DIU CampusCart

Submitted By

Student Name	Student ID
Abdullah Al Noman	232-15-797
Supan Roy	232-15-716
Md. Shahinur Kabir Antor	232-15-159
Nur Sayda	232-15-437
Shakira Rahman Simi	232-15-723

MINI LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE222: Object Oriented Programming Lab in the Computer Science and Engineering Department**

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

April 15, 2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Md. Sazzadur Ahamed**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

____

Md. Sazzadur Ahamed
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by

Abdullah Al Noman Student ID: 232-15-797 Dept. of CSE, DIU	
Supan Roy Student ID: 232-15-716 Dept. of CSE, DIU	Md. Shahinur Kabir Antor Student ID: 232-15-159 Dept. of CSE, DIU
Nur Sayda Student ID: 232-15-437 Dept. of CSE, DIU	Shakira Rahman Simi Student ID: 232-15-723 Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course has course outcomes as following:

Table 1: Course Outcome Statements

CO's	Statements
CO1	**Define** and **Relate** classes, objects, members of the class, and relationships among them needed for solving specific problems
CO2	**Formulate** knowledge of object-oriented programming and Java in problem solving
CO3	**Analyze** Unified Modeling Language (UML) models to **Present** a specific problem
CO4	**Develop** solutions for real-world complex problems **applying** OOP concepts while evaluating their effectiveness based on industry standards.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1, EP3
CO2	PO2	C2	KP3	EP1, EP3
CO3	PO3	C4, A1	KP3	EP1, EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1, EP3

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

Table of Contents

Chapter 1

Introduction

This chapter outlines the background, motivation, objectives, feasibility, gap analysis, and expected outcomes of the CampusCart project. It sets the context for building a classified ads platform exclusively for DIU students.

1.1 Introduction

In modern university campuses, students often need a reliable and secure platform to buy and sell items like books, gadgets, and accessories. Existing platforms like Facebook/Telegram groups or Bikroy.com lack institutional restrictions, making them unsuitable for DIU-specific interactions.

CampusCart addresses this problem by creating a web-based classified ads platform tailored only for DIU students. With DIU email verification, secure authentication, and user-friendly design, it ensures that only verified users can access and interact on the platform, enhancing trust and relevance.

1.2 Motivation

As students ourselves, we have often faced difficulties in finding a safe and trustworthy medium for selling or buying goods on campus. Most existing platforms are either too broad, lack verification, or are filled with spam, leading to a frustrating experience. Our motivation stems from the desire to create a focused, secure, and exclusive system where DIU students can trade easily. Solving this problem not only benefits the DIU community but also enhances our understanding of real-world web development practices, Spring Boot backend, and secure user authentication.

1.3 Objectives

The core aim of the CampusCart project is to design a platform that solves a specific problem within the DIU student community—facilitating a secure, student-only system for exchanging goods. While various online platforms exist for product listings, none cater to the DIU ecosystem with built-in user verification and academic exclusivity. Through this project, our objective is not only to address that gap but also to enhance our technical skills in full-stack web development using modern tools and frameworks.

The specific objectives of this project are as follows:

- To develop a secure and user-friendly platform dedicated to DIU students for listing and exchanging products.
- To implement DIU email-based verification to ensure user authenticity and access control.
- To provide features for uploading products with images, titles, descriptions, and categories.
- To build the system using Java Spring Boot for the backend and HTML with Tailwind CSS for the frontend.

- To apply Spring Security for login, registration, and session handling.
- To structure the project with clean architecture, separating controller, service, and repository layers.
- To use H2 as the development database for storing user and product data.
- To deploy a fully functional prototype that demonstrates both technical capability and user-centered design.

1.4 Feasibility Study

Many classified platforms like OLX, Bikroy, and Facebook Marketplace exist, offering similar services on a larger scale. However, these platforms are not restricted to specific institutions and lack the academic context that CampusCart aims to fulfill.

Some universities in other countries have developed internal exchange platforms, but none currently exist in DIU. Methodologically, our work aligns with microservice-based Spring Boot applications and follows secure web application development practices using Spring Security. By using technologies like Tailwind CSS and H2 DB, we can build a lightweight yet fully functional prototype suitable for university use.

1.5 Gap Analysis

While platforms for local classifieds exist, they are either too generalized or do not guarantee user authenticity. DIU students lack a platform specifically designed for them where trust and relevancy are maintained through email verification and internal access control.

CampusCart fills this gap by offering DIU-only access, secure authentication, and simple UI that is both responsive and user-centric.

1.6 Project Outcome

Upon completion, CampusCart provides:

- A functioning classified platform tailored for DIU students.
- A complete full-stack web application demonstrating backend and frontend integration.
- Secure login and registration with Spring Security.
- A clean UI using Tailwind CSS.
- Learnings on handling authentication, file/image upload, form validations, and controller-service-repo architecture.

Future outcomes may include implementing search filters, messaging between users, integrating Firebase or MySQL for production deployment, and possibly building a mobile version of the platform.

Chapter 2

Proposed Methodology/Architecture

This chapter outlines the system design, methodology, and overall project planning followed during the development of the CampusCart platform. It presents how the requirements were translated into a secure, scalable architecture tailored for DIU students.

