# Report 3: Core Feature Development Report (Development Phase)

## Development Progress Overview

The Student Freelance platform has been successfully developed as an ASP.NET Core web application. Most of the core planned functionalities have been implemented, including user management, project management, application processing, payment system, and communication features. The system includes robust authentication, authorization, and a clean architecture with separate layers for presentation, logic, and data.

## Implemented Features

**User Management:** Registration, login, role-based authorization (Admin, Moderator, Student, Business), profile editing, and verification.

**Project Management:** Create, update, delete projects; set categories and required skills; manage project status and attachments.

**Application System:** Students can apply; businesses can accept/reject applications; track application status.

**Payment & Wallet System:** User wallet management, project budget control, transaction history, VNPay integration, and bank account linking.

**Communication System:** Real-time chat (SignalR), system notifications, email alerts.

**Rating & Review:** Users can rate each other after project completion.

**Search & Filter:** Advanced filtering by categories, skills, and more.

**Reporting System:** Users can report projects or others; admins can review and manage reports.

**Advertisement Management:** Businesses can create/manage ads and packages.

**AI CV Generation:** Allows students to automatically generate a professional CV. The system analyzes their profile data (skills, experience, education) and formats it into a downloadable document using AI.

## Technical Implementation

**OOP Principles:** The project uses encapsulation, inheritance, abstraction, and polymorphism across services, models, and interfaces.

**EF Core:** Entity Framework Core is used for data access, with DbContext, entity relationships, migrations, and the Repository Pattern.

**ASP.NET MVC:** The project follows the MVC pattern, with Models, Views, Controllers, and ViewModels for data transport.

**Multithreading:** Asynchronous programming is used extensively (async/await) for database access and external operations. SignalR is used for real-time chat.

## Challenges and Solutions

**Complex User Roles:** Managed using ASP.NET Identity with role/claim-based access and a custom middleware to check status and permissions.

