Student Trade

Systems III Project Report

Hristijan Chupetreski
89211097@student.upr.si
UP FAMNIT
Koper, Slovenia

PROJECT LINKS

- **GitHub Repository:** https://github.com/cupetre/student-trade
- **Project Demo Video:** https://youtu.be/6KHKfpf9LKw
- **Deployed Website:** 88.200.63.158:3131

1 INTRODUCTION

The life of a student is often defined by transitions, from starting at a new university to relocating for an internship or moving abroad. These frequent changes bring a unique set of challenges, particularly when it comes to finding affordable necessities. Students are typically on a tight budget, and the costs of furnishing a new living space can quickly become overwhelming. The stress is compounded by the effort of acclimating to a new environment, navigating a new culture or language, and managing a demanding academic workload.

Currently, many students are forced to rely on general online marketplaces and social media groups to find affordable items. These platforms, however, are often rife with scams, dishonest listings, and a flood of digital noise that makes it difficult to discern legitimate offers from fraudulent ones. This can lead to rushed decisions that result in wasted time and money, making students vulnerable to a flawed system.

Student Trade was born from a desire to solve this problem. As a student myself, I experienced these frustrations firsthand and recognized the need for a better solution. This report outlines a dedicated platform designed to provide a safe, simple, and trustworthy environment where students can buy, sell, and trade items with their peers. By creating a community-focused marketplace, Student Trade aims to eliminate the risks associated with general platforms, empowering students to find what they need affordably and securely.

2 CORE FUNCTIONALITIES

This section comprehensively details each of the key functionalities implemented in the **Student Trade** application. For each feature, a thorough description is provided alongside the primary technical challenges encountered during development and clear instructions for users on how to utilize these functionalities effectively.

2.1 User Authentication (Login, Register, Logout)

Description. The authentication system is fundamental for secure platform access, enabling users to create accounts, authenticate with existing credentials, and securely end their sessions. This protects user data and maintains the integrity of interactions within the marketplace. Registration captures essential user information, login verifies credentials to grant access, and logout safely invalidates sessions to prevent unauthorized access.

Most Difficult Part of Implementation. Ensuring the security of user credentials was the most critical and complex aspect. To achieve this, a dual-layer security approach was implemented. First, passwords are hashed using Bcrypt, significantly reducing the risk of brute-force or rainbow table attacks. Second, session management is handled via JSON Web Tokens (JWTs), which offer stateless authentication. The implementation required careful design to manage token issuance, expiration, and secure renewal without compromising user experience or security. Handling token invalidation on logout and protecting against token theft or replay attacks demanded rigorous attention.

Instructions for Use.

- **Register:** Click the "Register" button on the homepage or navigation bar, then fill in all required fields such as username, email, and password. Upon submission, an account will be created.
- **Login:** Navigate to the login page and enter your registered email and password. If authentication is successful, you will be granted access to your personalized dashboard.
- **Logout:** To end your session, click on your profile icon located at the top-right corner and select "Logout" from the dropdown menu, which securely terminates your session.

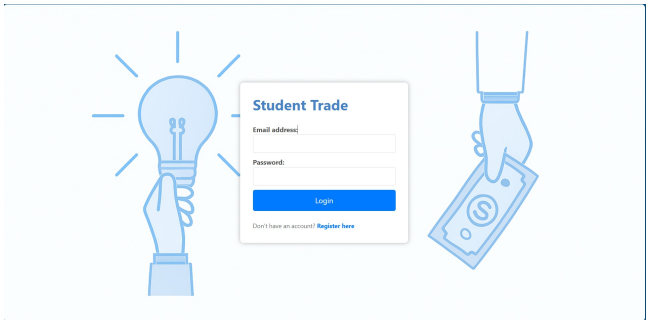

Figure 1: LoggingIn

2.2 User Profile Management

Description. This functionality empowers users to maintain accurate and up-to-date personal information, including their display name and profile picture. A well-maintained profile fosters trust among users and personalizes their experience on the platform. The

system supports image uploads and editing of textual information seamlessly.