2.1 Requirement Analysis & Design Specification

2.1.1 Overview

Before initiating the development of CampusCart, a preliminary requirement analysis was conducted to understand the specific needs of DIU students regarding on-campus product exchange. Informal surveys and discussions with peers revealed that students often face challenges when trying to buy or sell personal items such as books, electronics, and accessories due to the lack of a dedicated and secure platform. Most available platforms are either too broad or lack proper user verification, making them unsuitable for DIU-specific use. These insights led us to define a set of essential features including DIU-only access via email verification, product posting with image support, category-based browsing, and a clean, responsive user interface. Based on these requirements, we proceeded with designing a secure and structured system that meets the unique expectations of the DIU student community.

2.1.2 Proposed Methodology/ System Design

CampusCart follows the MVC (Model-View-Controller) design pattern using Java Spring Boot for backend and HTML with Tailwind CSS for frontend. The backend handles user management, product listing, authentication, and routing, while the frontend presents a simple, intuitive interface for interaction.

System Workflow:

1. Registration & Login: Students register using their DIU email. The system verifies email format (**@diu.edu.bd**) and allows access only if the condition is met.
2. Authentication: Implemented using Spring Security with session management and CSRF protection.
3. Product Upload: Users can upload items along with images, name, price, and category. The image is stored on the server and mapped with the database entry.
4. Product Browsing: All posted products are displayed and categorized. Each listing includes a "Mark as Sold" option.
5. Backend Logic: The service layer handles logic such as file validation, email checks, and repository interactions.
6. Database Layer: An H2 in-memory database is used during development for storing user and product information.

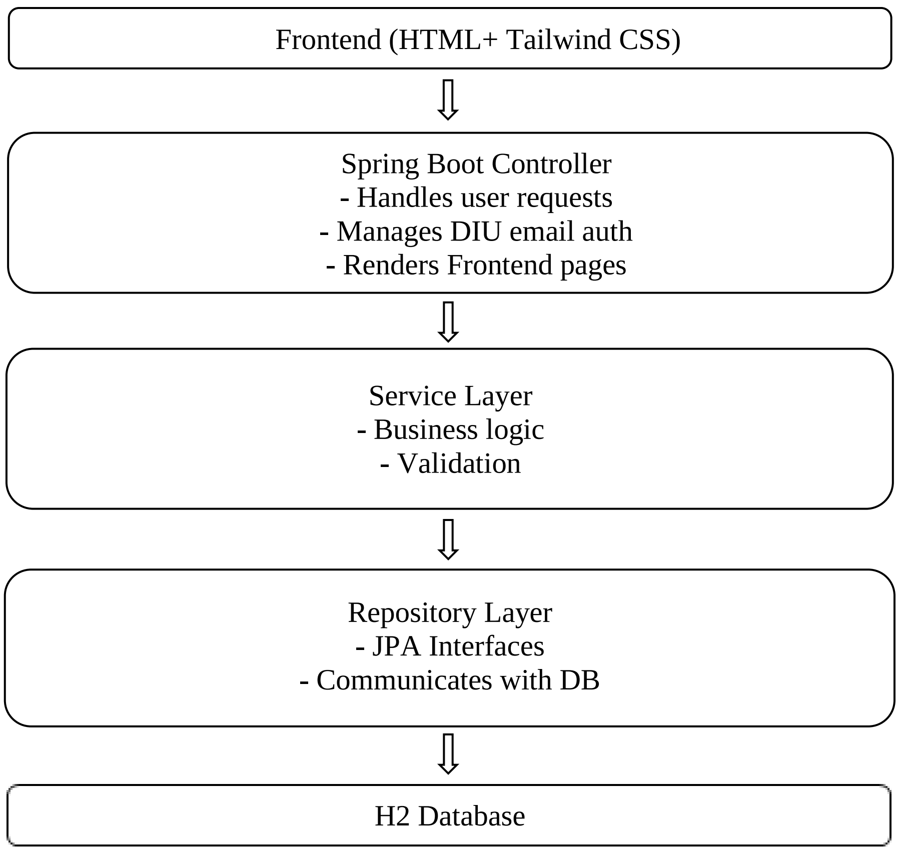


Figure 2.1: *System Architecture of CampusCart Platform using Spring Boot.*

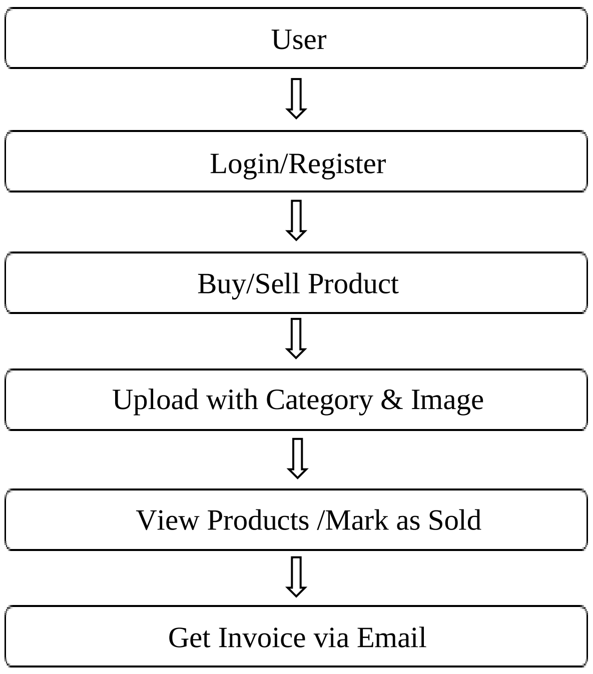


Figure 2.2: *User Interaction Flow.*

2.1.3 UI Design

To ensure a smooth and intuitive user experience, the platform has been designed with a clean and user-friendly interface. The frontend is built using HTML and Tailwind CSS, ensuring responsiveness and aesthetic consistency across devices. Below are the screenshots showcasing the key pages and features of the system, including product listing, user login, product upload, and the newly added invoice notification interface.

