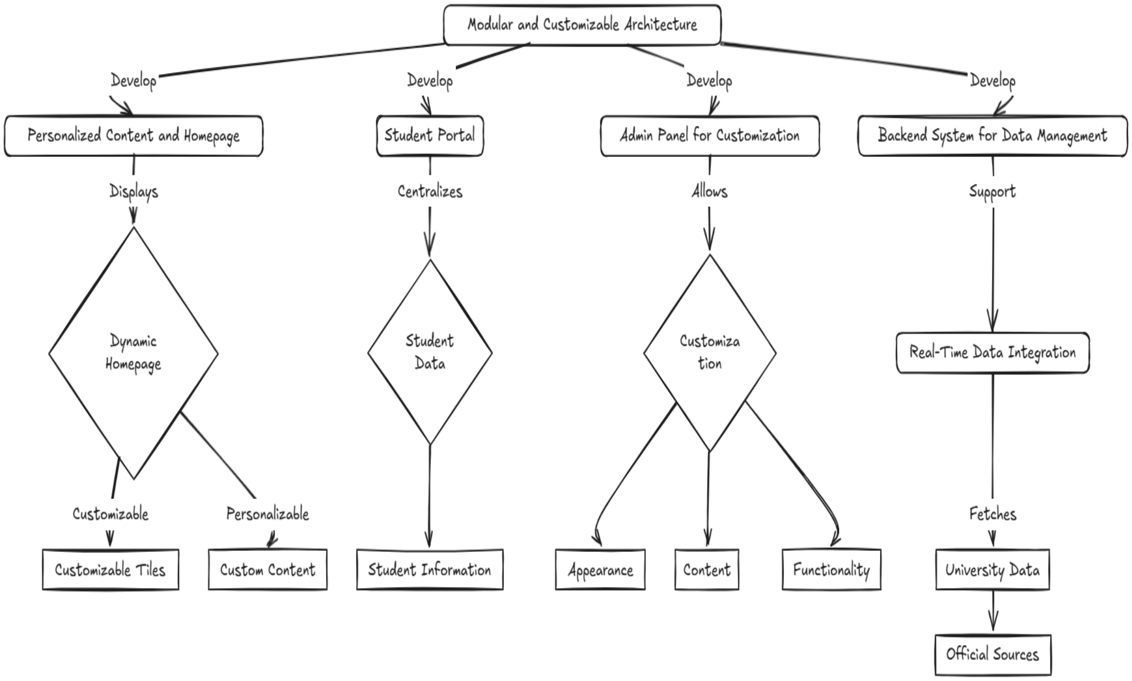
- **The system will encrypt all sensitive data in transit and at rest**
- **The system will implement OWASP security best practices**
- **The system will undergo regular security audits**
- **The system will comply with GDPR requirements**

Scalability

- **The system will scale horizontally to accommodate growing user bases**
- **The system will support multiple university instances**
- **The system will handle increasing data volumes efficiently**

Maintainability

- **The system will be well-documented**
- **The system will follow clean code principles**
- **The system will implement automated testing**
- **The system will use containerization for deployment**



User Stories

Student User Stories

ID	User Story	Acceptance Criteria	Priority
S1	As a student, I want to view my class schedule so that I can plan my day.	- Schedule displays all enrolled courses - Shows time, location, and instructor - Allows filtering by day/week	High
S2	As a student, I want to receive real-time announcements so that I stay informed about important events.	- Notifications appear for new announcements - Announcements are categorized by type	High

ID	User Story	Acceptance Criteria	Priority
		- Can mark announcements as read	
S3	As a student, I want to access my academic records so that I can track my progress.	- Shows current GPA and credits - Lists all courses with grades - Displays degree requirements progress	High
S4	As a student, I want to customize my homepage so that I can prioritize relevant information.	- Can add/remove/rearrange widgets - Settings persist between sessions -can set theme (light / dark) - Provides default layout for new users	Medium

