HuskyShare: A Northeastern Community for Resource Reuse

PROJECT PROPOSAL FINAL

EMGT 5220 Engineering Project Management

Spring 2025

Team 7

Team Members:

1. Bijarniya, Mayur Mahavir
2. Chilkuri, Krithik Reddy
3. Maram, Sreya Reddy
4. N, Dheeraj Reddy
5. Ramasubramanian, Jaanaki Raman
6. Telakala, Shravya Sree Reddy

Point of Contact:

Name: Ramasubramanian, Jaanaki Raman
Email: ramasubramanian.j@northeastern.edu
Phone Number: (857) 506-438

TABLE OF CONTENTS

LETTER OF TRANSMITTAL

Date: April 15, 2025

Prof. Himlona Palikhe
Graduate School of Engineering
Northeastern University
130 Snell Engineering
360 Huntington Avenue
Boston, MA 02115

Dear Professor Palikhe,

We are pleased to submit our final project proposal, *HuskyShare: A Northeastern app for Resource Reuse*, developed for EMGT 5220 – Engineering Project Management. This project addresses the need for a secure, sustainable, and affordable platform for peer-to-peer exchange of essential college items among students, faculty, and staff at Northeastern University.

HuskyShare is a mobile application that facilitates safe and convenient resource sharing through Northeastern SSO authentication, offering verified access and a seamless user experience. The proposal outlines the project's purpose, technical architecture, implementation strategy, stakeholder roles, risk mitigation, and financial planning. Developed using agile methodology, the project includes detailed scheduling, bi-weekly sprints, and resource allocation. The total estimated budget is $249,076, with justifications covering labor, infrastructure, materials, and contingency planning.

We have applied project management best practices throughout—from feasibility and WBS development to performance monitoring and stakeholder coordination—to ensure a secure, scalable, and impactful solution. The platform is designed not only to reduce campus waste but also to foster a culture of collaboration and sustainability.

We are sincerely grateful for your guidance throughout the semester. Your insights have been instrumental in shaping our direction and decision-making. We look forward to your feedback and appreciate the opportunity to present our work.

Sincerely,
Team 7 – EMGT 5220
Bijarniya, Mayur Mahavir
Chilkuri, Krithik Reddy
Maram, Sreya Reddy
N, Dheeraj Reddy
Ramasubramanian, Jaanaki Raman
Telakala, Shravya Sree Reddy

EXECUTIVE SUMMARY

HuskyShare is a comprehensive mobile application developed to enable a secure, affordable, and sustainable exchange of essential goods within the Northeastern University community. The project was initiated in response to two major challenges faced by students and staff: the high cost of acquiring frequently needed items such as textbooks, furniture, and electronics, and the significant waste generated during move-in and move-out periods. On average, a college student generates approximately 640 pounds of waste annually, a portion of which stems from discarded but reusable items. Currently, students often turn to unofficial platforms such as Craigslist or Facebook Marketplace, which lack verification, are susceptible to scams, and offer little support in ensuring fair and transparent transactions. HuskyShare fills this gap by introducing a reliable, university-exclusive platform where students, faculty, and staff can safely buy, sell, or donate used items through a verified, peer-to-peer system.

Technically, the application is built on a modern and scalable architecture using industry-standard technologies. The frontend is developed using React Native**,** allowing seamless performance across both iOS and Android platforms. The user interface is first prototyped using Figma**,** ensuring intuitive and accessible design based on actual student feedback. The backend is powered by Node.js and Express.js, with GraphQL APIs ensuring efficient data retrieval. For cloud infrastructure, the project employs MongoDB Atlas for NoSQL database management, AWS Lambda for scalable serverless computing, AWS S3 for secure and scalable media storage, and Firebase Cloud Messaging to enable real-time notifications and user engagement. In terms of security, HuskyShare enforces AES-256 encryption and TLS 1.3 protocols to protect data in transit and at rest. Furthermore, all users must authenticate via Northeastern's Single Sign-On (SSO) system, ensuring that only valid members of the university community can access the platform.

The project is organized into four structured phases: the Initiation Phase (June 2 - June 11, 2025) focuses on defining the scope and conducting a feasibility study; the Planning Phase (June 12 - July 16, 2025) includes requirement gathering, system design, and resource planning; the Execution Phase (July 17 - January 7, 2026) involves full-scale development, testing, and system integration; and finally, the Closure Phase (January 8 - February 10, 2026) concludes the project with final deployment, stakeholder training, and documentation handover. Agile methodology is employed throughout, utilizing bi-weekly sprints, sprint reviews, and retrospective sessions to maintain development efficiency and adaptability. Project progress is monitored using Gantt charts for milestone tracking and Earned Value Management (EVM) to evaluate cost and schedule performance. Comprehensive testing—including unit testing, integration testing, and User Acceptance Testing (UAT) - is conducted before launch to ensure platform stability and user satisfaction.

The total project budget for HuskyShare is $249,076.27, comprising $178,925.16 in labor costs and $48,544 in material costs. An 8% labor contingency and a 15% material contingency are included to manage unforeseen changes. Material expenses cover hardware, software, licenses, and servers, supporting development, testing, and deployment.

HuskyShare aims to not only reduce the financial burden on students but also promote sustainability and responsible consumption practices on campus. It encourages reuse over disposal, thereby minimizing environmental impact and aligning with Northeastern University's commitment to eco-conscious innovation. Additionally, the platform fosters a stronger sense of community by building a culture of sharing, transparency, and student-to-student support. As a non-profit, community-centered initiative developed specifically for the Northeastern population, HuskyShare offers long-term value through social impact, operational efficiency, and digital innovation.

1. INTRODUCTION

1.1 Problem

College life demands numerous essentials such as textbooks, electronics, and furniture that can be expensive and difficult to acquire affordably. People often resort to unofficial platforms or unreliable sources, leading to wasted money, logistical challenges, and potential scams. Simultaneously, many useful items go unused or are discarded, contributing to environmental waste.

> *"On average, each college student generates about 640 pounds of waste annually, with the climactic move-out contributing a significant portion."*

With sustainability becoming a critical global concern, Northeastern University lacks a structured and secure system to facilitate the reuse of essential resources within its community. Current solutions are fragmented, inconvenient, and lack verification, making it difficult to confidently buy, sell, or donate pre-owned items.

The challenge is clear: How can a reliable, secure, and user-friendly platform that enables students, faculty and staff to efficiently exchange resources while promoting affordability and sustainability be created?

1.2 Solution

The proposed solution is HuskyShare, a mobile application designed to streamline peer-to-peer resource exchange at Northeastern University. This intuitive platform will provide students, faculty, and staff with a secure and convenient marketplace for buying, selling, and donating essential items. Key features include Verified Listings & User Authentication, Smart Search & Category Filters, In-App Messaging, Seamless User Experience, and Sustainability Impact Metrics. This project will be developed using an agile methodology, ensuring continuous improvement through iterative development, user feedback integration, and optimized security protocols.

HuskyShare operates as a free-to-use platform for students, faculty, and staff at Northeastern University, ensuring accessibility without any subscription fees or in-app advertisements that could disrupt the user experience, with funding support from the university to sustain its operations. The platform is designed to facilitate the sustainable exchange of essential resources while maintaining a seamless and secure user experience. By leveraging university events and student organizations for promoting the application, HuskyShare is anticipating a high adoption rate, making it a transformative platform in sustainable living.

2. PURPOSE & OBJECTIVES

2.1 Purpose

The purpose of this project is to provide northeastern members with affordable access to essential college items without the high cost associated with purchasing new products by addressing the gap in reliable resources and guidance. One of its aims is to promote the reuse of college essentials that are still in good condition, helping reduce waste and supporting durability by preventing the need for new purchases. Additionally, the project aims to create a trustworthy platform for huskies to find affordable options, reducing their reliance on unofficial platforms that may lead to confusion or scams such as fraudulent listings, non-delivery, and price manipulation, etc. By building a stronger network within the university community, the project fosters collaboration and resource sharing, ensuring more informed and sustainable choices.

2.2 Objectives

To achieve the project's purpose, HuskyShare will be developed with a fully functional mobile application featuring a user-friendly interface and robust security mechanisms with Northeastern SSO authentication to ensure a safe and seamless experience. Additionally, it will include search, filter, and category-based navigation for effortless item discovery, along with in-app messaging to facilitate secure and direct communication between users. To ensure long-term success, the project will incorporate feedback-driven improvements post-launch, allowing for continuous optimization and scalability.

3. TECHNICAL OVERVIEW

The technical summary provides an in-depth explanation of how HuskyShare ensures high performance, scalability, and security, along with efficiency in processing and responsiveness in the system. Dynamic resource management ensures the optimal use of power, reduces the latency, and offers more significant interactions among the users. All the elements in the architecture are designed to offer a smooth experience, security compliance, and long-term sustainability.

Mobile Application Architecture

HuskyShare follows a three-tier mobile architecture to provide a scalable and smooth user experience.

- **Frontend (React Native):** The app will be designed using Figma to create a prototype before it's built using React Native. It will provide a seamless UI/UX with the same cross-platform look and feel on both iOS and Android, while it supports functionality such as browsing products, user profiles, and interactive forms.
- **Backend (Node.js, Express.js, GraphQL):** The API service manages authentication, messaging, and data transfer using GraphQL, enabling efficient retrieval of relevant information with improved search performance.
- **Cloud Infrastructure (MongoDB Atlas, AWS Lambda, AWS S3, Firebase Cloud Messaging):** It facilitates secure storage of data, real-time processing of work, and system alerts. MongoDB Atlas handles scalable NoSQL storage, AWS Lambda does the work in the background, AWS S3 manages media file storage such as item images, and Firebase Cloud Messaging delivers instant notifications.