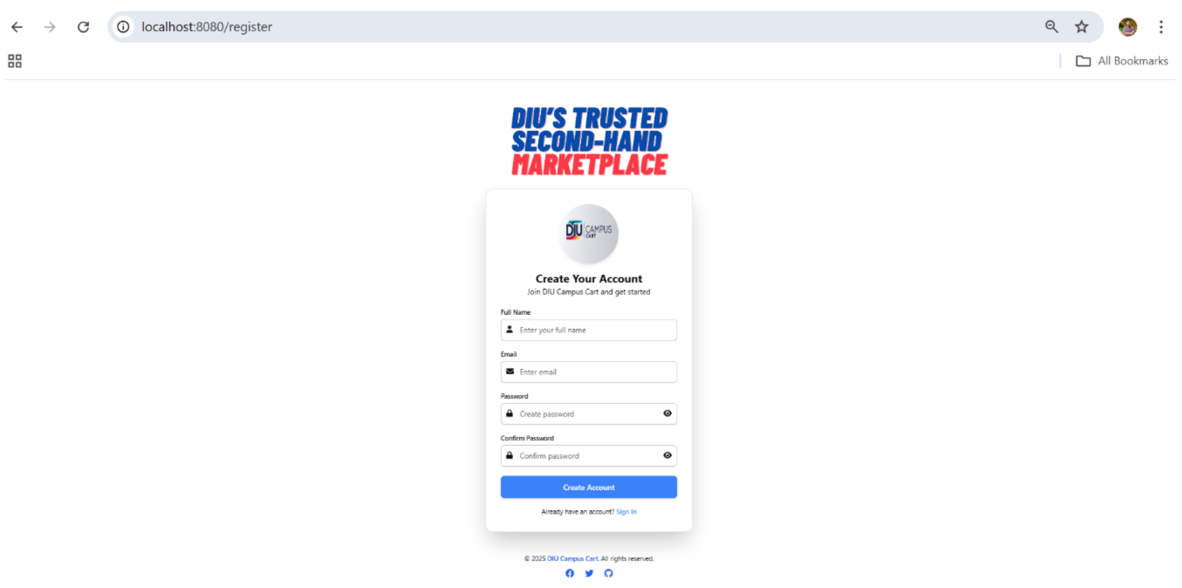


Figure 2.3: *Create Account/ Sign Up*

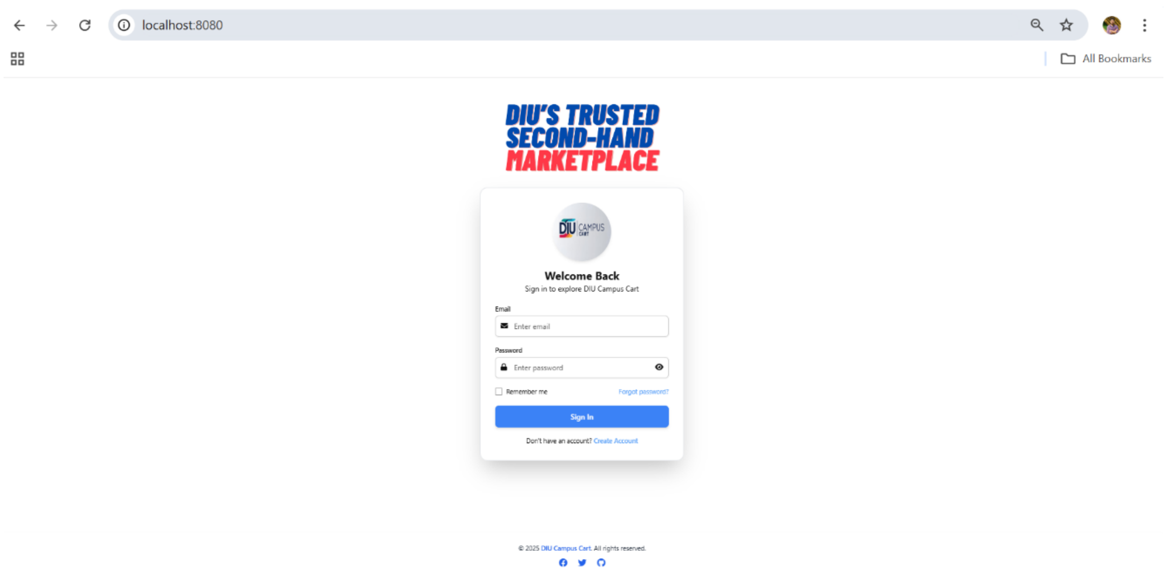

localhost:8080
All Bookmarks
DIU'S TRUSTED SECOND-HAND MARKETPLACE
Welcome Back
Sign in to explore DIU Campus Cart
Email
Enter email
Password
Enter password
Remember me
Forgot password?
Sign In
Don't have an account? Create Account
© 2025 DIU Campus Cart. All rights reserved.