Faculty User Stories

ID	User Story	Acceptance Criteria	Priority
F1	As a faculty member, I want to post announcements so that I can communicate with my students.	- Can create, edit, and delete announcements - Can target specific courses or departments - Can schedule future publication	High

ID	User Story	Acceptance Criteria	Priority
F2	As a faculty member, I want to view my teaching schedule so that I can prepare for classes.	- Shows all assigned courses - Displays classroom locations and times - Indicates number of enrolled students	High
F3	As a faculty member, I want to access student information so that I can track attendance and performance.	- Lists students enrolled in each course - Shows student contact information - Displays academic standing	Medium
F4			

Administrator User Stories

ID	User Story	Acceptance Criteria	Priority
A1	As an administrator, I want to customize the university's app interface so that it aligns with our branding.	- Can upload university logo - Can set primary and secondary colors - Can customize homepage layout	High
A2	As an administrator, I want to manage user permissions so that appropriate access is granted.	- Can create and modify user roles - Can assign roles to users	High

ID	User Story	Acceptance Criteria	Priority
		- Can define granular permissions	
A3	As an administrator, I want to configure data integration points so that information is accurate and up-to-date.	- Can set up API connections - Can schedule synchronization intervals - Can map external data to system fields	High

EPICS

EPIC 1: Authentication and User Management

- Implement secure login system
- Create user profile management
- Establish role-based access control
- Develop multi-factor authentication

EPIC 2: Personalized User Experience

- Design customizable homepage
- Implement personalized content delivery
- Create notification system
- Develop user preference settings

EPIC 3: Content Management System

- Build admin CMS interface
- Develop content creation and editing tools
- Implement content categorization and tagging

- Create approval workflows

EPIC 4: Data Integration and Synchronization

- Establish connections to university data sources
- Implement real-time data fetching
- Develop data processing and storage
- Create synchronization monitoring

EPIC 5: Multi-University Framework

- Design multi-domain architecture
- Implement university-specific customization
- Develop onboarding process
- Create documentation for adoption

EPIC 6: Theme and Style Management

- User Story: As an admin, I want to change the color schemes for the different instances.
- User Story: As an admin, I want to manage all the assets of the page.
- User Story: As a user(student or admin), I want to get the UI based on my university.

EPIC 7: Access management

- Admin can be able to reset the password of the users.
- User can able to access the application with all features working correctly under his/her role

EPIC 8: Load-balancing and Caching

- System can be able to balance load equally
- Server should be capable to cache and server client request effectively

System Backlog

High Priority

1. TypeScript rewrite of existing codebase
2. Authentication system implementation
3. Core backend architecture development with load-balancing and caching
4. Basic CMS functionality
5. Data integration framework

Medium Priority

1. Personalized homepage implementation
2. Admin panel for customization
3. Student portal development
4. Multi-domain support
5. Notification system

Low Priority

1. Advanced analytics
2. Responsive web app for multiple devices
3. Integration with additional university services
4. Advanced customization options
5. Community contribution framework

Architecture Database Design

The MyUni database is organized into several logical modules:

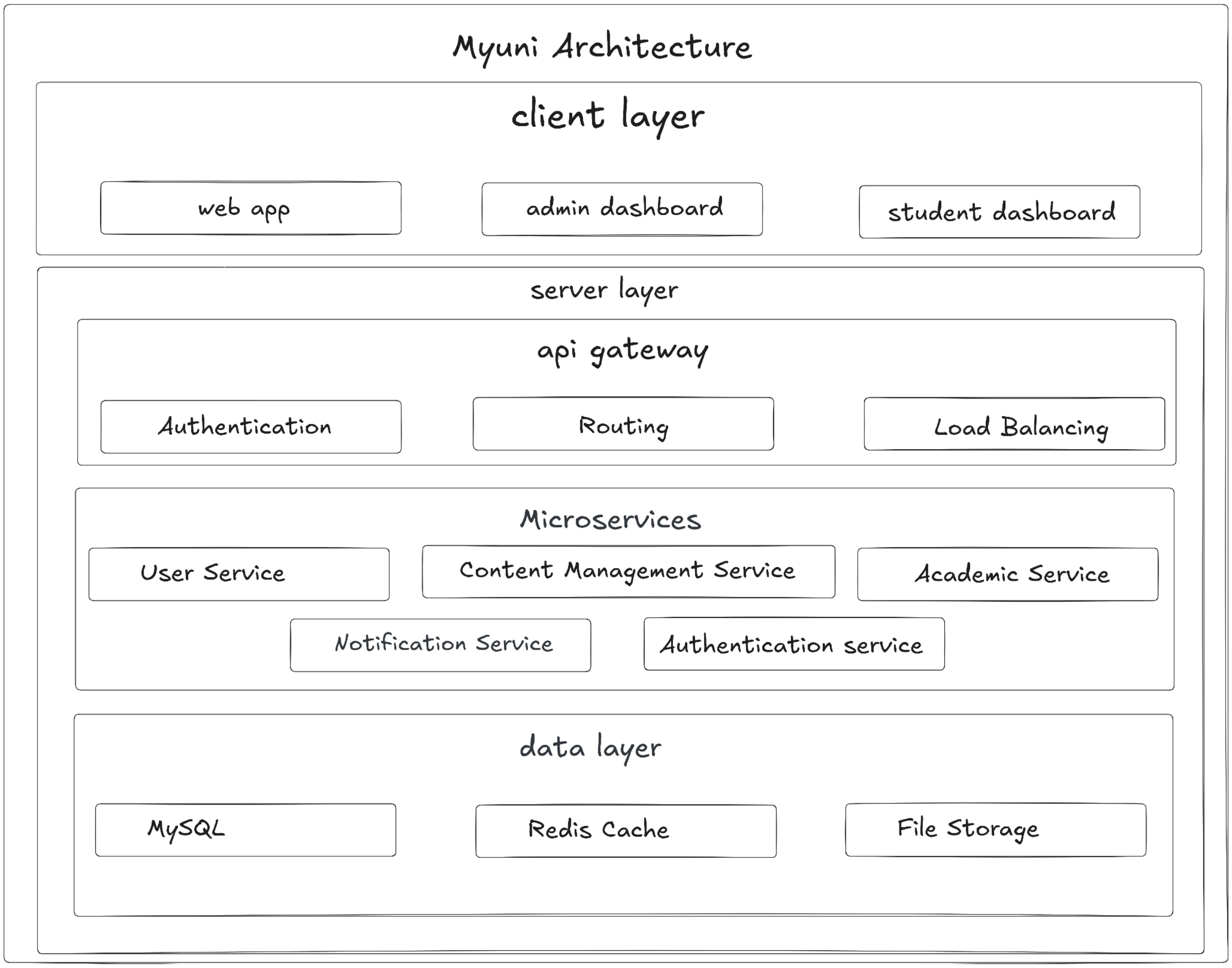
1. Core System: Users, roles, and authentication
2. Academic Structure: Departments, programs, and courses
3. Student Records: Enrollments, grades, and transcripts

4. Content Management: Announcements, events, and resources
5. System Configuration: Settings and configurations

Wire-frame & System communication Architecture

High Fidelity wireframe is here

MyUni Architecture diagram



The MyUni platform will follow a microservices architecture with the following components:

1. **Frontend Layer**
2. **React.js or Next.js with TypeScript**
3. **Responsive design for multiple devices**
4. **Component-based structure for reusability**
5. **API Gateway**
6. **Request routing and load balancing**
7. **Authentication and authorization**
8. **Rate limiting and caching**
9. **Microservices**
10. **User Service (authentication, profiles)**
11. **Content Service (CMS, announcements)**
12. **Academic Service (schedules, records)**
13. **Integration Service (data synchronization)**
14. **Notification Service**
15. **Data Layer**
16. **MySQL for structured data**
17. **Redis for caching**