Authentication & Security

- **User Authentication:** The platform integrates Northeastern SSO (Single Sign-On) to authenticate users securely using the university's existing identity management system. This method ensures that only verified members of the Northeastern community can access the platform, reducing the risk of fake profiles and fraudulent transactions.
- **Data Encryption:** Sensitive user information, including personal details and transaction history, will be encrypted using AES-256 encryption, a globally recognized standard for secure data storage. This ensures that even in the event of a security breach, the data remains unreadable to unauthorized entities.
- **Secure Communication:** All communication between the client application and backend services will be encrypted using TLS 1.3, providing robust protection against data interception, man-in-the-middle attacks, and other cybersecurity threats.

Performance Optimization & Compliance

- Guarantees application quality and stability through the execution of unit tests, integration tests, and user acceptance tests prior to deployment.
- Enhances data retrieval through GraphQL API to alleviate the backend workload and enhance search efficiency.
- Adjusts traffic dynamically by means of AWS Auto-Scaling to maintain consistent performance.
- Ensures security by conducting regular audits, penetration tests, and compliance to law-based privacy practices.
- Confirms compliance to Google Play and Apple App Store by monitoring them at regular intervals.

4. IMPLEMENTATION PLAN

4.1 WBS:

The Work Breakdown Structure (WBS) divides the HuskyShare project into four distinct phases. Each phase is designed to ensure smooth transitions, risk mitigation, and alignment with the overall project objectives.

1. **Initiation Phase:** The Initiation Phase focuses on defining the project scope, identifying key stakeholders, and establishing project governance. A feasibility study is conducted to assess the technical, financial, and operational viability of the project. Risk identification and mitigation strategies are outlined to ensure smooth execution. The Project Charter is developed, detailing the high-level project goals, deliverables, and stakeholder expectations. Approval from key decision-makers is secured before proceeding to the next phase.
2. **Planning Phase:** The Planning Phase involves gathering user requirements, defining technical specifications, and outlining system architecture. A detailed risk management plan is developed to address potential challenges. Resource allocation strategies are implemented, ensuring efficient distribution of personnel, infrastructure, and financial assets. A comprehensive project schedule is established, detailing major milestones and deliverables. Prototyping of UI/UX components is initiated, ensuring usability and accessibility standards are met.
3. **Development and Execution Phase:** The Execution Phase involves the development and integration of core system components. Backend development focuses on database management, authentication mechanisms, and API integrations. Frontend development prioritizes responsive UI, real-time communication features, and accessibility enhancements. Security protocols such as AES-256 encryption and TLS 1.3 are integrated to ensure data integrity and compliance with institutional and regulatory standards. Agile sprint cycles are implemented to facilitate iterative testing and continuous improvement.
4. **Closure Phase:** The Closure Phase ensures the finalization and transition of the system into operational use. A comprehensive testing framework is executed, including unit, integration, and user acceptance testing (UAT). Stakeholder reviews are conducted, and final adjustments are made based on feedback. The system is deployed on the designated cloud infrastructure, and end-user training manuals are provided. Documentation of the entire project lifecycle is completed, and a final review is conducted to assess project outcomes.

A detailed Work Breakdown Structure (WBS) is available in **Appendix A .**

4.2 Schedule

A milestone-driven approach has been adopted to ensure adherence to project timelines. The project schedule outlines key activities, interdependencies, and expected completion dates. Each phase of the project has designated timelines with phase-wise completion milestones. The Gantt Chart provides a visual representation of the project timeline, illustrating task dependencies and durations. It has been developed in Microsoft Project to track progress and monitor deviations. The project is structured into bi-weekly sprints, allowing iterative refinements and ensuring timely course corrections. A detailed project schedule is available in **Appendix B .**

4.3 Responsibility Chart

A Responsibility Assignment Matrix (RACI) has been developed to define roles and responsibilities for all project tasks. This ensures accountability, minimizes conflicts, and optimizes task execution efficiency.

The RACI Matrix classifies project roles into the following categories:

1. **Responsible (R):** Individuals who execute tasks.
2. **Accountable (A):** Individuals with decision-making authority.
3. **Consulted (C):** Subject matter experts providing guidance.
4. **Informed (I):** Individuals who receive updates on progress.

The matrix clearly delineates the responsibilities of the Project Manager, Developers, UI/UX Designer, Security Analyst, Quality Assurance engineer, and other Stakeholders. It provides clarity on project ownership and ensures smooth coordination among teams. A detailed RACI matrix is available in **Appendix C .**

4.4 Resource Allocation

The HuskyShare project follows a structured human resource allocation plan to ensure optimal workload distribution and project efficiency. Team members are assigned roles based on expertise, project requirements, and estimated working hours to ensure smooth execution.

The Project Manager (PM) and Product Owner (PO) oversee project planning, execution, and stakeholder coordination. The Backend Developer (BD) and Frontend Developer (FD) handle server-side logic, API development, and user interface implementation. The UI/UX Designer (UX) ensures an intuitive and user-friendly experience, while the Security Specialist (SS) is responsible for system security and compliance. The Cloud Engineer (CE) manages infrastructure deployment and performance, and the QA Engineer conducts testing and debugging. Additionally, the Marketing Lead (ML) handles user engagement and outreach, while the Legal Manager (LM) ensures compliance with regulations.

This resource allocation strategy ensures efficient execution, role clarity, and project success. A detailed breakdown of individual responsibilities and work distribution is provided in **Appendix D .**

4.5 Stakeholders

Refer to **Section 8** for team member credentials and **Appendix D f**or resource allocation.

Project Team:

- **Project Manager:** Chilkuri, Krithik Reddy
- **Product Owner:** Bijarniya, Mayur Mahavir
- **Product Manager:** Telakala, Shravya Sree Reddy
- **Relationship Manager**

The project team consists of a Project Manager, Product Owner, Product Manager and Relationship Manager. They are responsible for overseeing project execution, ensuring alignment with the defined objectives, and coordinating between different teams to maintain smooth operations.

Design Team:

- **UI/UX Designer:** Ramasubramanian, Jaanaki Raman

The UI/UX Designer ensures the platform is designed with an intuitive user interface, prioritizing accessibility and usability based on client needs. Through research-driven prototyping and user feedback, the design team refines the interface for an optimal experience.

Development Team:

- **Backend Developer:** Maram, Sreya Reddy

The Backend Developer is responsible for building and maintaining the server-side infrastructure, database management, and API integrations, ensuring seamless data flow between different components of the platform.

- **Frontend Developer:** N, Dheeraj Reddy

The Frontend Developer focuses on crafting a responsive, interactive, and accessible user experience, ensuring that the platform meets both functional and aesthetic requirements.

Data Team:

- **Data Scientist**

The Data Scientist analyses data from various sources to enhance user experiences on HuskyShare. This includes personalized recommendations for available resources based on individual needs and behavioural patterns.

Quality Assurance Engineer:

- **Quality Assurance**

The Quality Assurance engineer is responsible for identifying and resolving usability issues through extensive testing methodologies, ensuring platform reliability before deployment. They conduct unit testing, integration testing, and user acceptance testing (UAT) to mitigate potential risks.

Cloud & Security Team:

- **Cloud Engineer**

The Cloud Engineer ensures that the HuskyShare platform is secure, scalable, and optimized by leveraging cloud-based infrastructure. They oversee server management, performance monitoring, and deployment processes to maintain system stability.

- **Security Specialist**

The Security Specialist is responsible for implementing data security protocols, encryption standards, and vulnerability assessments to safeguard user data and maintain compliance with regulatory policies.

Marketing Team:

- **Marketing Lead**

The Marketing Lead oversees the promotional and outreach strategy for the HuskyShare platform, ensuring user adoption and engagement. Responsibilities include creating awareness campaigns, facilitating onboarding sessions, and gathering user feedback for continuous improvement.

Legal Team:

- **Legal Manager**

The Legal Manager reviews HuskyShare's platform functionalities to ensure compliance with all applicable data protection laws and privacy regulations.

5. EXECUTION PLAN

5.1 Project Monitoring

To ensure the timely and successful execution of HuskyShare, a robust monitoring framework will be adopted throughout the project lifecycle. This framework emphasizes data-driven insights, user-centric feedback loops, and agile tracking techniques to maintain alignment with scope, schedule, and quality.

Progress Reporting: Weekly SCRUM meetings will be conducted where each sprint's progress will be evaluated using visual tools like burn-down charts and Kanban boards. Project status will be communicated through dashboards summarizing completion percentages, blockers, and sprint velocity.

Performance Metrics: User interaction metrics (bounce rate, session length, conversion rate) and platform performance indicators (response time, crash rate, load time) will be monitored through Firebase Analytics and AWS CloudWatch.

Feedback Analysis: Post-sprint user feedback and prototype walkthroughs will guide iterative improvements. Usability scores and survey insights will help prioritize new features and UI/UX adjustments.

Earned Value Management (EVM): EVM will be used to assess cost and schedule performance across phases. Key metrics like Cost Performance Index (CPI) and Schedule Performance Index (SPI) will be calculated bi-weekly.

Risk & Issue Logs: Identified risks and active issues will be tracked in Jira with mitigation actions, responsible owners, and resolution timelines.

These monitoring efforts ensure proactive decision-making and provide a foundation for continuous delivery and improvement. All key findings and changes will be transparently shared with stakeholders in monthly review meetings.

5.2 Project Control

Project control will focus on maintaining scope, quality, and timeline through integrated tools and methodologies. Agile practices and milestone-based tracking ensure tight control over resources and deliverables.

