

Campus Marketplace

Project Journal

CMPE 202 - Software Systems Engineering
Team: Visionary Coders

September - December 2025

Team Members

- **Manasa Sadhu** - Buyer & Seller UI, Real-time Chat, Database Design
- **Krishna Panjiyar** - NLP Search Engine, GPT Chatbot Integration
- **Tejkiran Yenugunti** - Admin Dashboard, Moderation & User Management
- **Girith Choudhary** - Cloud Deployment, CI/CD, Infrastructure

Week 1: September 8-14, 2025

Sprint Goals

Initial project setup, technology stack selection, and team coordination. Focus on establishing development environment and project architecture.

Tasks Completed

- Created GitHub repository and established branching strategy (main, dev, feature branches)
- Set up Next.js 15.5.4 frontend with React 19 and Tailwind CSS
- Configured FastAPI backend with Python and SQLAlchemy ORM
- Designed initial database schema for users, listings, and categories
- Set up PostgreSQL database locally and created migration scripts

Technical Decisions

- **Frontend:** Chose Next.js for server-side rendering and SEO benefits
- **Backend:** Selected FastAPI for high performance and automatic API documentation
- **Database:** PostgreSQL chosen for relational data and ACID compliance
- **Authentication:** JWT-based auth with .edu email verification

Challenges & Solutions

- **Challenge:** Docker setup complexity with psycopg2 dependencies
- **Solution:** Configured Dockerfile with python:3.11 base image and installed build dependencies (gcc, python3-dev, libpq-dev)
- **Challenge:** Team coordination across different time zones
- **Solution:** Established daily standup meetings at 7 PM PST and Slack for async communication

Story Points: 28 Planned, 26 Completed (93%)

Week 2: September 15-21, 2025

Sprint Goals

Implement core authentication system and initial UI components for buyer and seller flows.

Tasks Completed

- Built sign-in and sign-up pages with form validation
- Implemented JWT authentication with access/refresh tokens
- Created user registration API with .edu email verification
- Developed role-based access control (Buyer, Seller, Admin)
- Built initial homepage layout with navigation components
- Created reusable UI components (buttons, forms, cards) in Tailwind

Technical Decisions

- Used bcrypt for password hashing with salt rounds of 12
- Implemented refresh token rotation for enhanced security
- Stored tokens in httpOnly cookies to prevent XSS attacks
- Created middleware for protected routes in Next.js

Challenges & Solutions

- **Challenge:** CORS issues between frontend and backend
- **Solution:** Configured FastAPI CORS middleware with appropriate origins and credentials
- **Challenge:** Form validation inconsistencies
- **Solution:** Integrated Zod for TypeScript schema validation on frontend

Story Points: 28 Planned, 28 Completed (100%)

Week 3: September 22-28, 2025

Sprint Goals

Develop product listing functionality with CRUD operations and image upload capabilities.

Tasks Completed

- Built seller dashboard to view and manage listings
- Created product listing form with title, description, price, category, and location fields
- Implemented image upload with preview and multi-image support
- Developed backend API endpoints for listing CRUD operations
- Added category dropdown (Textbooks, Electronics, Furniture, Clothing, Sports, Essentials)
- Implemented 'Mark as Sold' functionality for listings

Technical Decisions

- Used AWS S3 for image storage with presigned URLs
- Implemented image compression on frontend before upload (max 2MB per image)
- Set up database indexes on category and created_at columns for faster queries

- Added soft delete pattern for listings (archived rather than deleted)

Challenges & Solutions

- **Challenge:** Large image files causing slow uploads
- **Solution:** Implemented client-side compression using browser-image-compression library
- **Challenge:** Managing multiple image uploads in form state
- **Solution:** Used React state with array of File objects and preview URLs

Story Points: 32 Planned, 30 Completed (94%)

Week 4: September 29 - October 5, 2025

Sprint Goals

Build buyer browsing experience with product detail pages and search/filter functionality.

