< Group 5 - Student TradeHub > Project Proposal

Team members

Full Name (Student id)	GitHub id
Olaiya Oluwatomisin (202580313)	tomisiiiin
Labib Islam (202580788)	labib-islam
Nafiur Rahman (202584000)	Nafiur-rhyme
Anya Anya (202580440)	Chuxs
Md Minhajul Abedin (202483033)	Minhajul99
Yi Zhang (202580740)	1mag1ne1
Yixuan Liu (202581744)	Yixuan-Liu1

Introduction

Students often face challenges when trying to buy, sell, or exchange items affordably and safely. Public platforms like Facebook Marketplace or Kijiji are open systems that may expose users to scams or interactions with strangers. To address this, we propose a closed marketplace platform designed specifically for the student community, which requires a valid student email address or ID for authentication.

This marketplace will encourage students to trade among themselves at affordable prices, foster community trust, and reduce scam risks by restricting participation to verified users within the institution.

Problem Statement

- No existing marketplace or trading platform for students at Memorial University (or similar institutions).
- Students risk scams when using open/public platforms.
- Lack of a trusted and affordable system for students to buy, sell, and trade items within their community

Proposed Solution

Core features include:

1. Authentication & Login – student-only access via MUN email/ID.
2. CRUD Operations – for items, services, and user profiles.
3. Search, Filter & Categorize Items – by price, category, or condition.
4. Transaction History, Purchase & Payment – record and track purchases securely.
5. Admin Functions – monitor listings and manage user access.
6. Order tracking system.

Extended features include:

7. Reviews & ratings for sellers and buyers.

Frameworks and tools include:

- Frontend: Next.js, Tailwind CSS
- Backend: Node.js
- Database: MongoDB (shared database access for the group)
- Version Control: GitHub

Requirements

Functional Requirements:

- Role-based access control (Admin, Student)
- User registration, login & profile management
- Password reset & account recovery.
- Product CRUD operation.
- Product catalog (browse, search, filter, categorise).
- Add to cart, buy product, sell product.
- Set payment method, payment details, address etc.
- Order tracking (real-time delivery status).
- Admin: Manage users, products, categories, and orders.

Non-functional Requirements:

- Error checking
- Authenticate using JWT
- Email verification to register
- Session management

Project Plan

Timelines for the project:

- Weeks 1–3: Planning and architecture
- Weeks 4–7: Core feature development (MVP)
- Weeks 8–10: UI/UX improvements, secondary features, error handling
- Weeks 11–12: Documentation, polish, usability testing
- Week 13: Final submission & presentation

Week	Dates	Sprint Goal / Process	Milestone
1	Sept 8 - 12	Brainstorm project ideas, evaluate feasibility, pick final ideas.	
2	Sept 15 – 19	Define requirements (user stories, acceptance criteria). Start backlog creation.	
3	Sept 22 – 26	Refine backlog, finalize scope, outline architecture. Plan design (flow chart).	📌 Project Proposal due Sept. 28
4	Sept 29 – Oct 3	Sprint 1: Set up dev environment (GitHub), version control, and skeleton code.	
5	Oct 6 – 10	Sprint 2: Implement core features (MVP version). Start coding key functions.	

6	Oct 13 – 17	Sprint 3: Continue core feature dev. Write unit tests. Add more core features.	
7	Oct 20 – 24	Sprint 4: Integrate features, ensure basic workflow works.	📌 Progress Report due Oct. 26
8	Oct 27 – 31	Sprint 5: Add secondary features, improve UI/UX. Improve stability.	
9	Nov 3 – 7	Sprint 6: Optimize functionality, refine based on feedback. Add error handling.	
10	Nov 10 – 14	Sprint 7: Perform integration testing & bug fixing.	
11	Nov 17 – 21	Sprint 8: Documentation (user manual, technical docs).	
12	Nov 24 – 28	Sprint 9: Final polish, usability testing, prepare final presentation.	
13	Dec 1 – 5	Sprint 10: Submit & present Final Project.	📌 Final project due end of the week

Team member responsibilities:
Each team member works on each of the features of the project.

Expected Deliverables

- A functional student-only trading platform with core features implemented.
- Secure authentication using MUN ID and Email.
- Transaction system with purchase history.
- Working prototype ready for demonstration and further enhancement.
- Documentation and user manual.

References

- GitHub Authentication Documentation. Available at: https://docs.github.com/en/authentication
- MongoDB Documentation, 2024. Available at: https://www.mongodb.com/docs/drivers/node/current/