Schedule Control: Milestone adherence will be tracked using the Gantt chart developed in Microsoft Project. Bi-weekly sprint retrospectives will evaluate progress and adjust timelines if necessary.
Scope Control: A defined change management protocol will ensure that scope changes are reviewed, approved, and documented before implementation. This prevents scope creep while maintaining flexibility for value-driven adjustments.
Quality Assurance: Test cases will be documented and executed during every development sprint. Bugs will be logged in Jira, categorized by severity, and assigned based on resolution SLAs. Continuous Integration/Continuous Deployment (CI/CD) pipelines will automate code testing and deployment checks.
Compliance Audits: Regular internal audits will verify adherence to Northeastern's IT guidelines, App Store policies, and privacy regulations (FERPA, GDPR where applicable).

Responsibility Assignment:

Control Area	Responsible Role
Scope	Product Owner
Schedule	Project Manager
Quality	QA Engineer
Security & Compliance	Security Specialist
Resource & Budget	Product Manager

Table: 5.2.1 Responsibility Assignment

5.3 Project Auditing

To maintain transparency and validate alignment with objectives, comprehensive auditing will be carried out across multiple dimensions of the project.

- **Scope Audit:** Verifies that project activities and outcomes match the approved charter. Changes to functionality or feature set will be reviewed for scope creep mitigation.

- **Requirements Audit:** Cross-checks functional and non-functional requirements defined in the planning phase against actual implemented features. Confirms stakeholder involvement and requirement traceability.

- **Resource Audit:** Evaluates the sufficiency and utilization of personnel, software, and cloud infrastructure. Allocation mismatches or bottlenecks will be flagged for resolution.

- **Timeline Audit:** Assesses task durations and dependency management. Deviations will be logged and reviewed for systemic or operational causes.

- **Security Audit:** Ensures that AES-256 encryption, TLS 1.3 protocols, and user authentication through NEU SSO are properly implemented and tested. Penetration test results and access logs will be part of this audit.

- **Budget Audit:** Verifies that actual expenses align with the approved budget, and contingency usage is justified. Cost overrun analysis will be conducted if thresholds are breached.

- **Communication Audit:** Ensures stakeholder updates, meeting cadences, and documentation flows are being followed to prevent misalignment or misinformation.

Each audit will result in a report of findings, corrective actions, and future recommendations to the core leadership team and stakeholders. These reports will be stored in a shared repository for recordkeeping and future reference.

5.4 Project Closure

The closure phase marks the formal completion and handover of the HuskyShare platform. It is essential to document outcomes, evaluate project performance, and ensure knowledge transfer for post-deployment continuity.

- **System Handover:** Final version of HuskyShare will be deployed on designated production infrastructure. Access credentials, monitoring tools, and system documentation will be handed over to the university's designated IT team.

- **Final Testing & Review:** A full suite of UAT, regression testing, and security validation will be executed before final approval. Any residual bugs or feature requests will be added to the post-launch backlog.

- **Documentation & Manuals:** Includes detailed user guides, admin dashboards, maintenance procedures, and FAQs. These will be distributed in PDF format and via the project repository.

- **Training & Orientation:** Demo sessions and training workshops will be conducted for key university stakeholders, including IT staff, student ambassadors, and marketing teams.

- **Performance Review & Lessons Learned:** Stakeholder surveys and performance analytics will be gathered to evaluate platform adoption and usage. Lessons learned and recommendations will be compiled into a closure report.

- **Acknowledgments & Communication:** A formal closure communication will be sent across all stakeholder groups, recognizing contributions and announcing full project completion. The Relationship Manager will facilitate this process.

6. RISK ASSESSMENT MANAGEMENT PLAN

Throughout the development and deployment of HuskyShare, various risks-both technical and organizational may arise that could impact project timelines, functionality, adoption, or compliance. A proactive approach to risk assessment and mitigation has been adopted to ensure smooth execution across all phases. The identified risks have been categorized based on their likelihood and potential impact on the project. Each risk is evaluated to determine its overall risk level and the appropriate response strategy.

The table below summarizes the key risks along with their mitigation plans:

ID	Risk Description	Likelihood	Impact	Risk Level	Mitigation Strategy
R1	Low user adoption	Medium	High	High	Promote app via student orgs, events, email campaigns, incentives
R2	Security breach or data leak	Low	High	Medium	AES-256 encryption, TLS 1.3, regular penetration testing
R3	Technical delays in app development	Medium	Medium	Medium	Agile sprints, sprint retrospectives, prioritize core features
R4	Inaccurate/misleading listings	High	Medium	High	SSO authentication, user reviews, moderation/report tools
R5	Server downtime / infrastructure failures	Low	Medium	Low	AWS Auto-scaling, uptime monitoring, fallback servers
R6	Approval delays from university stakeholders	Medium	High	High	Early engagement with IT, Legal, and Student Services
R7	Data privacy non-compliance	Low	High	Medium	Legal review pre-launch, minimal data collection
R8	Budget overrun due to underestimated resource cost	Medium	Medium	Medium	Regular budget reviews, contingency buffer, strict scope control
R9	Poor user experience (bugs, lag, crashes)	Medium	Medium	Medium	Continuous testing, UAT, staged rollout
R10	Lack of meaningful user feedback post-launch	Medium	Low	low	In-app feedback tools, surveys, focus groups
R11	App store rejection or policy violations (Apple/Google)	Low	High	Medium	Follow app store guidelines, regular updates and audits
R12	Misuse of platform (e.g., selling banned/unsafe items)	Medium	Medium	Medium	Community guidelines, flag/report system, admin intervention
R13	Legal disputes from user transactions gone wrong	Low	High	Medium	Terms of Service, liability disclaimers, transaction policy page

Table 6.1 Risk Table

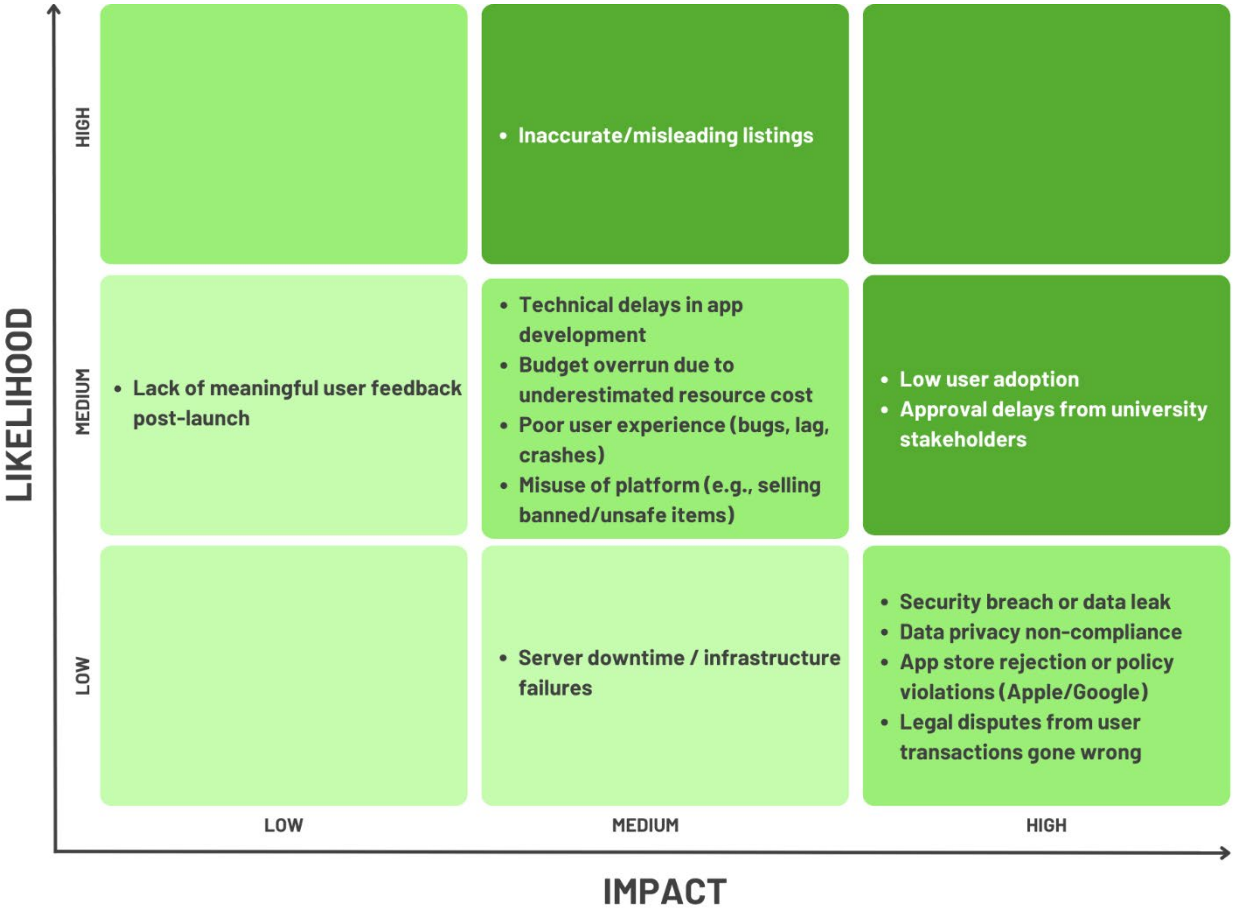
6.1 Risk Management Strategy

To effectively manage the risks identified above, HuskyShare incorporates a multi-faceted strategy involving technical safeguards, procedural controls, and engagement with university stakeholders. The following actions will be undertaken as part of the mitigation plan:

- **User Engagement Initiatives**: Collaboration with student organizations, launch campaigns, and referral incentives will be used to ensure high initial adoption and sustained usage.
- **Security Compliance**: All sensitive data will be encrypted using AES-256, with secure communications ensured via TLS 1.3. Penetration testing and source code reviews will be conducted to eliminate vulnerabilities.