Tasks Completed

- Created product listings grid view for buyers
- Developed detailed product page with image gallery and seller information
- Implemented category filter with checkbox selection
- Added price range filter (min/max) with slider UI
- Built basic keyword search functionality
- Added pagination for listings (20 items per page)

Technical Decisions

- Used URL query parameters for filters to enable shareable filtered views
- Implemented debouncing for search input to reduce API calls
- Created database view for optimized listing queries with seller info
- Used React Context API for managing global filter state

Challenges & Solutions

- **Challenge:** Slow query performance with multiple filters
- **Solution:** Added composite indexes on (category, price, created_at) columns
- **Challenge:** Image gallery not responsive on mobile
- **Solution:** Implemented Swiper.js for mobile-friendly image carousel

Story Points: 32 Planned, 32 Completed (100%)

Week 5: October 6-12, 2025

Sprint Goals

Implement NLP-powered search using GPT API and begin real-time chat system development.

Tasks Completed

- Integrated OpenAI GPT-4 API for natural language search
- Built query preprocessing to extract intent and entities
- Created search endpoint that processes queries like 'textbook for CMPE202'

- Designed chat database schema (conversations, messages tables)
- Set up WebSocket connection for real-time messaging
- Implemented chat UI with message threads per listing

Technical Decisions

- Used GPT-4 with custom system prompts for search query understanding
- Implemented caching layer for common search queries (Redis)
- Selected Socket.IO for WebSocket implementation with fallback to polling
- Stored message history in PostgreSQL with indexed conversation_id

Challenges & Solutions

- **Challenge:** GPT API latency affecting search speed
- **Solution:** Added loading states and implemented query result caching
- **Challenge:** WebSocket connection drops on mobile
- **Solution:** Implemented automatic reconnection with exponential backoff

Story Points: 30 Planned, 28 Completed (93%)

Week 6: October 13-19, 2025

Sprint Goals

Complete real-time chat system and begin admin dashboard development.

Tasks Completed

- Finalized buyer-seller chat with message delivery confirmation
- Added chat notifications for new messages
- Built chat history view showing all conversations
- Created admin dashboard layout with navigation sidebar
- Implemented listing approval workflow (pending → approved → active)
- Built admin panel to view all pending listings

Technical Decisions

- Implemented read/unread status for messages using timestamp comparison
- Used browser notifications API for desktop notifications
- Created admin middleware to enforce role-based access
- Added listing moderation queue with status transitions

Challenges & Solutions

- **Challenge:** Message order inconsistencies with concurrent users
- **Solution:** Used server timestamps and optimistic UI updates with rollback
- **Challenge:** Admin dashboard permissions not properly enforced
- **Solution:** Added server-side role verification on all admin endpoints

Story Points: 30 Planned, 29 Completed (97%)

Week 7: October 20-26, 2025

Sprint Goals

Complete admin features and begin Azure cloud deployment preparation.

Tasks Completed

- Built user management interface for admins
- Implemented report handling system for inappropriate listings
- Added platform statistics dashboard (active users, listings, transactions)
- Created Azure account and set up resource groups
- Configured Azure PostgreSQL database instance
- Set up GitHub Actions workflow for CI/CD

Technical Decisions

- Used Azure Database for PostgreSQL Flexible Server for better scalability
- Configured automated database backups with 7-day retention
- Set up separate staging and production environments
- Implemented analytics tracking using database triggers

Challenges & Solutions

- **Challenge:** Azure PostgreSQL firewall blocking local connections
- **Solution:** Configured firewall rules and used Azure VNet for secure access
- **Challenge:** GitHub Actions failing on database migrations
- **Solution:** Added migration step in deployment pipeline with rollback support

Story Points: 28 Planned, 27 Completed (96%)

Week 8: October 27 - November 2, 2025

Sprint Goals

Deploy backend to Azure and frontend to Vercel with full integration testing.