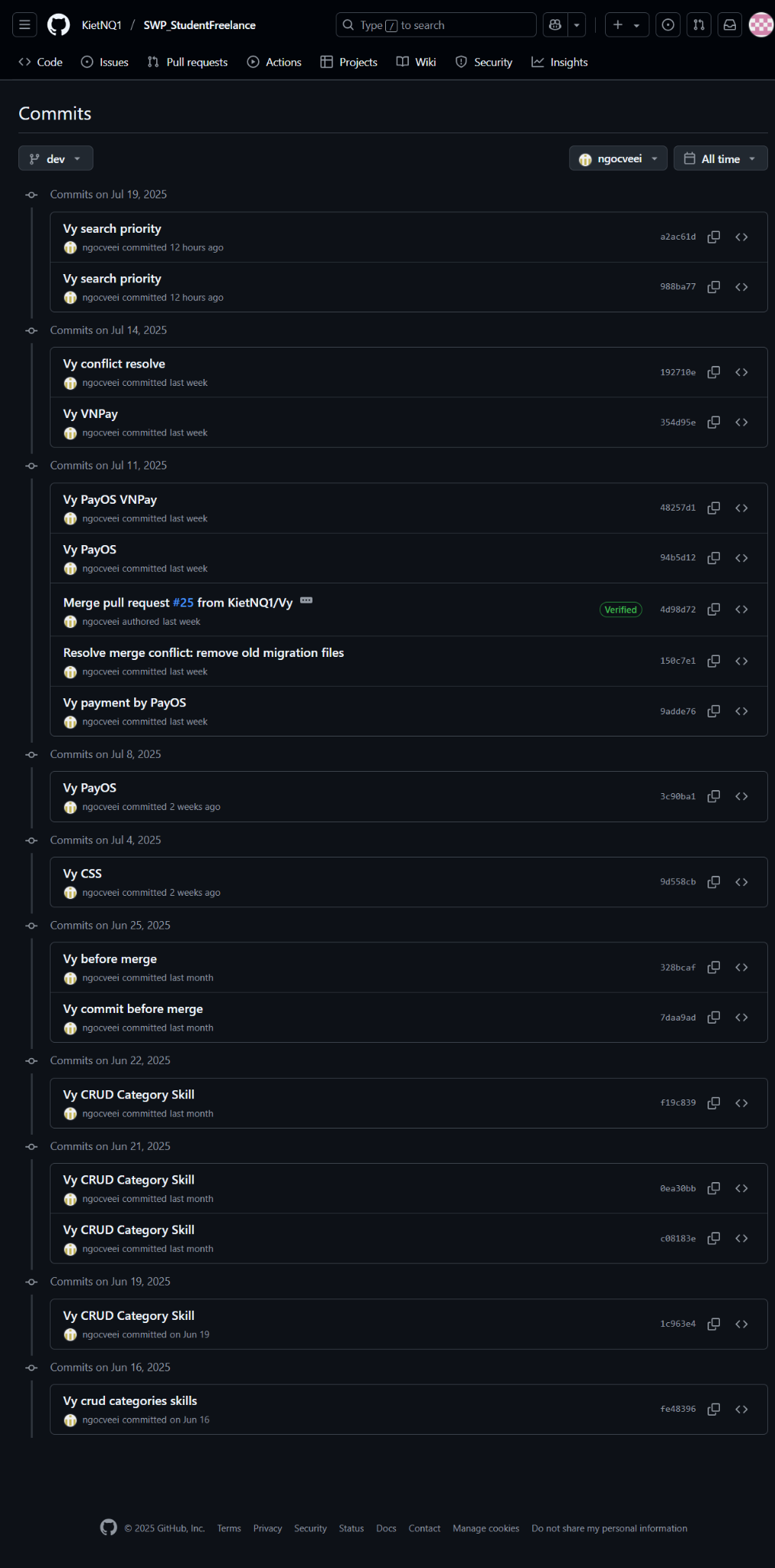
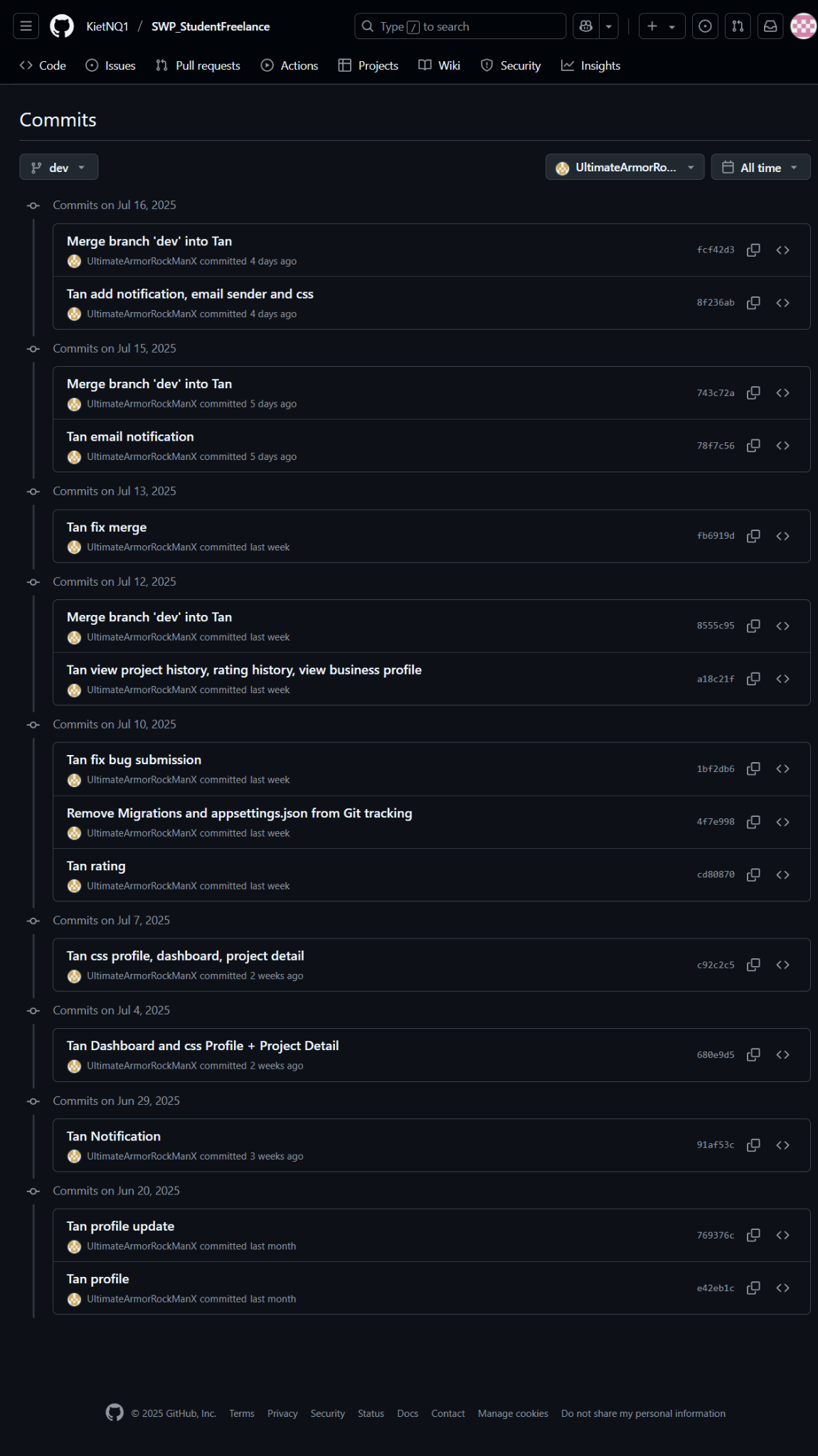
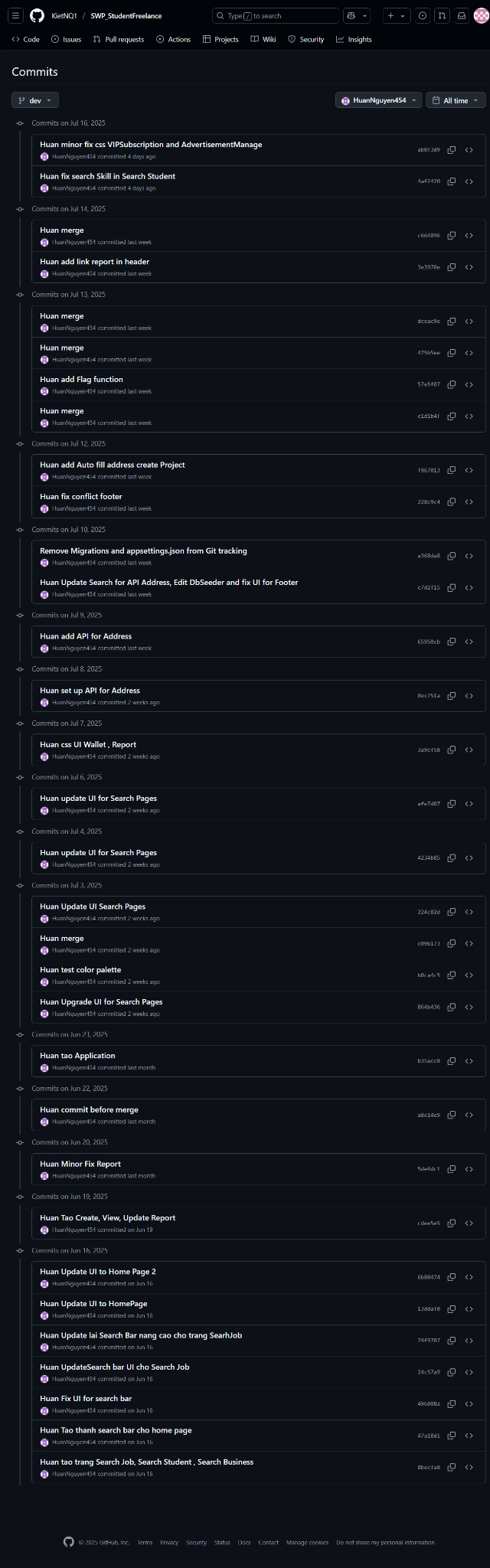
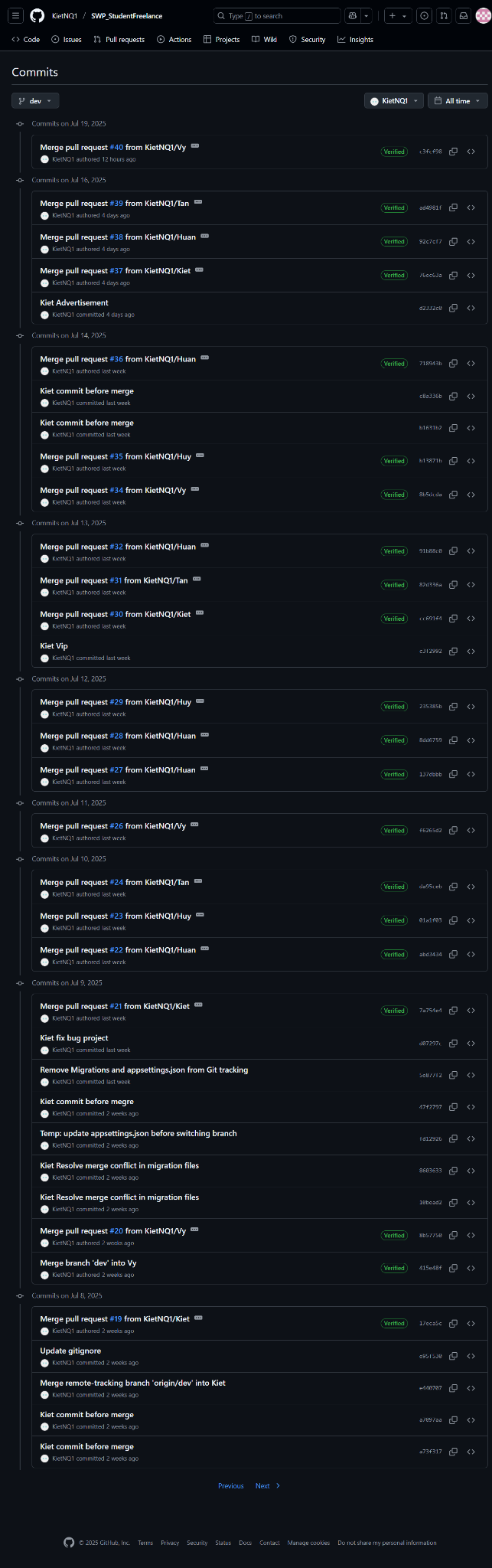
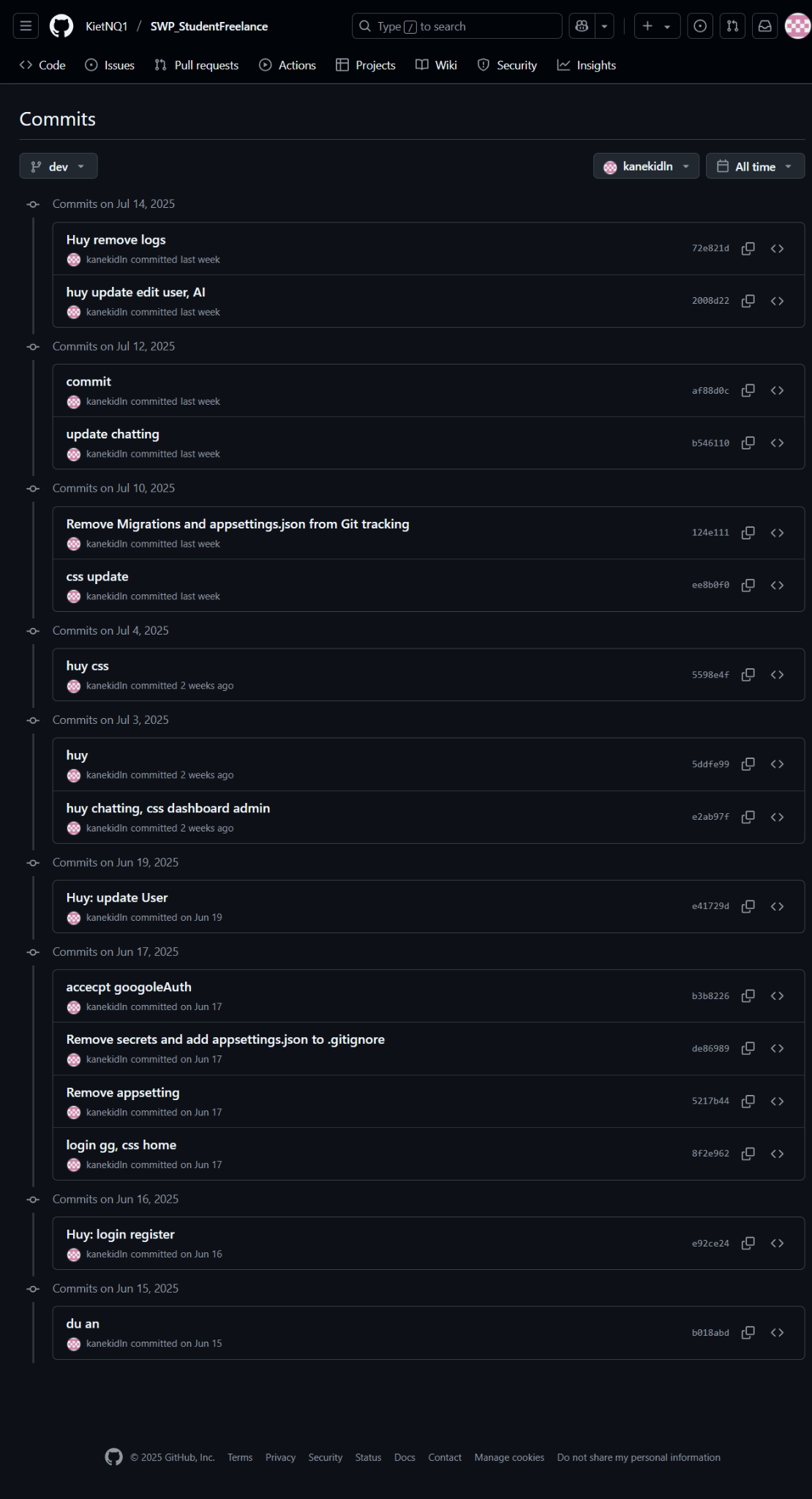
**Payment System:** Developed a secure wallet service with transaction management and integrated VNPay for external payments.

**Real-Time Features:** Implemented chat and notifications using SignalR with custom services.

**File Upload Handling:** Built a secure file upload service with validation and directory management.

## Git Commit History

Team members contributed consistently to different features. Git commit history shows clear division of work across user management, project logic, payment processing, and communication. Commit links or screenshots were provided to verify contributions per member.

* 
* 
* 
* 

## Code Quality and Documentation

Consistent naming conventions (PascalCase for classes, camelCase for variables).

Organized folder structure separating Models, Services, Controllers, Views.

Dependency Injection is used across the app.

Code is well-commented, especially in areas with complex logic.

## Testing Activities

Manual Testing: Conducted for registration, login, project CRUD, application flow, and payments.

Edge Case Testing: Focused on invalid inputs, permission issues, and wallet validations.

Known Issues:

Some UI responsiveness issues on mobile devices.

Occasional delays in real-time notifications.

File size limits for uploads may need to be increased.

## Next Steps

* Integrate all core modules into a seamless system.
* Add analytics and recommendations based on user skills.
* Improve mobile responsiveness and accessibility.
* Optimize database performance.
* Add automated testing coverage.
* Introduce additional payment gateways.
* Expand admin dashboard functionality.