- **Agile Development & Testing**: Agile sprints allow for rapid identification and resolution of development issues. Continuous integration pipelines will support automated testing and performance validation.
- **Listing Moderation**: Users must authenticate via Northeastern SSO, and listing/reporting features will ensure platform integrity by allowing community-driven moderation.
- **Infrastructure Stability**: AWS Lambda and Auto-Scaling groups will be used to guarantee uptime. Firebase logs and AWS CloudWatch will assist in real-time performance tracking.
- **Stakeholder Alignment**: All university-level compliance checks will be initiated early in the planning phase to minimize downstream approval delays.
- **Legal Risk Containment**: Clear usage policies, disclaimers, and user education around safe transactions will minimize liability concerns.

6.2 Risk Matrix



LIKELIHOOD	LOW	MEDIUM	HIGH
HIGH		• Inaccurate/misleading listings	
MEDIUM	• Lack of meaningful user feedback post-launch	• Technical delays in app development • Budget overrun due to underestimated resource cost • Poor user experience (bugs, lag, crashes) • Misuse of platform (e.g., selling banned/unsafe items)	• Low user adoption • Approval delays from university stakeholders
LOW		• Server downtime / infrastructure failures	• Security breach or data leak • Data privacy non-compliance • App store rejection or policy violations (Apple/Google) • Legal disputes from user transactions gone wrong

Figure 6.2.1 Risk Matrix

7. FINANCIAL PLAN WITH BUDGET

7.1 High Level Details

The HuskyShare project is structured with a well-defined financial plan to ensure cost efficiency, resource allocation, and financial sustainability. The estimated total budget for the project is **$249,076** covering all development, implementation, and operational expenses. This budget is designed to ensure that the platform is developed, deployed, and maintained effectively while adhering to industry standards and institutional compliance.

The financial strategy is divided into four major project phases:

- **Project Initiation Phase**: Focuses on defining the project scope, stakeholder engagement, and feasibility analysis.
- **Project Planning Phase**: Includes requirement gathering, technical system design, and resource allocation.
- **Project Execution Phase**: Covers software development, testing, security enhancements, and deployment.
- **Project Closure Phase**: Involves final validation, system documentation, stakeholder sign-offs, and user onboarding.

S. No	Expense Category	Total Estimated Cost ($)
1	Project Initiation Phase	$4,718.60
2	Project Planning Phase	$20,922.44
3	Project Execution Phase	$123,793.04
4	Project Closure Phase	$29,491.08
5	Labor Contingency Reserve (8%)	$14,314.01
6	Material Costs (Hardware/Software)	$48,554
7	Material Contingency Reserve (15%)	$7,283
	Total Estimated Budget	**$249,076**

Table 7.1 High-level budget details

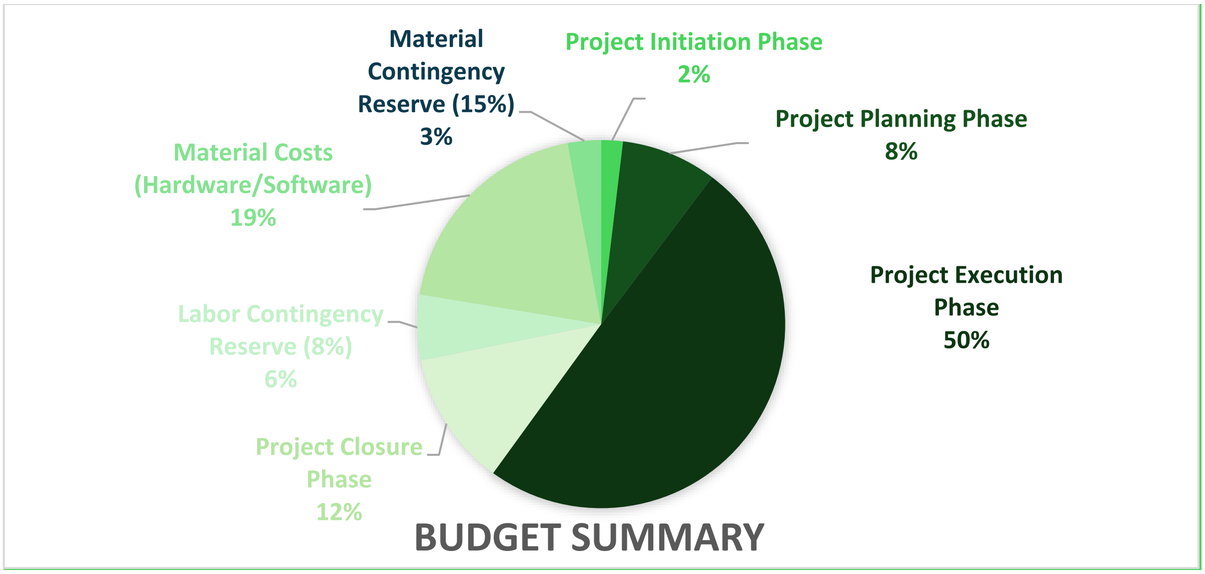


Figure 7.1 Budget Summary Chart

7.2 Budget Justification

Each budget category has been carefully estimated to ensure operational feasibility and cost efficiency.

Labor Costs

The largest budget component is labour, ensuring competitive compensation for project personnel. This includes:

Project Manager, Product Owner, and Product Manager: Oversee project coordination, planning, and execution.

Development Team (Backend Developer, Frontend Developer, UI/UX Designer): Responsible for software development and interface design.

Security Specialist & Cloud Engineer: Ensure system security, compliance, and performance optimization.

Marketing Lead & Relationship Manager: Manage outreach, adoption strategies, and community engagement.

Legal Manager: Ensures regulatory compliance with data protection laws.

The estimation was calculated assuming that Responsible (R) is 100%, Accountable (A) is 75%, Consulted (C) is 50% and Informed (I) is 25% from the RACI matrix for each of the task.

Material Costs

Material expenses cover essential tools such as:

- Cloud hosting and server costs to maintain system scalability.
- Software licenses for development tools, testing frameworks, and encryption mechanisms.
- Security compliance software to safeguard user data and ensure regulatory adherence.

Contingency Reserve

To ensure financial flexibility and risk mitigation throughout the project, a contingency reserve has been incorporated for both labor and material costs. An 8% contingency reserve for labor, amounting to $14,314.01, has been added to accommodate potential changes in scope, resource availability, or development delays. Additionally, a 15% contingency reserve for materials is embedded as miscellaneous costs within each material category-hardware, software, licenses, and servers-to cover unforeseen procurement needs, price fluctuations, or scaling adjustments. These reserves help maintain the project's financial stability and ensure smooth execution without compromising deliverables or quality standards.

A detailed breakdown of activities and associated costs is provided in **Appendix E** .

8. TEAM CREDENTIALS

Bijarniya, Mayur Mahavir is a graduate student in Engineering Management at Northeastern University with a background in Computer Science & Engineering from Lovely Professional University. He has expertise in digital product design, data analysis, and strategic planning, leveraging tools like Python, R, MySQL, MongoDB, and Figma. He played a key role in UniCircle, a university-exclusive social engagement platform designed to help students build meaningful connections through shared interests and campus events. He conducted extensive user research, competitive analysis, and roadmap development to refine the platform's features. He also applied statistical modeling to analyze the impact of new timekeeping rules in the Premier League, assessing game dynamics, player injuries, and competitive balance.

Chilkuri, Krithik Reddy attends Northeastern University as an Engineering Management Master's student with a Mechanical Engineering background. At Shivaganga Polymers, he specialized in supply chain optimization and logistics while performing inventory management and process improvement duties. His skills include Python along with C++, MySQL, Minitab, and IOR tutorial, enabling him to execute work on mechanical designs while performing statistical modeling and predictive analytics. His expertise stems from obtaining certifications in Google's Project Management, enabling him to provide a data-driven strategy for operational excellence.

Maram, Sreya Reddy is a graduate student in Data Analytics Engineering at Northeastern University with a background in Computer Science Engineering. She has developed a variety of academic projects, including a customer segmentation via RFM analysis, a financial risk management report in Power BI, a DBMS for technology staffing optimization, and a personalized book recommendation system. Her work shows expertise in preprocessing, clustering, visualization, and predictive analysis. She is skilled in SQL, Python, Power BI, and machine learning algorithms. She is interested in getting insights out of data and wants to apply them in real-life business scenarios.

N, Dheeraj Reddy a Master's student, pursuing, Engineering Management at Northeastern University after completing his Computer Science Engineering degree from Keshav Memorial Institute of Technology. Working as a business analyst at ATD and as a research intern at the University of Hyderabad, Dheeraj demonstrated his talent for solving problems through data-based decisions. Through his expertise with Python combined with SQL and Excel, Dheeraj successfully implemented projects that optimized production flow, raised machine efficiency by 20%, and decreased expenses.

Ramasubramanian, Jaanaki Raman is a graduate student at Northeastern University, pursuing a Master's in Engineering Management with a focus on Supply Chain and Lean Concepts. He holds a bachelor's in production engineering from PSG College of Technology and has experience as a Supply Chain Program Manager at Rajan Groups and Supply Chain Engineer Intern at PSG Rotating Machinery Division. Skilled in inventory management, procurement, supplier performance, and process improvement, he has successfully led lean initiatives, optimized supply chain efficiency, and enhanced operational performance.

Telakala, Shravya Sree Reddy is a graduate student in Engineering Management at Northeastern University, specializing in Supply Chain, with a background in Mechanical Engineering from Vasavi College of Engineering. With experience as a Design Assistant Intern at Brahmos Aerospace, she contributed to composite material analysis and structural validation for missile storage. Her expertise spans supply chain optimization, data analytics, and project management, demonstrated in projects like optimizing Wayfair's supply chain and analyzing public vs. private health insurance trends. Proficient in Python, R, Tableau, and ANSYS, she combines technical acumen with strategic problem-solving, leadership, and cross-functional collaboration.