18. NginX for load-balancing

19. File storage for documents and media

20. Docker containers

Data Flow

- User requests are received by the frontend
- Requests are routed through the API gateway
- Appropriate microservices process the requests
- Data is retrieved from or written to the data layer
- Responses are returned to the frontend
- Real-time updates are pushed via WebSockets

Database Schema

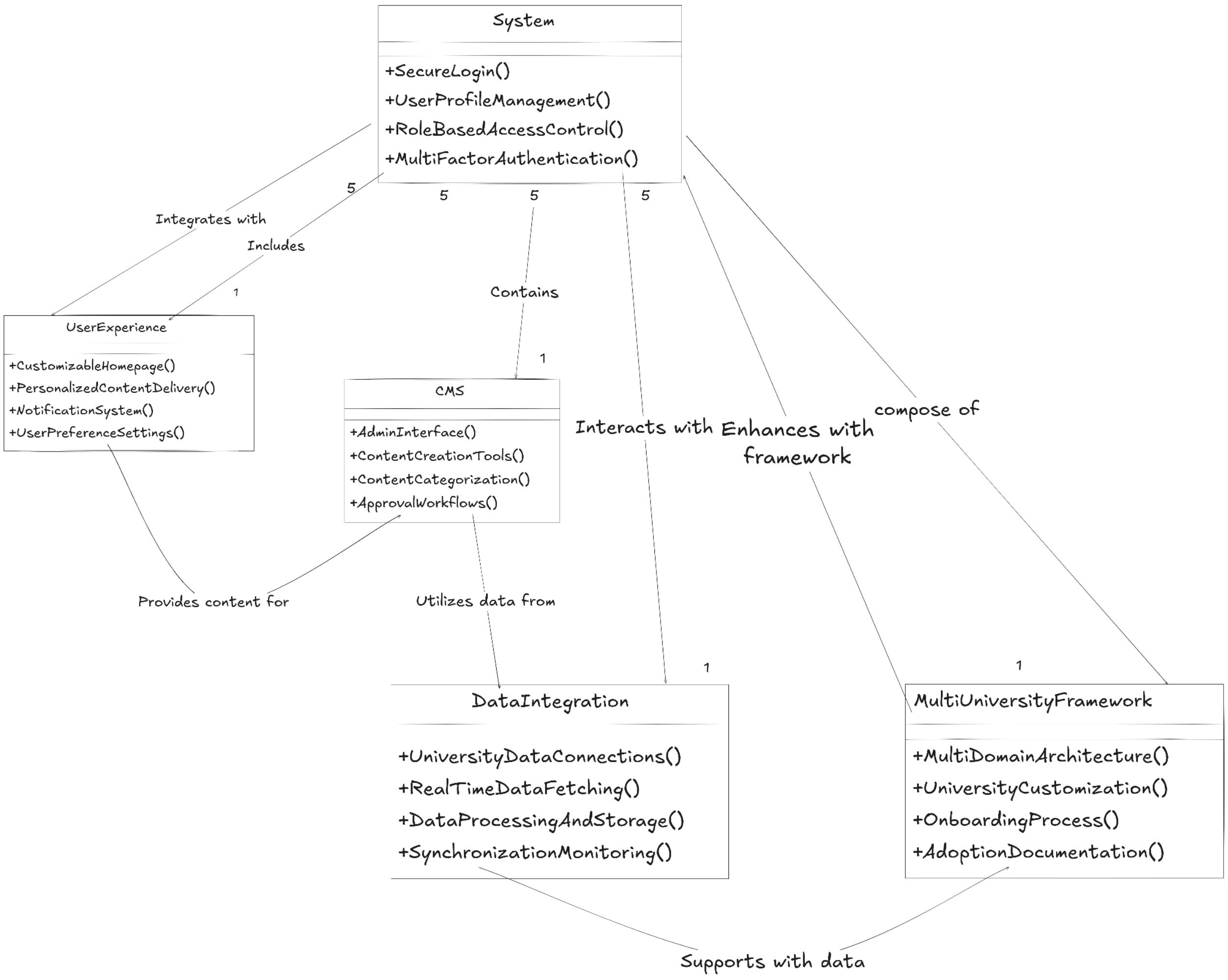
The database schema will include the following key entities:

- Users: Authentication and profile information
- Roles: Permission definitions
- Universities: Institution-specific configurations
- Content: CMS-managed information
- Courses: Academic course details
- Schedules: Class and event timing
- Announcements: University and course notifications
- UserPreferences: Personalization settings

The CMS Interface provides:

- Content creation with any suitable and light text-editor
- Content categorization and tagging system

- **Publishing workflow with draft, review, and publish states**
- **Content scheduling for timed publication**
- **Media library for image and file management**
- **Search and filtering capabilities for content management**



Deliverables:

- **Web application that provide good user experience(UX) and has good User Interface(UI)**
- **Admin dashboard where myuni admin can able to handle CMS and more**

- **Student dashboard where student can able to take-care of their profile, notifications of academics, and more**
- **Backend Architecture which serves as solution to all project requirements**
- **REST APIs with at least 95% test coverage**
- **REST APIs with modern design patterns**
- **Docker container image of entire app**

Implementation Plan (350 Hours)

Community Bonding Period (1 week, 20 hours)	● **I'll Meet with mentors and community** ● **I'll Study existing codebase and documentation** ● **I'll Set up development environment next.js or react-vite with type-script and tailwind in frontend and express.js in server** ● **I'll Refine project plan and requirements** ● **I'll create detailed wireframes for all planned screens (Landing Page,Admin Dashboard, Student Dashboard, CMS) to align with**

	my mentors on the UI/UX approach. • I'll develop a comprehensive test plan covering backend APIs, and frontend components to ensure robust quality throughout development.

Coding Period 1 (4 weeks, 120 hours)

Week 1-2: **Core Infrastructure (60 hours)**	• I'll Begin TypeScript rewrite of existing codebase • I'll Implement authentication service • I'll Design database schema • I'll Set up CI/CD pipeline
Week 3-4:	• I'll Develop user profile management • I'll Create role-based access control

Authentication and User Management (60 hours)	• **I'll Build login and registration interfaces**

Coding Period 2 (4 weeks, 120 hours)

Week 5-6: Backend Services (60 hours)	• **I'll Develop Content Service for CMS** • **I'll Implement Academic Service for student data** • **I'll Create Integration Service for external data sources** • **I'll Create authentication & authorisation** • **I'll add NginX for load-balancing** • **I'll add redis for caching**
Week 7-8: Frontend Development (60 hours)	• **I'll Build responsive UI component library** • **I'll Implement Student Dashboard** • **I'll Create Admin Dashboard with fully support of CMS with editor**

Coding Period 3 (3 weeks, 80 hours)

Week 9-10: CMS and Customization (50 hours)	- I'll Complete CMS interface - I'll Implement university customization features - I'll Develop multi-domain support
Week 11: Testing and Refinement (30 hours)	- I'll Write unit and integration tests - I'll Perform security audit - I'll Optimize performance

Final Week: **Documentation and Submission (1 week, 30 hours)**	- I'll Create user and administrator documentation - I'll Prepare Docker deployment configuration - I'll Finalize code comments and API documentation - I'll Prepare final submission and presentation

Post GSOC plan

Once GSOC is near to end, I'm not just walking away, I want to stick around and keep it uptodate, I've many points to cover after GSOC.

- **Collecting Feedback from myuni users : The first and main point is I want to collect feedback from myuni users, this will help us to improve our project as per our user's need and we can serve actually all those functionality which is actually helpful.**
- **Enhancing performance : I am logging performance and making it more accurate, so users can feel smooth using this project.**
- **Ai agent : now in Ai time everybody talks about it, I've this idea comes many time in mind that myuni should have ai agent for student who responsible to lead student towards success, I'd start small, maybe prototyping it with basic NLP, and build from there with community feedback.**

Finally I'll be more happy if any other idea represents me from mentors to implement it.