

PROJECT CHARTER

for (DeSales Exchange Hub)

more detailed project information

Prepared for
Desales University
Street Address
City, State, Zip
Attn: Name, title
email address

Prepared By
Carlos Parra Camacho
Desales University
Street Address
City, State, Zip
cp3274@desales.edu
+1 (484) 666-2303

Contents

Prepared for Desales University Street Address City, State, Zip Attn: Name, title email address0

Prepared By Carlos Parra Camacho Desales University Street Address City, State, Zip
cp3274@desales.edu +1 (484) 666-2303.....0

Executive Summary2

Project Purpose.....2

Business Objectives2

Project Details2

Requirements2

Scope Statement.....3

Deliverables3

Estimated Schedule.....3

Procurements4

Risks4

Completion Criteria4

Define Project Success.....5

Assumptions5

Constraints.....5

Project Authorization.....6

EXECUTIVE SUMMARY

DeSales Exchange Hub is a web-based marketplace application designed specifically for the international and exchange student community at DeSales University. Every semester, incoming students struggle to find affordable essential items (such as furniture, textbooks, and appliances) for their short-term stay, while departing students often discard usable goods due to a lack of convenient resale options.

This project aims to bridge this gap by providing a centralized, secure, and user-friendly platform where students can buy, sell, and rent items within the campus ecosystem. By facilitating these peer-to-peer transactions, the project will promote sustainability, reduce waste, and lower the financial burden for students. The application will be developed using Python (Flask) and SQLite, ensuring a robust and scalable solution that meets the specific needs of the university community.

PROJECT PURPOSE

To develop a web-based platform specifically designed for international and exchange students at DeSales University to buy, sell, or rent essential items (furniture, appliances, textbooks, sports equipment, etc) securely within the campus community.

BUSINESS OBJECTIVES

Sustainability: Promote a circular economy on campus by reusing items instead of throwing them away at the end of the semester.

Cost Reduction: Help incoming students acquire essentials at a lower cost compared to retail prices.

Community Building: Create a centralized, trusted hub for student-to-student transactions, eliminating the uncertainty of external platforms like Facebook Marketplace.

PROJECT DETAILS

The project consists of a web application built with Python (Flask) and SQLite. It will feature user authentication (limited to university emails if possible), a product listing catalog with categories (Dorm essentials, Sports, Books), and a contact system between buyers and sellers.

REQUIREMENTS

User Account Management: The system must allow users to register with an email and password, log in, and log out securely.

Product Listing (CRUD): Users must be able to Create, Read, Update, and Delete their own sales listings. Each listing should include a title, description, price, category, and an image.

Search and Filter: The application must allow users to browse items and filter them by category (e.g., Furniture, Books, Electronics) or price range.

Contact Mechanism: The product page must display the seller's contact information (email) to logged-in users to facilitate the transaction.

Responsive Interface: The web application must be accessible and usable on both desktop computers and mobile devices (using Bootstrap).

Data Persistence: All user and product data must be stored and retrieved accurately from a relational database (SQLite).

SCOPE STATEMENT

In Scope: User registration/login, posting items (image + description + price), searching/filtering items, and a dashboard to manage own listings.

Out of Scope: Processing payments directly in the app (transactions will be cash/Venmo in person) and shipping logistics.

DELIVERABLES

Fully functional Web Application source code (Python/HTML/CSS).

User Manual / Documentation.

Video demonstration (5 min).

Prototype 1, Prototype 2, and Final presentations.

ESTIMATED SCHEDULE

Milestone / Phase	Goal / Description	Due Date (Approx)
Project Setup	Set up the development environment	Feb 8
Prototype 1	User Auth & Structure: Login/Register pages created. Database designed (User table + Product table). Basic Layout (HTML/Sass) implemented.	Feb 18
Development Phase A	Create Listings: Backend logic to allow users to fill a form and save an item to the database.	Mar 1
Prototype 2	Display Listings: Homepage showing items from the database. Product Detail page working. Ability to delete own items.	Mar 23
Development Phase B	Search & Polish: Add search bar functionality (filter by category). Improve CSS/Design. Add "Contact Seller" button (shows email).	Apr 1

Milestone / Phase	Goal / Description	Due Date (Approx)
Final Polish	Testing & Documentation: Record video, write report, fix bugs.	Apr 20
Final Submission	Submit final code and presentation.	Apr 20

PROCUREMENTS

Description	Source	Estimated Cost
Integrated Development Environment (IDE)	VS Code (Microsoft)	\$0 (Free License)
Backend Framework	Python & Flask (Open Source)	\$0
Database Engine	SQLite (Open Source)	\$0
Frontend Framework	Bootstrap 5 (Open Source)	\$0
Version Control Repository	GitHub	\$0
Development Hardware	My Laptop	\$unestimable

RISKS

Technical Curve: Developing in Python for the first time might slow down the initial backend setup. *Mitigation:* Use lightweight framework (Flask) and extensive documentation.

Data Persistence: Managing user uploads (images) efficiently without cloud storage. *Mitigation:* Store images locally in the server folder for the scope of this project.

COMPLETION CRITERIA

The project is considered complete when a user can successfully register, post an item for sale, and another user can find that item via search and view the contact details.

DEFINE PROJECT SUCCESS

To be considered successful, the project must meet the following criteria:

Functionality: All "Core Features" (Registration, Posting, Searching) must work without errors during the final demonstration.

Deadlines: Successful submission of the Project Charter, Prototype 1, Prototype 2, and Final Code by their respective due dates.

Adoption: The final prototype must be capable of handling at least 5 simultaneous dummy user accounts and 20 product listings during testing.

Documentation: Submission of a complete User Manual that allows a non-technical user to run the application locally.

Skill Acquisition: The developer demonstrates a working understanding of Python/Flask architecture through the code structure and Q&A session.

ASSUMPTIONS

Learning Curve: It is assumed that i will acquire the necessary Python and Flask syntax knowledge within the first 4 weeks of the schedule.

Deployment: The project is assumed to be presented and graded on a "Localhost" environment; a live public server deployment is not a requirement for completion.

User Hardware: It is assumed that end-users have access to a device with a modern web browser (Chrome, Firefox, or Edge) and internet connection.

Transaction Handling: It is assumed that financial transactions (payments) occur offline (cash/Venmo) between students; the app handles only the connection, not the money (for now).

CONSTRAINTS

Time: The absolute deadline is the Final Exam date (May 2024). All development must cease by this date.

Budget: The project has a strictly zero-dollar (\$0) budget. No paid software, licenses, or hosting services can be used.

Resources: This is an individual project. All design, coding, testing, and documentation must be performed by a single developer.

Technology Stack: The backend must be developed using Python to fulfill the student's personal learning objectives, which introduces a learning constraint.

Compliance: The content must adhere to DeSales University's code of conduct (e.g., no illegal items allowed on the marketplace).

PROJECT AUTHORIZATION

Date: _____

By initialing each page and signing below, I _____, the Project Sponsor, approve the project described herein and authorize it to begin.

By: _____
Signature of Project Sponsor

Project Sponsor Printed Name