Most Difficult Part of Implementation. The main challenge was implementing a reliable and secure image upload pipeline. The backend had to accept file uploads, validate file types (to prevent malicious files), enforce size limits for optimal storage and bandwidth use, and store images securely on the server. I implemented and used multer in order to give the uploads of pictures a directory to be uploaded to and saved. The database saves the path of the picture in to the local storage of the project folder and then when it needs to be pulled/viewed, the place and exact pictures is fetched from both the database(path+name) and previews the photo from the folder.

Instructions for Use.

- Access your profile page by clicking your username or profile icon.
- Click the "Edit Profile" button to modify your personal details.
- Update your name, email, or upload a new profile picture using the provided form.
- Save your changes. The updated information, including your new picture, will immediately reflect across the platform.

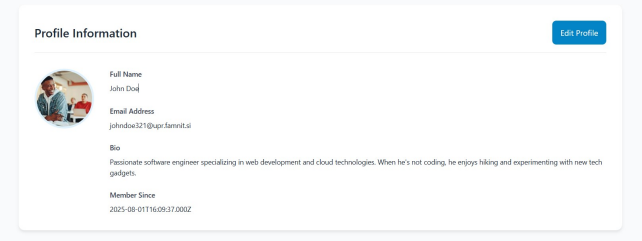

Figure 2: EditingProfile

2.3 Listings Creation and Management

Description. Listings represent the core marketplace functionality, allowing users to post items for sale, update item details, delete listings once items are sold, and browse other users' items. This system facilitates the buying and selling process by providing a structured and searchable inventory of items, complete with images, prices, and seller information.

Most Difficult Part of Implementation. Managing multiple concurrent listings efficiently in the frontend was the biggest technical hurdle. The application needed to handle CRUD operations (Create, Read, Update, Delete) on potentially large datasets while ensuring real-time UI updates without requiring page reloads. This required precise React state management, optimization to prevent performance bottlenecks, and synchronization with backend APIs. Additionally, image handling for listings and ensuring consistency across distributed clients posed challenges. The system also needed to handle edge cases like partial updates, error states, and optimistic UI updates for better responsiveness.

Instructions for Use.

- **Post an Item:** Click the "Create New Post" button prominently displayed on the dashboard or listings page. Fill in all required details, including the item title, description, price, and upload clear images of the item. Submit the form to add your listing to the marketplace.
- **Edit/Delete an Item:** Navigate to your profile to view all your active listings. Select a listing to access detailed controls, then use the "Edit" button to modify information or the "Delete" button to remove the listing from the platform.
- **View Items:** Browse through the main feed or use the integrated search bar to find items. Listings display key details such as the item's title, price, seller's name, and upload date to help users make informed decisions.