Figure 2.4: *Sign In*

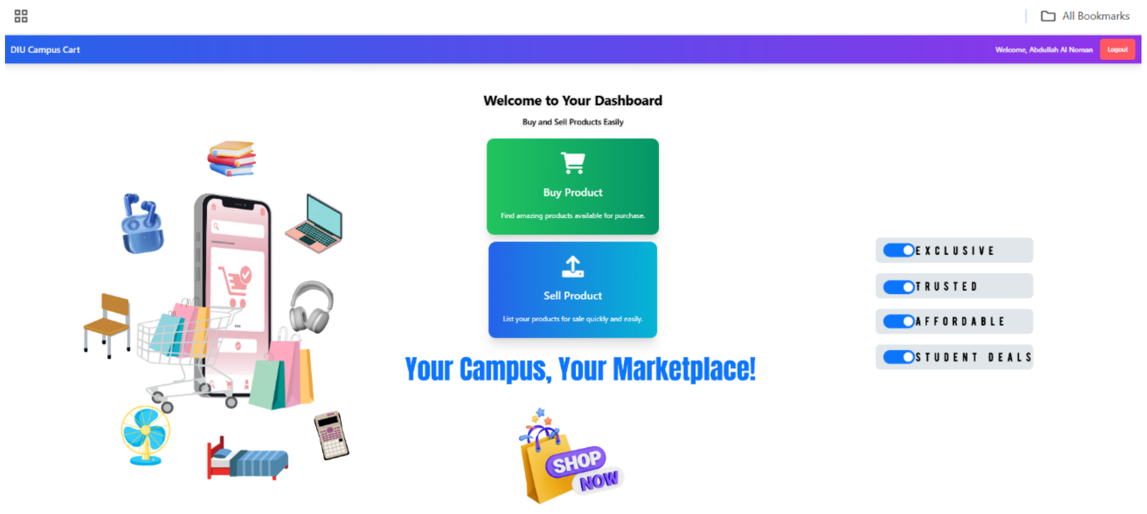

All Bookmarks
DIU Campus Cart
Welcome, Abdullah Al Noman
Logout
Welcome to Your Dashboard
Buy and Sell Products Easily
Buy Product
Find amazing products available for purchase.
Sell Product
List your products for sale quickly and easily.
Your Campus, Your Marketplace!
SHOP NOW
EXCLUSIVE
TRUSTED
AFFORDABLE
STUDENT DEALS

Figure 2.5: *User Dashboard*

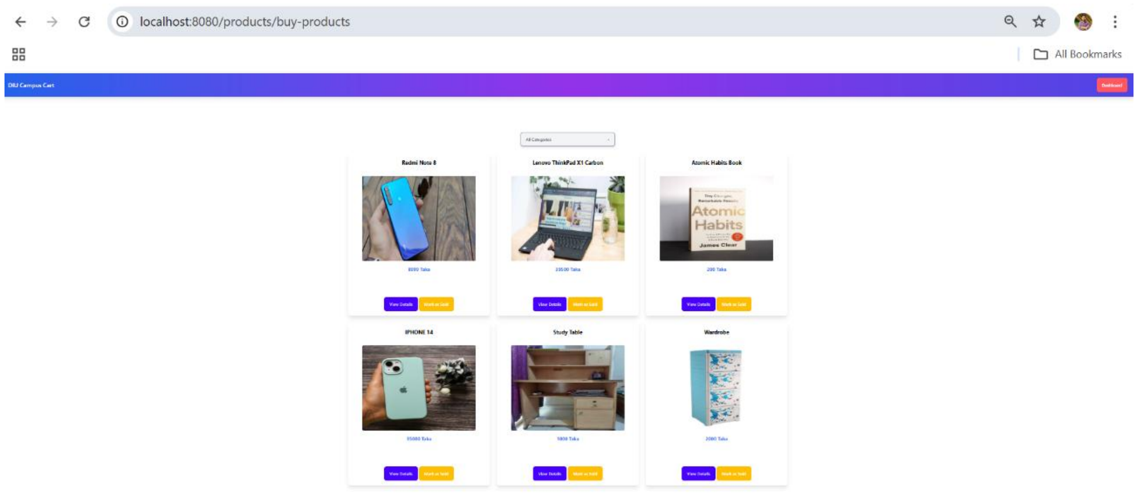
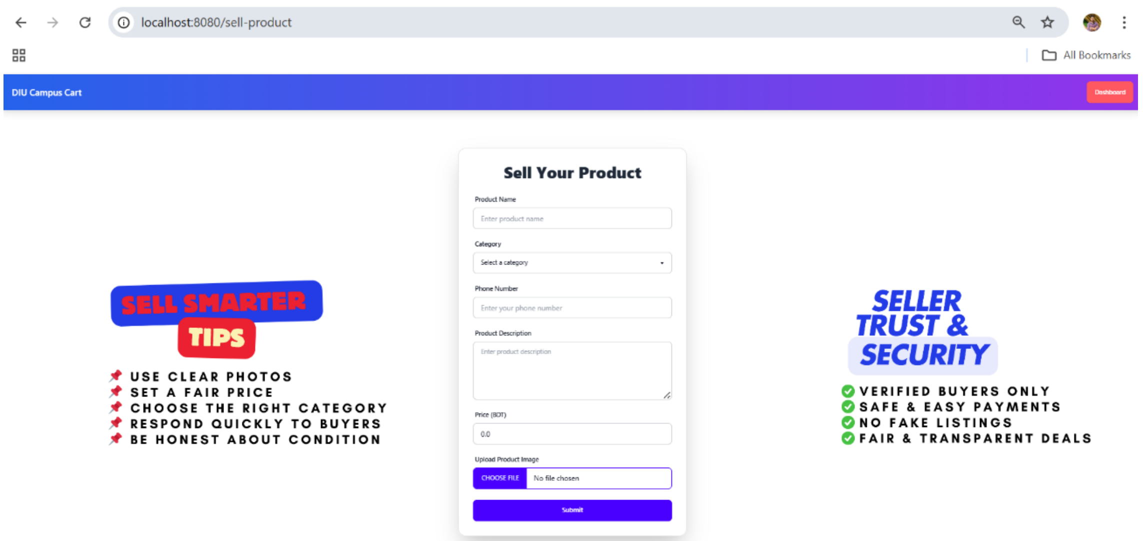