APPENDICES

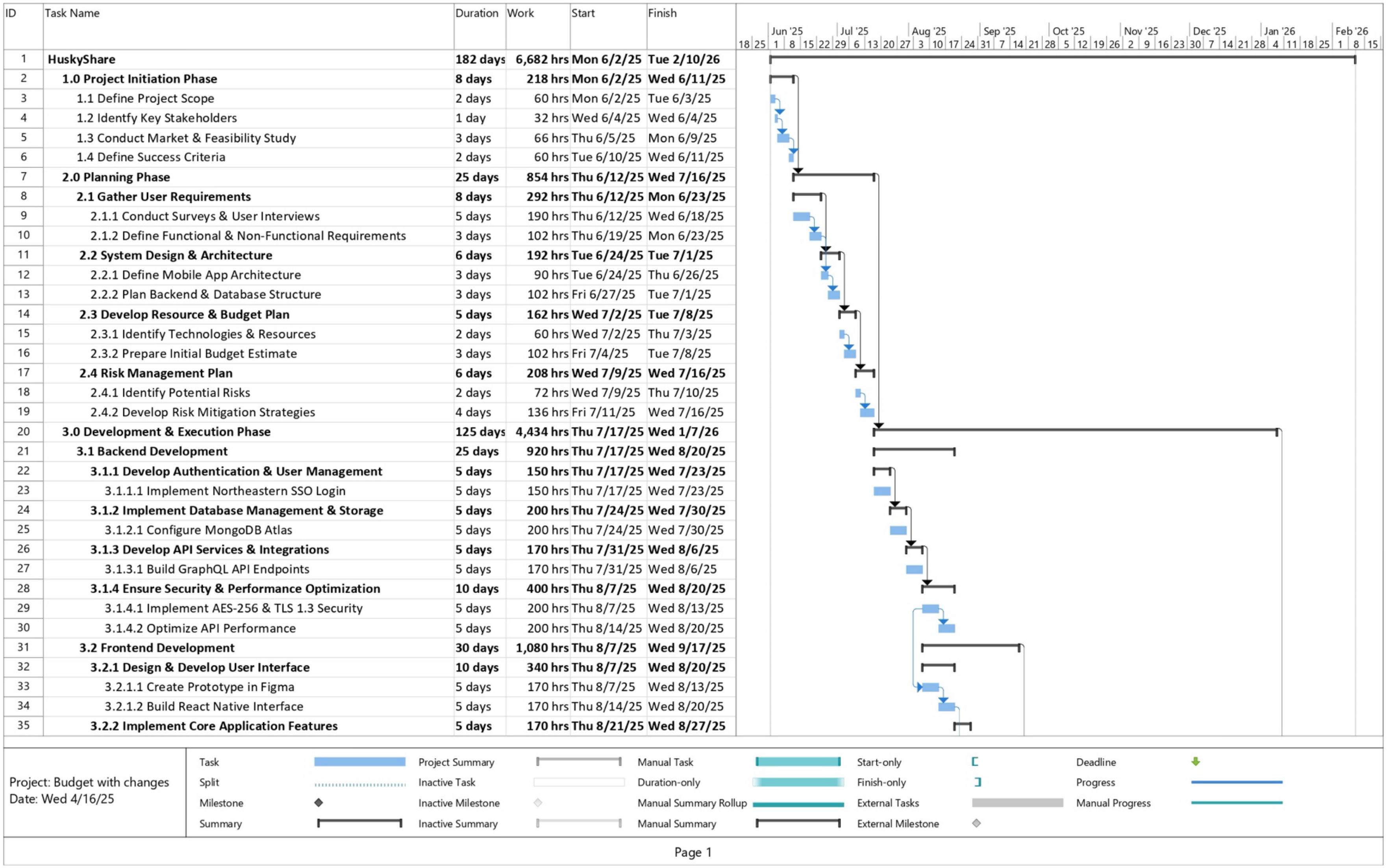
Appendix A: WBS

ID	Task
	HuskyShare
1	**1.0 Project Initiation Phase**
2	1.1 Define Project Scope
3	1.2 Identify Key Stakeholders
4	1.3 Conduct Market & Feasibility Study
5	1.4 Define Success Criteria
6	**2.0 Planning Phase**
7	2.1 Gather User Requirements
8	2.1.1 Conduct Surveys & User Interviews
9	2.1.2 Define Functional & Non-Functional Requirements
10	2.2 System Design & Architecture
11	2.2.1 Define Mobile App Architecture
12	2.2.2 Plan Backend & Database Structure
13	2.3 Develop Resource & Budget Plan
14	2.3.1 Identify Technologies & Resources
15	2.3.2 Prepare Initial Budget Estimate
16	2.4 Risk Management Plan
17	2.4.1 Identify Potential Risks
18	2.4.2 Develop Risk Mitigation Strategies
19	**3.0 Development & Execution Phase**
20	3.1 Backend Development
21	3.1.1 Develop Authentication & User Management
22	3.1.1.1 Implement Northeastern SSO Login
23	3.1.2 Implement Database Management & Storage
24	3.1.2.1 Configure MongoDB Atlas
25	3.1.3 Develop API Services & Integrations
26	3.1.3.1 Build GraphQL API Endpoints
27	3.1.4 Ensure Security & Performance Optimization
28	3.1.4.1 Implement AES-256 & TLS 1.3 Security
29	3.1.4.2 Optimize API Performance
30	3.2 Frontend Development
31	3.2.1 Design & Develop User Interface
32	3.2.1.1 Create Prototype in Figma
33	3.2.1.2 Build React Native Interface
34	3.2.2 Implement Core Application Features
35	3.2.2.1 Develop Listing & Browsing Modules

ID	Task
36	3.2.3 Enhance Search, Navigation & Interaction
37	3.2.3.1 Add Search & Filter Features
38	3.2.4 Integrate Communication & Notifications
39	3.2.4.1 Build In-App Messaging
40	3.2.4.2 Integrate Firebase Notifications
41	3.3 Cloud & Infrastructure Setup
42	3.3.1 Configure Cloud Hosting & Deployment
43	3.3.1.1 Set Up AWS Lambda & Environment
44	3.3.2 Implement System Monitoring & Alerts
45	3.3.2.1 Deploy AWS Auto-Scaling & Alerts
46	3.4 Testing & Debugging
47	3.4.1 Conduct Unit Testing
48	3.4.2 Conduct Integration Testing
49	3.4.3 Conduct User Acceptance Testing (UAT)
50	3.5 Deploy Mobile App
51	3.5.1 Configure Production Environment
52	3.5.2 Publish App on Google Play & Apple Store
53	3.6 Marketing & User Engagement
54	3.6.1 Create Awareness Campaign
55	3.6.2 Conduct Product Demos & Workshops
56	3.7 Performance & Security Monitoring
57	3.7.1 Monitor System Performance & Security
58	3.7.2 Conduct Regular Security Audits
59	3.8 Feedback & Continuous Improvement
60	3.8.1 Gather User Feedback & Analytics
61	3.8.2 Implement Updates & Bug Fixes
62	**4.0 Project Closure**
63	4.1 Final Performance Review
64	4.1.1 Evaluate Performance & Engagement
65	4.2 Closure Report & Future Recommendations
66	4.2.1 Document Outcomes & Recommendations
67	4.3 Official Project Completion
68	4.3.1 Obtain Final Approval & Close Accounts