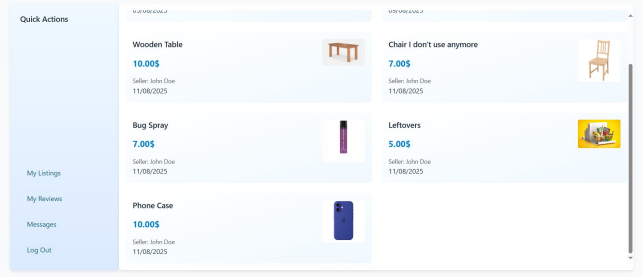

Figure 3: MainPage

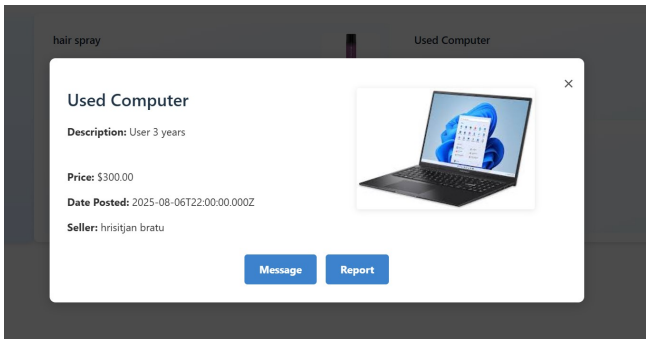

Figure 4: Enter Caption

2.4 Real-time Communication System

Description. This feature enables private messaging between buyers and sellers, allowing them to discuss item details, negotiate prices, or arrange transactions. While the interface provides near real-time updates, messages are stored in a central database and retrieved via API calls. The system is designed to provide a responsive and reliable chat experience without relying on constant page refreshes.

Most Difficult Part of Implementation. The messaging system uses a REST API for sending and retrieving messages, combined with periodic front-end polling to keep conversations updated. Implementing this required designing efficient database queries to fetch messages in the correct order, managing sender/receiver relationships, and ensuring proper authentication so users could only

access their own chats. Key challenges included minimizing polling intervals to improve responsiveness without overloading the server, preventing duplicate message loads, and ensuring that updates were reflected quickly in the user interface.

Instructions for Use.

- From any listing, click the "Message Seller" button to initiate a chat session.
- A chat interface will open, allowing you to exchange messages instantly with the seller.
- To close the chat, simply navigate away from the conversation or click the close icon within the chat window.

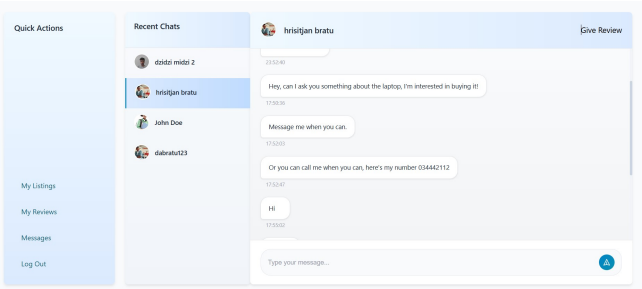

Figure 5: ChatSystem

2.5 User Feedback with Reviews

Description. The review system allows users to rate and provide feedback on each other following interactions, such as purchases or communications. This reputation system builds accountability and trust within the community by showcasing user reliability and service quality.

Most Difficult Part of Implementation. The main thing with implementing the functionality was maintaining a nice, delicate and a simple system for giving and getting reviews. We made this possible with a star system where the user can rate from 1-5 stars on how satisfied he was with the other users collaboration and a simple description box where we can type something if he'd like. Other than that simplicity was the key and was finished with only 2-3 routes + API points.

Instructions for Use.

- After engaging in a chat, the user can choose if he'd like to leave a review for the other party.
- Select an appropriate rating (e.g., 1 to 5 stars) and write a concise, honest comment.
- Submit the review to contribute to the user's overall reputation score.