Figure 2.6: *Buy Products*



Sell Your Product

Product Name: Enter product name

Category: Select a category

Phone Number: Enter your phone number

Product Description: Enter product description

Price (BDT): 0.0

Upload Product Image: CHOOSE FILE No file chosen

Submit

Figure 2.6: *Sell Products*

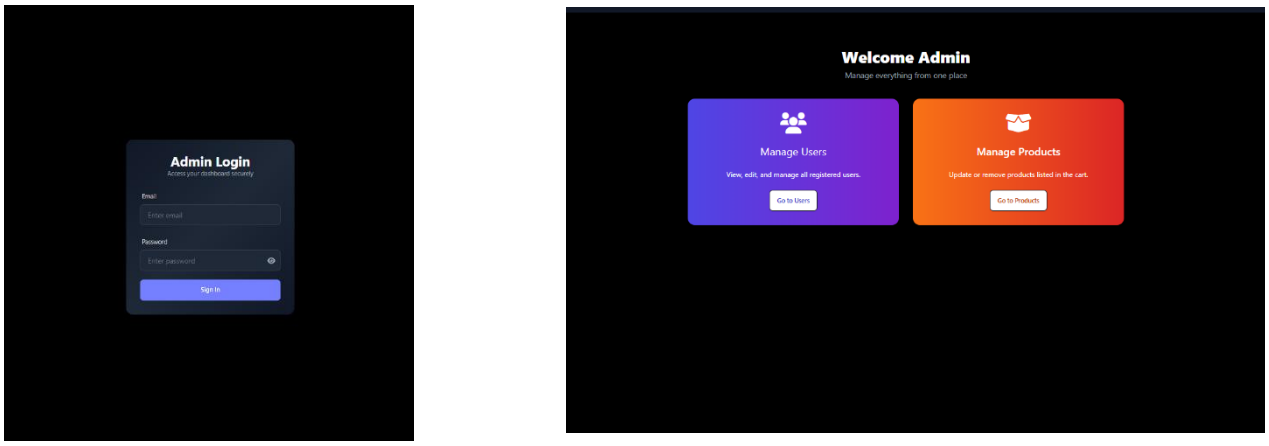
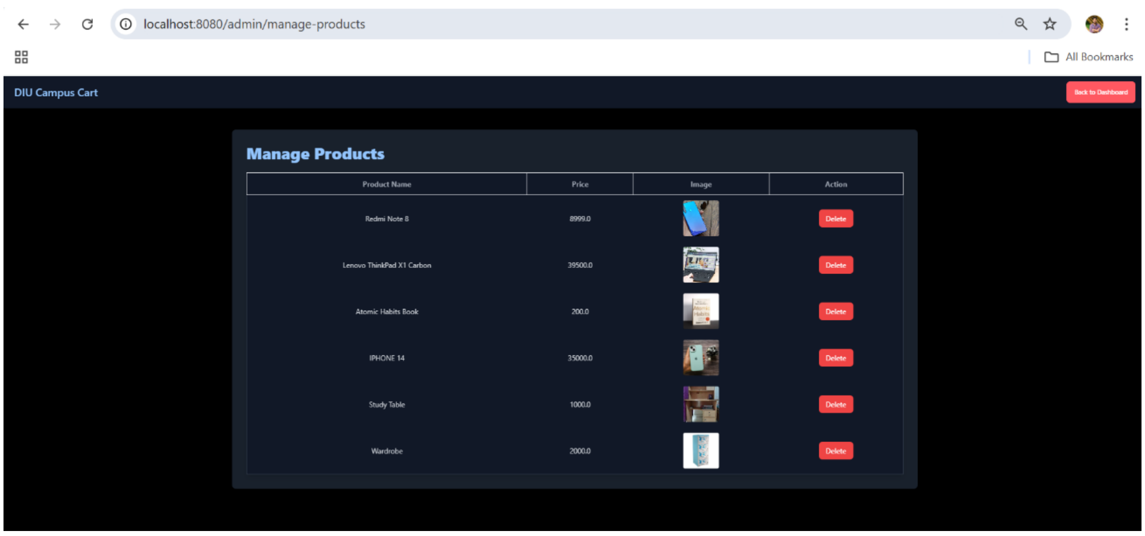