Table A: Work Breakdown Structure

Appendix B: SCHEDULE

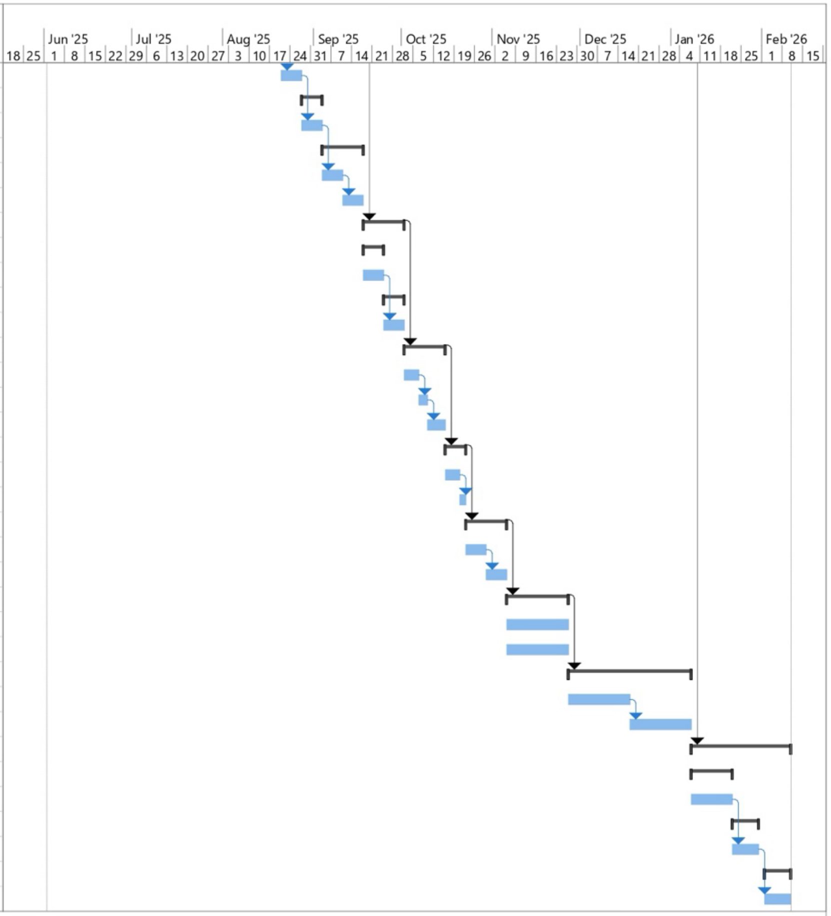


ID	Task Name	Duration	Work	Start	Finish
1	**HuskyShare**	**182 days**	**6,682 hrs**	**Mon 6/2/25**	**Tue 2/10/26**
2	**1.0 Project Initiation Phase**	**8 days**	**218 hrs**	**Mon 6/2/25**	**Wed 6/11/25**
3	1.1 Define Project Scope	2 days	60 hrs	Mon 6/2/25	Tue 6/3/25
4	1.2 Identfy Key Stakeholders	1 day	32 hrs	Wed 6/4/25	Wed 6/4/25
5	1.3 Conduct Market & Feasibility Study	3 days	66 hrs	Thu 6/5/25	Mon 6/9/25
6	1.4 Define Success Criteria	2 days	60 hrs	Tue 6/10/25	Wed 6/11/25
7	**2.0 Planning Phase**	**25 days**	**854 hrs**	**Thu 6/12/25**	**Wed 7/16/25**
8	**2.1 Gather User Requirements**	**8 days**	**292 hrs**	**Thu 6/12/25**	**Mon 6/23/25**
9	2.1.1 Conduct Surveys & User Interviews	5 days	190 hrs	Thu 6/12/25	Wed 6/18/25
10	2.1.2 Define Functional & Non-Functional Requirements	3 days	102 hrs	Thu 6/19/25	Mon 6/23/25
11	**2.2 System Design & Architecture**	**6 days**	**192 hrs**	**Tue 6/24/25**	**Tue 7/1/25**
12	2.2.1 Define Mobile App Architecture	3 days	90 hrs	Tue 6/24/25	Thu 6/26/25
13	2.2.2 Plan Backend & Database Structure	3 days	102 hrs	Fri 6/27/25	Tue 7/1/25
14	**2.3 Develop Resource & Budget Plan**	**5 days**	**162 hrs**	**Wed 7/2/25**	**Tue 7/8/25**
15	2.3.1 Identify Technologies & Resources	2 days	60 hrs	Wed 7/2/25	Thu 7/3/25
16	2.3.2 Prepare Initial Budget Estimate	3 days	102 hrs	Fri 7/4/25	Tue 7/8/25
17	**2.4 Risk Management Plan**	**6 days**	**208 hrs**	**Wed 7/9/25**	**Wed 7/16/25**
18	2.4.1 Identify Potential Risks	2 days	72 hrs	Wed 7/9/25	Thu 7/10/25
19	2.4.2 Develop Risk Mitigation Strategies	4 days	136 hrs	Fri 7/11/25	Wed 7/16/25
20	**3.0 Development & Execution Phase**	**125 days**	**4,434 hrs**	**Thu 7/17/25**	**Wed 1/7/26**
21	**3.1 Backend Development**	**25 days**	**920 hrs**	**Thu 7/17/25**	**Wed 8/20/25**
22	**3.1.1 Develop Authentication & User Management**	**5 days**	**150 hrs**	**Thu 7/17/25**	**Wed 7/23/25**
23	3.1.1.1 Implement Northeastern SSO Login	5 days	150 hrs	Thu 7/17/25	Wed 7/23/25
24	**3.1.2 Implement Database Management & Storage**	**5 days**	**200 hrs**	**Thu 7/24/25**	**Wed 7/30/25**
25	3.1.2.1 Configure MongoDB Atlas	5 days	200 hrs	Thu 7/24/25	Wed 7/30/25
26	**3.1.3 Develop API Services & Integrations**	**5 days**	**170 hrs**	**Thu 7/31/25**	**Wed 8/6/25**
27	3.1.3.1 Build GraphQL API Endpoints	5 days	170 hrs	Thu 7/31/25	Wed 8/6/25
28	**3.1.4 Ensure Security & Performance Optimization**	**10 days**	**400 hrs**	**Thu 8/7/25**	**Wed 8/20/25**
29	3.1.4.1 Implement AES-256 & TLS 1.3 Security	5 days	200 hrs	Thu 8/7/25	Wed 8/13/25
30	3.1.4.2 Optimize API Performance	5 days	200 hrs	Thu 8/14/25	Wed 8/20/25
31	**3.2 Frontend Development**	**30 days**	**1,080 hrs**	**Thu 8/7/25**	**Wed 9/17/25**
32	**3.2.1 Design & Develop User Interface**	**10 days**	**340 hrs**	**Thu 8/7/25**	**Wed 8/20/25**
33	3.2.1.1 Create Prototype in Figma	5 days	170 hrs	Thu 8/7/25	Wed 8/13/25
34	3.2.1.2 Build React Native Interface	5 days	170 hrs	Thu 8/14/25	Wed 8/20/25
35	**3.2.2 Implement Core Application Features**	**5 days**	**170 hrs**	**Thu 8/21/25**	**Wed 8/27/25**

Project: Budget with changes
Date: Wed 4/16/25

Page 1

ID	Task Name	Duration	Work	Start	Finish
36	3.2.2.1 Develop Listing & Browsing Modules	5 days	170 hrs	Thu 8/21/25	Wed 8/27/25
37	**3.2.3 Enhance Search, Navigation & Interaction**	**5 days**	**170 hrs**	**Thu 8/28/25**	**Wed 9/3/25**
38	3.2.3.1 Add Search & Filter Features	5 days	170 hrs	Thu 8/28/25	Wed 9/3/25
39	**3.2.4 Integrate Communication & Notifications**	**10 days**	**400 hrs**	**Thu 9/4/25**	**Wed 9/17/25**
40	3.2.4.1 Build In-App Messaging	5 days	200 hrs	Thu 9/4/25	Wed 9/10/25
41	3.2.4.2 Integrate Firebase Notifications	5 days	200 hrs	Thu 9/11/25	Wed 9/17/25
42	**3.3 Cloud & Infrastructure Setup**	**10 days**	**400 hrs**	**Thu 9/18/25**	**Wed 10/1/25**
43	**3.3.1 Configure Cloud Hosting & Deployment**	**5 days**	**200 hrs**	**Thu 9/18/25**	**Wed 9/24/25**
44	3.3.1.1 Set Up AWS Lambda & Environment	5 days	200 hrs	Thu 9/18/25	Wed 9/24/25
45	**3.3.2 Implement System Monitoring & Alerts**	**5 days**	**200 hrs**	**Thu 9/25/25**	**Wed 10/1/25**
46	3.3.2.1 Deploy AWS Auto-Scaling & Alerts	5 days	200 hrs	Thu 9/25/25	Wed 10/1/25
47	**3.4 Testing & Debugging**	**10 days**	**360 hrs**	**Thu 10/2/25**	**Wed 10/15/25**
48	3.4.1 Conduct Unit Testing	3 days	108 hrs	Thu 10/2/25	Mon 10/6/25
49	3.4.2 Conduct Integration Testing	3 days	108 hrs	Tue 10/7/25	Thu 10/9/25
50	3.4.3 Conduct User Acceptance Testing (UAT)	4 days	144 hrs	Fri 10/10/25	Wed 10/15/25
51	**3.5 Deploy Mobile App**	**5 days**	**144 hrs**	**Thu 10/16/25**	**Wed 10/22/25**
52	3.5.1 Configure Production Environment	3 days	72 hrs	Thu 10/16/25	Mon 10/20/25
53	3.5.2 Publish App on Google Play & Apple Store	2 days	72 hrs	Tue 10/21/25	Wed 10/22/25
54	**3.6 Marketing & User Engagement**	**10 days**	**180 hrs**	**Thu 10/23/25**	**Wed 11/5/25**
55	3.6.1 Create Awareness Campaign	5 days	90 hrs	Thu 10/23/25	Wed 10/29/25
56	3.6.2 Conduct Product Demos & Workshops	5 days	90 hrs	Thu 10/30/25	Wed 11/5/25
57	**3.7 Performance & Security Monitoring**	**15 days**	**660 hrs**	**Thu 11/6/25**	**Wed 11/26/25**
58	3.7.1 Monitor System Performance & Security	15 days	360 hrs	Thu 11/6/25	Wed 11/26/25
59	3.7.2 Conduct Regular Security Audits	15 days	300 hrs	Thu 11/6/25	Wed 11/26/25
60	**3.8 Feedback & Continuous Improvement**	**30 days**	**690 hrs**	**Thu 11/27/25**	**Wed 1/7/26**
61	3.8.1 Gather User Feedback & Analytics	15 days	390 hrs	Thu 11/27/25	Wed 12/17/25
62	3.8.2 Implement Updates & Bug Fixes	15 days	300 hrs	Thu 12/18/25	Wed 1/7/26
63	**4.0 Project Closure**	**24 days**	**1,176 hrs**	**Thu 1/8/26**	**Tue 2/10/26**
64	**4.1 Final Performance Review**	**10 days**	**560 hrs**	**Thu 1/8/26**	**Wed 1/21/26**
65	4.1.1 Evaluate Performance & Engagement	10 days	560 hrs	Thu 1/8/26	Wed 1/21/26
66	**4.2 Closure Report & Future Recommendations**	**7 days**	**322 hrs**	**Thu 1/22/26**	**Fri 1/30/26**
67	4.2.1 Document Outcomes & Recommendations	7 days	322 hrs	Thu 1/22/26	Fri 1/30/26
68	**4.3 Official Project Completion**	**7 days**	**294 hrs**	**Mon 2/2/26**	**Tue 2/10/26**
69	4.3.1 Obtain Final Approval & Close Accounts	7 days	294 hrs	Mon 2/2/26	Tue 2/10/26