Tasks Completed

- Deployed FastAPI backend to Azure App Service
- Configured environment variables and secrets in Azure
- Deployed Next.js frontend to Vercel with production build
- Set up custom domain and SSL certificates
- Configured CORS for production domain
- Implemented health check endpoints for monitoring

Technical Decisions

- Used Azure App Service with Docker container deployment
- Configured auto-scaling rules based on CPU and memory usage
- Set up Azure Application Insights for performance monitoring
- Implemented CDN caching for static assets

Challenges & Solutions

- **Challenge:** Frontend-backend integration failing on production
- **Solution:** Debugged CORS headers and updated API base URL configuration
- **Challenge:** WebSocket connections not working through Azure proxy
- **Solution:** Configured WebSocket support in Azure App Service settings

Story Points: 30 Planned, 30 Completed (100%)

Week 9: November 3-9, 2025

Sprint Goals

UI refinement, bug fixes, and performance optimization across the platform.

Tasks Completed

- Refined UI/UX based on user testing feedback
- Fixed mobile responsiveness issues on product detail pages
- Optimized image loading with lazy loading and WebP format
- Improved search performance with query optimization
- Added loading states and error handling throughout app
- Implemented form field validation with user-friendly error messages

Technical Decisions

- Used Next.js Image component for automatic optimization
- Implemented skeleton loaders for better perceived performance
- Added database query explain analysis to identify slow queries
- Configured connection pooling for database connections

Challenges & Solutions

- **Challenge:** Slow page load times on listing pages
- **Solution:** Implemented server-side pagination and reduced initial data payload
- **Challenge:** Chat messages not updating in real-time for all users
- **Solution:** Fixed WebSocket room management and message broadcasting logic

Story Points: 26 Planned, 26 Completed (100%)

Week 10: November 10 - December 3, 2025

Sprint Goals

Final testing, documentation, and demo preparation for project presentation.

Tasks Completed

- Conducted end-to-end testing of all user flows
- Fixed remaining bugs and edge cases
- Created comprehensive README with setup instructions
- Generated API documentation using FastAPI's automatic docs
- Prepared demo scenarios for buyer, seller, and admin roles
- Created presentation slides and architecture diagrams
- Seeded database with realistic mock data for demo

Technical Decisions

- Used Faker library for generating realistic test data
- Created separate demo environment with pre-populated data
- Set up monitoring dashboard to display during demo
- Documented deployment architecture and infrastructure details

Final Reflections

- **Successes:** Delivered all core features, maintained consistent velocity, excellent team collaboration

- **Challenges:** Cloud deployment complexity, WebSocket reliability, GPT API cost management
- **Lessons Learned:** Importance of early CI/CD setup, value of daily standups, benefits of pair programming for complex features
- **Future Improvements:** Mobile app development, payment integration, machine learning recommendations, enhanced analytics

Story Points: 30 Planned, 30 Completed (100%)

Project Summary

Overall Statistics

- **Total Story Points:** 120
- **Completed Story Points:** 117
- **Average Velocity:** 97.5%
- **Duration:** 10 weeks (September 8 - December 3, 2025)
- **Team Size:** 4 members

Key Achievements

- Successfully deployed full-stack marketplace platform to production
- Implemented AI-powered natural language search using GPT-4
- Built real-time chat system with WebSocket support
- Created comprehensive admin moderation system
- Deployed on Azure with auto-scaling infrastructure
- Achieved 97.5% story point completion rate

Technology Stack

- **Frontend:** Next.js 15.5.4, React 19, Tailwind CSS, TypeScript
- **Backend:** FastAPI, Python, SQLAlchemy ORM
- **Database:** PostgreSQL on Azure
- **Authentication:** JWT with .edu email verification
- **AI/NLP:** OpenAI GPT-4 API
- **Real-time:** Socket.IO for WebSocket connections
- **Cloud:** Azure App Service, Azure PostgreSQL, Vercel (frontend)
- **CI/CD:** GitHub Actions