Admin Login
Access your dashboard securely
Email
Enter email
Password
Enter password
Sign In
Welcome Admin
Manage everything from one place
Manage Users
View, edit, and manage all registered users.
Go to Users
Manage Products
Update or remove products listed in the cart.
Go to Products

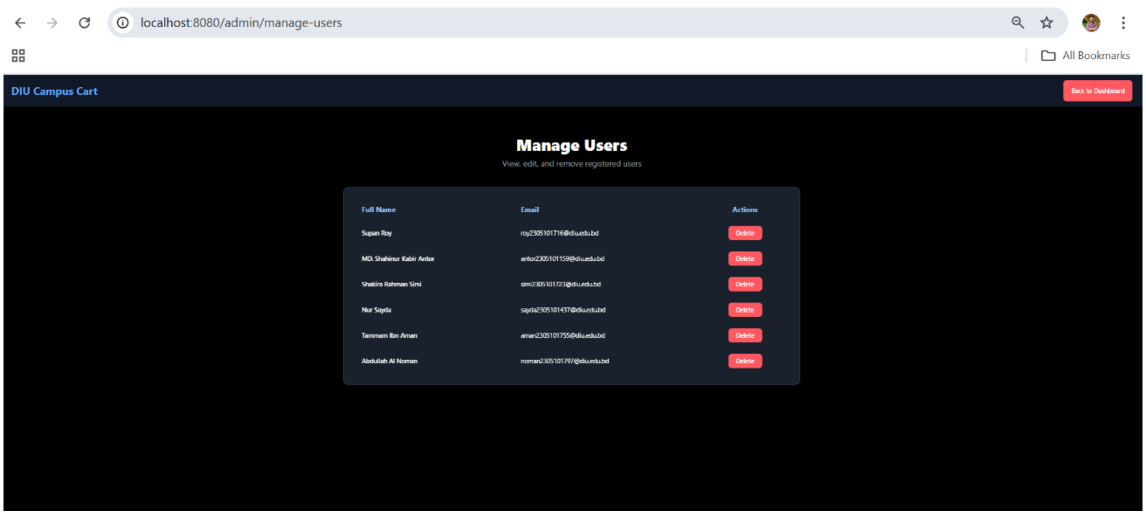
Figure 2.7: *Admin Login and Dashboard*



Manage Products

Product Name	Price	Image	Action
Redmi Note 8	8999.0		Delete
Lenovo ThinkPad X1 Carbon	39500.0		Delete
Atomic Habits Book	200.0		Delete
IPHONE 14	35000.0		Delete
Study Table	1000.0		Delete
Wardrobe	2000.0		Delete

Figure 2.8: *Manage Products*



Manage Users

View, edit, and remove registered users

Full Name	Email	Actions
Supan Roy	roy2305101716@diu.edu.bd	Delete
MD. Shahinur Kabir Antor	antor2305101159@diu.edu.bd	Delete
Shakira Rahman Simi	simi2305101723@diu.edu.bd	Delete
Nur Sayda	sayda2305101437@diu.edu.bd	Delete
Tammam Ibn Aman	aman2305101755@diu.edu.bd	Delete
Abdullah Al Noman	noman2305101797@diu.edu.bd	Delete

Figure 2.9: *Manage Users*

2.2 Overall Project Plan

To ensure smooth development and timely completion, the CampusCart project was divided into multiple phases. Each phase focused on specific technical goals, using appropriate tools and technologies to achieve them efficiently.

Phase	**Task**	**Tools/Tech Used**	**Duration**
Phase 1	Requirement Analysis, Tool Selection	Google Docs, GitHub	1 Week
Phase 2	Backend Setup (User Model, Auth)	Java, Spring Boot, H2 DB	2 Weeks
Phase 3	Frontend Development (UI Screens)	HTML, Tailwind CSS	1 Week
Phase 4	Image Upload & Product Logic	Spring Boot, Multipart Config	1 Week
Phase 5	Security & Email Restriction	Spring Security	1 Week
Phase 6	Testing, Final Fixes, Documentation	Browser, Git	1 Week

Chapter 3

Implementation and Results

This chapter presents the actual implementation of the **CampusCart** system along with an analysis of its performance and results. It explains the major technical components implemented, their integration, and evaluates how the system meets its intended objectives.

3.1 Implementation

The implementation of CampusCart followed a modular, layer-based approach using Java Spring Boot and MVC architecture. The application was divided into packages such as controller, service, repository, model, and config, each responsible for handling different parts of the system.