Project: Budget with changes
Date: Wed 4/16/25

Task	▬	Project Summary	⊓	Manual Task	▬	Start-only	[	Deadline	⬇
Split	┈	Inactive Task	▭	Duration-only	▬	Finish-only	]	Progress	─
Milestone	◆	Inactive Milestone	◇	Manual Summary Rollup	▬	External Tasks	▬	Manual Progress	─
Summary	⊓	Inactive Summary	⊓	Manual Summary	⊓	External Milestone	◆		

Page 2

Table B: Gantt Chart

Appendix C: RESPONSIBILITY CHART

ID	Task	Project Manager (PM)	Product Owner (PO)	Backend Developer (BD)	Frontend Developer (FD)	UI/UX Designer (UX)	Data Scientist (DS)	Marketing Lead (ML)	Quality Assurance Engineer (QA)	Cloud Engineer (CE)	Security Specialist (SS)	Product Manager (PMgr)	Relationship Manager (RM)	Legal Manager (LM)
	HuskyShare													
1	**1.0 Project Initiation Phase**													
2	1.1 Define Project Scope	A	R				C		I			C	C	I
3	1.2 Identify Key Stakeholders	A	R				R					C	I	C
4	1.3 Conduct Market & Feasibility Study	R	I					C				C	I	I
5	1.4 Define Success Criteria	A	R				R					C	I	I
6	**2.0 Planning Phase**													
7	2.1 Gather User Requirements													
8	2.1.1 Conduct Surveys & User Interviews	R	R				A	I				R	C	I
9	2.1.2 Define Functional & Non-Functional Requirements	A	R	C	C							R	C	
10	2.2 System Design & Architecture													
11	2.2.1 Define Mobile App Architecture	A	C		R	R						C		
12	2.2.2 Plan Backend & Database Structure	A	I	R			R			R		I		
13	2.3 Develop Resource & Budget Plan													
14	2.3.1 Identify Technologies & Resources	A	I	R	R				I			C		
15	2.3.2 Prepare Initial Budget Estimate	R	A					C				R	C	C
16	2.4 Risk Management Plan													
17	2.4.1 Identify Potential Risks	A	R				C					A	R	C
18	2.4.2 Develop Risk Mitigation Strategies	A	R				R	I					I	R
19	**3.0 Development & Execution Phase**													
20	3.1 Backend Development													
21	3.1.1 Develop Authentication & User Management													
22	3.1.1.1 Implement Northeastern SSO Login	A	C	R	I						R	I		
23	3.1.2 Implement Database Management & Storage													

ID	Task	Project Manager (PM)	Product Owner (PO)	Backend Developer (BD)	Frontend Developer (FD)	UI/UX Designer (UX)	Data Scientist (DS)	Marketing Lead (ML)	Quality Assurance Engineer (QA)	Cloud Engineer (CE)	Security Specialist (SS)	Product Manager (PMgr)	Relationship Manager (RM)	Legal Manager (LM)
24	3.1.2.1 Configure MongoDB Atlas	A	C	R	I		R				R	C		
25	3.1.3 Develop API Services & Integrations													
26	3.1.3.1 Build GraphQL API Endpoints	A	C	R	I		I				R	C		
27	3.1.4 Ensure Security & Performance Optimization													
28	3.1.4.1 Implement AES-256 & TLS 1.3 Security	A	C	R	I		R				R	C		
29	3.1.4.2 Optimize API Performance	A	C	R	I		R				R	C		
30	3.2 Frontend Development													
31	3.2.1 Design & Develop User Interface													
32	3.2.1.1 Create Prototype in Figma	A	C	I	R	R	I					C		
33	3.2.1.2 Build React Native Interface	A	C	I	R	R	I					C		
34	3.2.2 Implement Core Application Features													
35	3.2.2.1 Develop Listing & Browsing Modules	A	C	I	R	R	I					C		
36	3.2.3 Enhance Search, Navigation & Interaction													
37	3.2.3.1 Add Search & Filter Features	A	C	I	R	R	I					C		
38	3.2.4 Integrate Communication & Notifications													
39	3.2.4.1 Build In-App Messaging	A	C	R	R	R	I					C		
40	3.2.4.2 Integrate Firebase Notifications	A	C	R	R	R	I					C		
41	3.3 Cloud & Infrastructure Setup													
42	3.3.1 Configure Cloud Hosting & Deployment													
43	3.3.1.1 Set Up AWS Lambda & Environment	A	C	R	I		R			R		C		
44	3.3.2 Implement System Monitoring & Alerts													
45	3.3.2.1 Deploy AWS Auto-Scaling & Alerts	A	C	R	I		R			R		C		
46	3.4 Testing & Debugging													
47	3.4.1 Conduct Unit Testing	I	I	R	R	I	I		A	I		C		
48	3.4.2 Conduct Integration Testing	I	I	R	R	I	I		A	I		C		

ID	Task	Project Manager (PM)	Product Owner (PO)	Backend Developer (BD)	Frontend Developer (FD)	UI/UX Designer (UX)	Data Scientist (DS)	Marketing Lead (ML)	Quality Assurance Engineer (QA)	Cloud Engineer (CE)	Security Specialist (SS)	Product Manager (PMgr)	Relationship Manager (RM)	Legal Manager (LM)
49	3.4.3 Conduct User Acceptance Testing (UAT)	I	I	R	R	I	I		A	I		C		
50	3.5 Deploy Mobile App													
51	3.5.1 Configure Production Environment	A		C	R		I					C		
52	3.5.2 Publish App on Google Play & Apple Store	A	C	R	R	I		I					C	I
53	3.6 Marketing & User Engagement													
54	3.6.1 Create Awareness Campaign	A						R					C	
55	3.6.2 Conduct Product Demos & Workshops	A						R					C	
56	3.7 Performance & Security Monitoring													
57	3.7.1 Monitor System Performance & Security	C					C		C	C	R			
58	3.7.2 Conduct Regular Security Audits	C							C		R			C
59	3.8 Feedback & Continuous Improvement													
60	3.8.1 Gather User Feedback & Analytics	A	C			C	R	C						
61	3.8.2 Implement Updates & Bug Fixes	C		C	R				C					
62	**4.0 Project Closure**													
63	4.1 Final Performance Review													
64	4.1.1 Evaluate Performance & Engagement	R	A	C	C	C	C	I	R	C	C	I	C	I
65	4.2 Closure Report & Future Recommendations													
66	4.2.1 Document Outcomes & Recommendations	R	R, A	I	I	I	I	I	R	I	I	I	I	C
67	4.3 Official Project Completion													
68	4.3.1 Obtain Final Approval & Close Accounts	R	R	I	I	I	I	I	I	I	I	I	I	A

Table C: RACI Matrix

Appendix D: RESOURCE ALLOCATION

Resource	Type	Company	Role	Effort in Hours	Rate/Hour ($)	% Allocation
Chilkuri, Krithik Reddy	Labor	NEU	Project Manager	1182	17.90[2]	81.18%
Bijarniya, Mayur Mahavir	Labor	NEU	Product Owner	718	25[3]	49.31%
Maram, Sreya Reddy	Labor	NEU	Backend Developer	688	63[4]	47.25%
N, Dheeraj Reddy	Labor	NEU	Frontend Developer	670	25[5]	46.02%
Ramasubramanian, Jaanaki Raman	Labor	NEU	UI/UX Designer	416	26.71[6]	28.57%
Telakala, Shravya Sree Reddy	Labor	NEU	Product Manager	508	23[7]	34.89%
Data Scientist	Labor	NEU	Data Scientist	670	24[8]	46.02%
Marketing Lead	Labor	NEU	Marketing Lead	234	19[9]	16.07%
Quality Assurance Team	Labor	NEU	Quality Assurance	398	28[10]	27.34%
Cloud Engineer	Labor	NEU	Cloud Engineer	252	21[11]	17.31%
Security Specialist	Labor	NEU	Security Specialist	508	22[12]	34.89%
Legal Manager	Labor	NEU	Legal Manager	234	21[13]	16.07%
Relationship Manager	Labor	NEU	Relationship Manager	204	19[14]	14.01%
Total Hours			6,546			
Contingency			8%			

Table D: Resource Allocation

Appendix E: BUDGET JUSTIFICATION

Project Name: HuskyShare **Estimated Start time: June 2nd, 2025**						**Total Budget**		**$249,076.27**
Labor Cost								
ID	**Task**	**Duration (Days)**	**No. of people involved**	**Working Hours**	**Net. Working Hours**	**Cumulative Rate**	**Hourly Rate**	**Task Total Cost**
1	**1.0 Project Initiation Phase**							
2	1.1 Define Project Scope	2	7	16	60	$83.68	$11.95	$1,338.80
3	1.2 Identify Key Stakeholders	1	6	8	32	$89.18	$14.86	$713.40
4	1.3 Conduct Market & Feasibility Study	3	6	24	66	$55.15	$9.19	$1,323.60
5	1.4 Define Success Criteria	2	6	16	60	$83.93	$13.99	$1,342.80
							Phase Total	**$4,718.60**
6	**2.0 Planning Phase**							
7	2.1 Gather User Requirements							
8	2.1.1 Conduct Surveys & User Interviews	5	7	40	190	$103.40	$14.77	$4,136.00
9	2.1.2 Define Functional & Non-Functional Requirements	3	6	24	102	$114.93	$19.15	$2,758.20
10	2.2 System Design & Architecture							
11	2.2.1 Define Mobile App Architecture	3	5	24	90	$89.14	$17.83	$2,139.24
12	2.2.2 Plan Backend & Database Structure	3	6	24	102	$133.43	$22.24	$3,202.20
13	2.3 Develop Resource & Budget Plan							
14	2.3.1 Identify Technologies & Resources	2	6	16	60	$126.18	$21.03	$2,018.80
15	2.3.2 Prepare Initial Budget Estimate	3	6	24	102	$89.15	$14.86	$2,139.60
16	2.4 Risk Management Plan							
17	2.4.1 Identify Potential Risks	2	6	16	72	$97.18	$16.20	$1,554.80