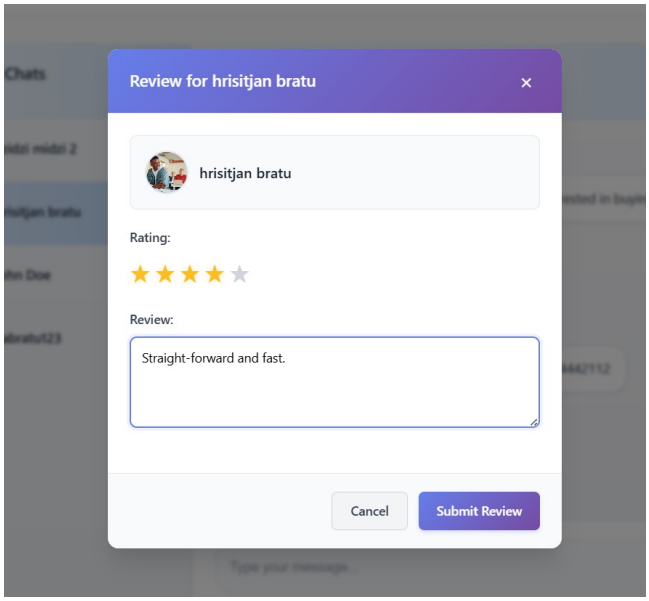

Figure 6: Reviews

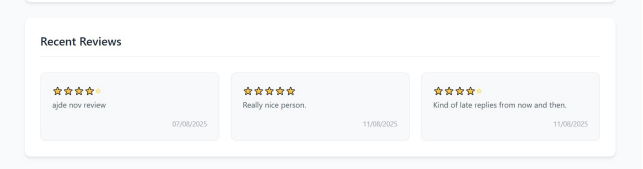

Figure 7: Personal Reviews

2.6 System for Reporting Items

Description. To maintain platform safety and integrity, users can report listings they believe to be inappropriate, fraudulent, or violating platform policies. This feature empowers the community to self-regulate and helps administrators identify problematic content quickly.

Most Difficult Part of Implementation. Creating an efficient backend workflow for report management was essential. This involved designing database models to store report details so the admin can see and view them. In future purpose this would work better in order for the Listings and Users be flagged for the reports they get and whether they are true or not.

Instructions for Use.

- On any listing page, click the clearly marked "Report" button.
- Choose the reason for your report from a predefined list (e.g., spam, inappropriate content, fraud).
- Submit the report. The platform will notify moderators for further review and potential action.

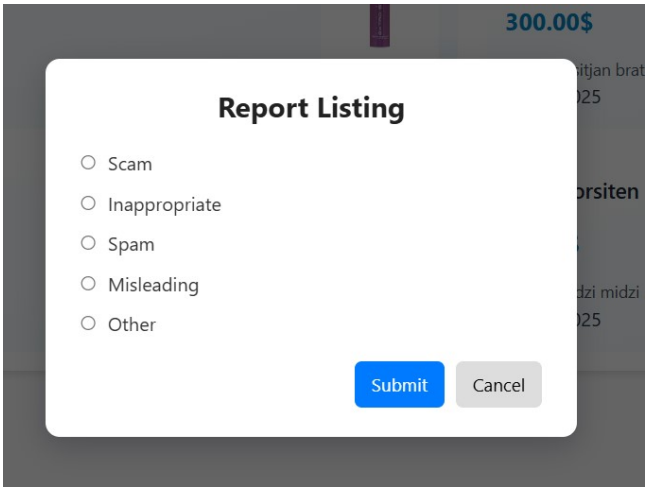

Figure 8: Reports

3 CONCLUSION

The **Student Trade** application successfully delivers a robust, user-centric online marketplace tailored for students to buy, sell, and trade items with ease and security. By integrating core functionalities such as secure user authentication, comprehensive profile management, and dynamic listings handling, the system ensures a seamless and trustworthy user experience.

The real-time communication feature significantly enhances the platform's interactivity, allowing users to negotiate and coordinate transactions directly, which fosters a more engaging and transparent marketplace environment. Furthermore, the incorporation of a user feedback and review system introduces accountability, encouraging positive interactions and building community trust.

The safety measures, including the item reporting system, demonstrate a commitment to maintaining platform integrity and user safety, allowing swift detection and management of inappropriate content or fraudulent activities.

From a technical perspective, the project showcases careful balancing of security, scalability, and usability. Challenges such as secure credential management, efficient state synchronization for listings, and real-time messaging were addressed using industry-standard technologies like JWT, Bcrypt, React state management, and WebSockets. These choices ensure that the platform is not only functional but also prepared to scale and adapt to future requirements.

In summary, **Student Trade** stands as a comprehensive solution that addresses the unique needs of a student marketplace, providing a secure, efficient, and interactive environment for peer-to-peer commerce. Future enhancements may include integrating advanced search capabilities, expanding the real-time features to group chats, and implementing AI-driven recommendations to further personalize the user experience.

This project lays a strong foundation for ongoing development and serves as a practical demonstration of applying modern web technologies to solve real-world problems in an academic context.