- **User Authentication**: Registration and login were developed using Spring Security. DIU email restriction was enforced using regular expression validation on the registration form to ensure only DIU **(@diu.edu.bd)** emails were allowed.
- **Product Upload**: A multipart form was used to upload product images and details. Image files were saved in the server directory, and corresponding metadata (title, price, category, status, etc.) was saved in the H2 database.
- **Secure Access**: Authorization checks ensured that only logged-in users could access the upload or dashboard features. Cross-Site Request Forgery (CSRF) protection and session timeout were configured as part of Spring Security.
- **Category-wise Browsing & Marking Sold Items**: Products could be browsed by predefined categories. A "**Mark as Sold**" feature was implemented that updated the product status in the database and displayed it visually on the product card.
- **Frontend Integration**: All pages were created using HTML with Tailwind CSS for a responsive and clean interface. Data was dynamically loaded using Thymeleaf templating.

3.2 Performance Analysis

As the system was built using Spring Boot and tested locally with the H2 database, performance remained efficient and lightweight. Load testing with multiple product uploads and logins revealed:

- Fast Load Time: Product pages loaded within 1-2 seconds even with 30+ entries.

- Minimal Memory Usage: H2's in-memory database ensured low RAM usage during development and testing.

- Secure Authentication: No unauthorized access was possible thanks to Spring Security's protection against CSRF and session hijacking.

- Scalability Potential: The modular structure and RESTful approach allow easy extension to production-grade databases like MySQL or Firebase in the future.

3.3 Results and Discussion

The completed system successfully met all the core requirements outlined in the project objectives. DIU email verification worked accurately, preventing access from external users. Product listings were successfully posted with image previews and sorted by categories. Sold items were clearly marked, improving browsing clarity.
User testing within a closed group of DIU students showed positive responses regarding usability and design simplicity. The clean UI and smooth interaction, even with minimal training, highlighted the effectiveness of our chosen tools and methodologies. Overall, the implementation results were satisfactory and laid a strong foundation for future scalability and feature enhancements.

Chapter 4

Engineering Standards and Mapping

This chapter highlights how the CampusCart project aligns with engineering standards, ethical values, and sustainable practices. It also discusses the project's impact on society and environment, and provides a breakdown of project management, team collaboration, and cost analysis.

4.1 Impact on Society, Environment and Sustainability

4.1.1 Impact on Life

CampusCart provides a convenient platform for DIU students to exchange personal items securely within their own community. This reduces the dependency on external marketplaces and enables students to buy essentials like books, electronics, or accessories at affordable prices. It contributes positively to student life by promoting reuse, saving money, and supporting a helpful community environment.

4.1.2 Impact on Society & Environment

By encouraging the resale and reuse of items, CampusCart helps reduce waste generation and promotes a more sustainable lifestyle. Instead of disposing of unused products, students can now sell them to peers, reducing environmental impact. The platform also builds a sense of trust and safety in the campus environment through DIU-only access, minimizing risks associated with public classified platforms.

4.1.3 Ethical Aspects

From the start, the project emphasized ethical practices, particularly in terms of data privacy, security, and fair usage. Spring Security was integrated to protect user information, and no sensitive personal data (like phone numbers or addresses) is displayed publicly. The platform avoids any form of discrimination or bias, maintaining fairness in product visibility and user access.

4.1.4 Sustainability Plan

CampusCart is designed with scalability and long-term sustainability in mind. The project is structured using clean code practices and modular architecture, which allows future teams or contributors to maintain or extend its functionality. As demand grows, the system can be migrated from H2 to cloud-based databases like Firebase or MySQL. A potential mobile version and additional features like chat, search filters, and delivery coordination are also part of the sustainability roadmap.

4.2 Project Management and Team Work

The development of CampusCart followed a collaborative, phase-based plan. Tasks were divided among team members based on their strengths—some focused on backend logic, others on frontend design or security configuration. Version control was maintained using GitHub, ensuring code consistency and allowing smooth collaboration.

Budget Analysis

Since this was an academic project, the team utilized free and open-source tools to minimize cost. Here's a breakdown of both the **original budget** (based on potential real-world deployment) and the

alternate budget (based on what was actually used):

Item	**Original Budget (BDT)**	**Alternate Budget (BDT)**	**Rationale**
Domain & Hosting	2,500	0	Localhost used during development
Developer Tools/IDE	1,200	0	IntelliJ Community & VS Code
Database Service	1,500	0	H2 in-memory DB used for free
UI Libraries or Templates	500	0	Tailwind CSS (open-source)
Miscellaneous	300	0	Used existing systems, no external cost incurred
Total	**6,000 BDT**	**0 BDT**	Based on academic and open-source development

Team Work

Team Members and Their Contributions:

Name	**Student ID**	**Responsibilities**	**Estimated Time Spent (Hours)**
Supan Roy	232-15-716	Frontend development (UI design, Tailwind CSS integration, product pages)	25
Abdullah Al Noman	232-15-797	Backend development (Spring Boot configuration, controllers, services, mailing system)	150
Md. Shahinur Kabir Antor	232-15-159	Database, Survey and Data Collection	15
Nur Sayda	232-15-437	Documentation and Project Slide	10
Shakira Rahman Simi	232-15-723	Report Writing and Project Slide	10

4.3 Complex Engineering Problem

In this, we discuss the complex engineering problems encountered during the development of the **DIU CampusCart** project. We provide an analysis of how the project required the application of engineering principles to address these problems effectively.

4.3.1 Mapping of Program Outcome

In this section, we map the problems encountered during the development of the **DIU CampusCart** project and the corresponding solutions to the targeted Program Outcomes (POs). This mapping helps demonstrate how the project aligns with the learning outcomes set by the program, ensuring that each stage of the project contributes to our overall educational goals.