ID	Task	Duration (Days)	No. of people involved	Working Hours	Net. Working Hours	Cumulative Rate	Hourly Rate	Task Total Cost
18	2.4.2 Develop Risk Mitigation Strategies	4	6	32	120	$92.93	$15.49	$2,973.60
							Phase Total	**$20,922.44**
19	**3.0 Development & Execution Phase**							
20	3.1 Backend Development							
21	3.1.1 Develop Authentication & User Management							
22	3.1.1.1 Implement Northeastern SSO Login	5	6	40	150	$122.93	$20.49	$4,917.00
23	3.1.2 Implement Database Management & Storage							
24	3.1.2.1 Configure MongoDB Atlas	5	7	40	200	$152.68	$21.81	$6,107.00
25	3.1.3 Develop API Services & Integrations							
26	3.1.3.1 Build GraphQL API Endpoints	5	7	40	170	$134.68	$19.24	$5,387.00
27	3.1.4 Ensure Security & Performance Optimization							
28	3.1.4.1 Implement AES-256 & TLS 1.3 Security	5	7	40	200	$152.68	$21.81	$6,107.00
29	3.1.4.2 Optimize API Performance	5	7	40	200	$152.68	$21.81	$6,107.00
30	3.2 Frontend Development							
31	3.2.1 Design & Develop User Interface							
32	3.2.1.1 Create Prototype in Figma	5	7	40	170	$110.89	$15.84	$4,435.40
33	3.2.1.2 Build React Native Interface	5	7	40	170	$110.89	$15.84	$4,435.40
34	3.2.2 Implement Core Application Features							
35	3.2.2.1 Develop Listing & Browsing Modules	5	7	40	170	$110.89	$15.84	$4,435.40
36	3.2.3 Enhance Search, Navigation & Interaction							
37	3.2.3.1 Add Search & Filter Features	5	7	40	170	$110.89	$15.84	$4,435.40
38	3.2.4 Integrate Communication & Notifications							
39	3.2.4.1 Build In-App Messaging	5	7	40	200	$158.14	$22.59	$6,325.40

ID	Task	Duration (Days)	No. of people involved	Working Hours	Net. Working Hours	Cumulative Rate	Hourly Rate	Estimated Cost
40	3.2.4.2 Integrate Firebase Notifications	5	7	40	200	$158.14	$22.59	$6,325.40
41	3.3 Cloud & Infrastructure Setup							
42	3.3.1 Configure Cloud Hosting & Deployment							
43	3.3.1.1 Set Up AWS Lambda & Environment	5	7	40	200	$151.68	$21.67	$6,067.00
44	3.3.2 Implement System Monitoring & Alerts							
45	3.3.2.1 Deploy AWS Auto-Scaling & Alerts	5	7	40	200	$151.68	$21.67	$6,067.00
46	3.4 Testing & Debugging							
47	3.4.1 Conduct Unit Testing	3	9	24	108	$149.15	$16.57	$3,579.66
48	3.4.2 Conduct Integration Testing	3	9	24	108	$149.15	$16.57	$3,579.66
49	3.4.3 Conduct User Acceptance Testing (UAT)	4	9	32	144	$149.15	$16.57	$4,772.88
50	3.5 Deploy Mobile App							
51	3.5.1 Configure Production Environment	3	5	24	72	$87.43	$17.49	$2,098.20
52	3.5.2 Publish App on Google Play & Apple Store	2	8	16	72	$140.10	$17.51	$2,241.64
53	3.6 Marketing & User Engagement							
54	3.6.1 Create Awareness Campaign	5	3	40	90	$41.93	$13.98	$1,677.00
55	3.6.2 Conduct Product Demos & Workshops	5	3	40	90	$41.93	$13.98	$1,677.00
56	3.7 Performance & Security Monitoring							
57	3.7.1 Monitor System Performance & Security	15	5	120	300	$67.45	$13.49	$8,094.00
58	3.7.2 Conduct Regular Security Audits	15	4	120	240	$55.45	$13.86	$6,654.00
59	3.8 Feedback & Continuous Improvement							
60	3.8.1 Gather User Feedback & Analytics	15	5	120	390	$72.78	$14.56	$8,733.60
61	3.8.2 Implement Updates & Bug Fixes	15	4	120	300	$79.45	$19.86	$9,534.00
							Phase Total	**$123,793.04**

ID	Task	Duration (Days)	No. of people involved	Working Hours	Net. Working Hours	Cumulative Rate	Hourly Rate	Estimated Cost
62	**4.0 Project Closure**							
63	4.1 Final Performance Review							
64	4.1.1 Evaluate Performance & Engagement	10	13	80	560	$180.76	$13.90	$14,460.40
65	4.2 Closure Report & Future Recommendations							
66	4.2.1 Document Outcomes & Recommendations	7	13	56	322	$142.08	$10.93	$7,956.34
67	4.3 Official Project Completion							
68	4.3.1 Obtain Final Approval & Close Accounts	7	13	56	294	$152.68	$22.59	$7,074.34
							Phase Total	**$29,491.08**
							Total Labor Cost	**$178,925.16**

Materials Cost

ID	Materials	Quantity	Unit Price	Estimated Cost
69	**1.0 Hardware**			
70	1.1 Development Laptops	6	$1,500	$9,000
71	1.2 Testing Devices	1	$2,000	$2,000
72	**2.0 Cloud Infrastructure**			
73	2.1 MongoDB Atlas (M20 cluster)	1	$7,800.00	$7,800
74	2.2 AWS Lambda	1	$4,200.00	$4,200
75	2.3 AWS S3	1	$3,600.00	$3,600
76	2.4 AWS CloudFront	1	$1,200.00	$1,200
77	2.5 Firebase Cloud Messaging	1	$1,200.00	$1,200
78	2.6 AWS Auto-Scaling & Load Balancer	1	$3,600.00	$3,600

ID	Materials	Quantity	Unit Price	Estimated Cost
79	**3.0 Development Tools**			
80	3.1 Figma (UI/UX Design)	1	$180	$180
81	3.2 GraphQL Tools/Extensions	1	$200	$200
82	3.3 Development IDEs	1	$400	$400
83	3.4 Testing Frameworks	1	$250	$250
84	**4.0 Security**			
85	4.1 SSL Certificates	1	$300	$300
86	4.2 Security Testing Tools	1	$2,000	$2,000
87	4.3 Penetration Testing Services	1	$5,000	$5,000
88	**5.0 Deployment & Distribution**			
89	5.1 Apple Developer Account	1	$99	$99
90	5.2 Google Play Developer Account	1	$25	$25
91	5.3 CI/CD Pipeline Tools	1	$900	$900
92	**6.0 Monitoring & Support**			
93	6.1 Performance Monitoring	1	$2,400	$2,400
94	6.2 Error Tracking Software	1	$1,200	$1,200
95	6.3 Analytics Platform	1	$1,800	$1,800
96	6.4 Log Management System	1	$1,200	$1,200
			Total Material Cost	**$48,554**
			Contingency to Material Cost (15%)	**$7,283**
			Total Labor Cost	**$178,925.16**
			Contingency to Labor Cost (8%)	**$14,314.01**
			Overall Budget Estimate	**$249,076**

Table E: Budget Justification

REFERNCES

1. DailyTrojan. It’s high time that we emphasize more sustainable college move-ins [2025 Feb 6]. https://dailytrojan.com/2023/08/31/move-in-out-waste/
2. ZipRecruiter. *Project manager intern salary in Massachusetts*. https://www.ziprecruiter.com/Salaries/Project-Manager-Intern-Salary--in-Massachusetts#:~:text=How%20much%20does%20a%20Project,Massachusetts%20is%20%2417.90%20an%20hour
3. ZipRecruiter. *Intern product owner salary*. https://www.ziprecruiter.com/Salaries/Intern-Product-Owner-Salary#:~:text=How%20much%20does%20an%20Intern,States%20is%20%2422.87%20an%20hour
4. ZipRecruiter. *Backend developer intern salary*. https://www.ziprecruiter.com/Salaries/Backend-Developer-Intern-Salary
5. ZipRecruiter. *Intern front-end web developer salary*. https://www.ziprecruiter.com/Salaries/Intern-Front-End-Web-Developer-Salary
6. Indeed. *User experience design intern salaries*. https://www.indeed.com/career/user-experience-design-intern/salaries
7. ZipRecruiter. *Product management intern salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Product-Management-Intern-Salary-in-Boston%2CMA
8. ZipRecruiter. *Data science intern salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Data-Science-Intern-Salary-in-Boston,MA
9. ZipRecruiter. *Marketing intern salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Marketing-Intern-Salary-in-Boston,MA
10. ZipRecruiter. *QA internship salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Qa-Internship-Salary-in-Boston,MA
11. ZipRecruiter. *Intern cloud engineer salary*. https://www.ziprecruiter.com/Salaries/Intern-Cloud-Engineer-Salary
12. ZipRecruiter. *Cyber security intern salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Cyber-Security-Intern-Salary-in-Boston,MA#:~:text=How%20much%20does%20a%20Cyber,Boston%20is%20%2422.16%20an%20hour
13. ZipRecruiter. *Legal intern salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Legal-Intern-Salary-in-Boston,MA
14. ZipRecruiter. *Public relations intern salary in Boston, MA*. https://www.ziprecruiter.com/Salaries/Public-Relations-Intern-Salary-in-Boston,MA