Table 4.1: Justification of Program Outcomes

PO's	**Justification**
PO1	Demonstrated understanding of core software development principles, especially with Spring Boot and secure login implementation.
PO2	Applied object-oriented programming and layered architecture concepts effectively for product and user management.
PO3	Designed a socially impactful platform enhancing campus sustainability through internal product exchange and reusability.

4.3.2 Complex Problem Solving

In this section, we map the **Complex Engineering Problem** (CEP) to different problem-solving categories. Below is the detailed mapping, followed by rationale (Table 4.2)

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	**EP2** Range of Conflicting Requiremen ts	**EP3** Depth of Analysis	**EP4** Familiarity of Issues	**EP5** Extent of Applicable Codes	**EP6** Extent Of Stakeholder Involvement	**EP7** Inter-dependence
CO1, CO2, CO3, CO4, KP3, KP4	CO3	CO1, CO2, CO4	-	-	-	-

Rationale for Table 4.2:

- **EP1 (Depth of Knowledge):** CO1–CO4 cover foundational to advanced OOP concepts (e.g., classes, UML, industry standards).
- **EP2 (Conflicting Requirements):** CO4 requires trade-offs in design (e.g., scalability vs. readability).
- **EP3 (Depth of Analysis):** CO3 (UML) and CO4 (solution evaluation) demand critical analysis.

Rationale for KP Mapping:

- **KP3** aligns with CO1–CO3's focus on applying OOP fundamentals.
- **KP4** reflects CO4's higher complexity (evaluating solutions against industry norms).

4.3.3 Engineering Activities

In this section, we map the complex engineering activities to the tasks carried out in the project. Below is the detailed mapping, followed by rationale (Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
CO1, CO2, CO3, CO4	CO1, CO2, CO3, CO4	CO2, CO3, CO4	CO3, CO4	CO1, CO2, CO3, CO4

Rationale for Table 4.3:

- **CO1: EA1, EA2, EA5** – Learning about resource structure (classes/objects), low-to-moderate interactions, and developing familiarity with programming elements.
- **CO2: EA1, EA2, EA3, EA5** – Building logical components with Java, managing interactions in service-controller-DB flows, and applying innovative logic for efficiency.
- **CO3: EA1–EA5** – Complete system analysis and modeling with consideration for environmental impact and interaction with all major software components.
- **CO4: EA1–EA5** – Full-stack development involving multiple resources and systems, innovative feature inclusion (like DIU-only login), and planning for sustainability.

Chapter 5

Conclusion

This chapter summarizes the work done in the CampusCart project, discusses its limitations, and outlines the potential future work. It reflects on the key findings and provides insight into how the system can evolve moving forward.

5.1 Summary

The **DIU CampusCart** project aimed to create a secure, user-friendly classified ads platform exclusively for DIU students. The platform allows students to buy and sell items within their community, ensuring that only DIU-verified users can access the service. Key features such as DIU email verification, product uploads with images, category-based browsing, and secure login were successfully implemented. Using Java Spring Boot for the backend, Tailwind CSS for the frontend, and H2 for the database, the system met all functional requirements. The project was developed with a focus on simplicity, scalability, and security, providing a foundation for future expansion.

5.2 Limitation

While the project met its primary objectives, certain limitations became apparent during development and testing:

- **Limited Scalability:** The system was tested with a small dataset and on local servers. It needs further optimization for handling larger amounts of data and users, especially for real-world deployment.

- **H2 Database Constraints:** The use of H2 as an in-memory database limited long-term data storage. A transition to a more robust database like MySQL or Firebase is necessary for future scalability.

- **Mobile Compatibility:** Although the frontend was responsive, it was not fully optimized for mobile use, limiting accessibility across different devices.

- **No Delivery System:** Currently, the system lacks integration with a delivery mechanism, which could enhance its functionality in a real-world setting.

- **No Payment Gateway Integration:** Currently, the system does not support online payments, which limits its capability for more secure and streamlined transactions.

5.3 Future Work

Looking ahead, several improvements can be made to enhance the functionality and usability of the CampusCart platform:

- **Mobile App**: Developing a mobile version of the platform would improve accessibility and make it more convenient for students to use while on the go.
- **Database Migration**: Migrating from H2 to a more scalable and persistent database like MySQL or

Firebase to support a growing number of users and products.

- **Buyer-Seller Messaging**: Implementing a messaging system would allow direct communication between buyers and sellers, fostering smoother transactions.
- **Payment Integration**: Adding a payment gateway for secure online transactions could streamline the buying process, especially for higher-value items.
- **Delivery Coordination**: Integrating a delivery system for product transactions would add convenience and make the platform more comprehensive for users.
- **Advanced Search & Filters**: Implementing advanced search features with filters based on location, price range, and category would improve user experience.

References

[1] Tailwind Labs, *Tailwind CSS Documentation*. https://tailwindcss.com/docs, 11 March,2025, 16:37
[2] Spring Boot, *Documentations* https://spring.io/projects/spring-boot, 16 March, 2025, 21:11
[3] Baeldung, *H2 Database*, https://www.baeldung.com/spring-boot-h2-database, 17 March, 2025, 11:40