



United International University
Department of CSE
SRS Report

Course Code: CSE 3412

Course Name: System Analysis and Design Laboratory

Sec: D

Group No: 5

Group name: Game of Threads

Members:

ID	Name
0112230295	Tabassum Sumaiya
0112230836	Jarin Tabassum
0112230293	Md. Hasibul Hossain
0112230043	Abu Hurayra Mahbe
0112230028	Ashraful Islam Tanzil

Earn & Learn

Campus Marketplace platform



Table of Contents

1.2 Intended Audience	5
3.2 Gap Analysis.....	14
3.4 Feature List Fixation	15
4.1 User Registration & Profile Management.....	18
4.2 Job and Gig Posting & Application System.....	19
4.3 Material Exchange System.....	20
4.4 Skill Sharing System.....	21
4.5 Escrow Payment System.....	22
4.6 Real-Time Messaging & Notification System	23
4.7 Campus Social Network.....	24
4.8 Project & Task Management.....	25
4.9 Financial & Savings Management	25
5. External & Non-Functional Requirements.....	27
5.1 User Interfaces	27
5.2 Software Interfaces	31
5.3 Performance Requirements	31
5.4 Security Requirements	33
5.5 Maintainability	33
6. Feasibility & Project Management Analysis	35
6.1 Technical Feasibility	35
6.2 Economic / Financial Feasibility	36
6.4 Human Resources Feasibility.....	38
6.5 Market, Legal, and Political Feasibility	38
6.6 Schedule Feasibility	39
6.7 SWOT Analysis	41
7.1 Data Flow Diagram (DFD)	43
7.2 Use Case Diagram.....	45
7.3 Class Diagram	47
Key Classes and Their Roles	47
7.4 Sequence Diagram	49
Purpose.....	49
Key Components.....	49
8. Prototype Design	51
8.1 Overview.....	51
8.2.12 Earn and Learn (Project Management).....	57
	57

.....	57
8.3 UI Design Principles – Guidelines Followed.....	58
9.1 Future Enhancements.....	59
10. References	59
11. Appendices	61
The appendices provide supplementary information that supports the main content of this SRS. They include definitions of key terms, acronyms, and abbreviations, as well as references to tools, software, and diagrams used throughout the project. This section helps readers and developers understand specialized terminology, design references, and resources, ensuring clarity and consistency in system development and documentation.....	61
Appendix A – Glossary.....	61
Appendix B – Tools & Software Used	62
Appendix C – References to Diagrams.....	62

1. Introduction

1.1 Purpose

The Earn-N-Learn platform is designed to tackle the scattered nature of student opportunities and services on university campuses. Right now, students depend on multiple platforms like Facebook groups, OLX, LinkedIn, or physical bulletin boards to find part-time jobs, share academic materials, and seek peer assistance. This scatter creates inefficiency, distraction, and a lack of security.

Our project aims to develop a single, campus-focused digital ecosystem that brings these activities together into one secure platform. The system will enable students to earn money from their skills, find verified jobs and gigs, share materials safely, work together on projects, and manage payments openly.

1.2 Intended Audience

- **Primary Users:** University and college students looking for on-campus jobs, freelance work, peer tutoring, and material exchanges.
- **Secondary Users:**
 - University career centers for approved job postings.
 - Freelancers and service seekers in the campus community.
 - Student organizations and clubs that manage campus events.
- **Stakeholders:**
 - University administration including security, verification, and trust.
 - Employers and recruiters targeting students.

1.3 Scope of the System

The Earn-N-Learn system will provide:

- **Job Marketplace** – A safe place to post and apply for on-campus, remote, part-time roles with application tracking.
- **Skill Sharing Marketplace** – Peer-to-peer tutoring and freelance service exchange with pricing and availability.
- **Material Exchange** – A structured system for buying, selling or renting textbooks and academic resources.
- **Community Hub** – Spaces for campus discussions, event announcements, and group communities.
- **Secure Wallet System** – A way to handle payments through Escrow, track transaction history, withdraw instantly, and set savings goals.

- **Collaboration Tools** – Features for project management, task tracking, and real-time messaging.

The goal is to improve student productivity, ensure safe transactions, and create a verified and unified environment for campus life and financial growth and independence.

2. System Overview

2.1 Product Perspective

The system will be a standalone web and mobile platform designed specifically for university students, featuring secure payments through services like SSLCommerz and Stripe. Unlike general platforms like Upwork, Fiverr, and OLX, Earn-N-Learn focuses on campuses and meets students' needs.

It will operate as a self-contained system but may connect with:

- University verification systems for student ID validation.
- Payment gateways for deposits and withdrawals.
- Cloud storage for media uploads, including materials and portfolios.

2.2 Product Functions (High-Level Features)

Here is a summary of the Earn & Learn's main features which can provide a brief guideline of what the system can offer:

1. **Job & Gig Listings** – Peer-to-peer and institutional postings.
2. **Tutoring & Skill-Sharing** – Connect learners with peer or professional tutors.
3. **Student Marketplace** – Buy/sell textbooks, materials, and personal items safely.
4. **In-App Chat** – Secure, moderated communication between users.
5. **Digital Wallet** – Safe transactions for tutoring, gigs, and marketplace.
6. **Profile & Reputation System** – Trust-building via reviews, ratings, and verified profiles.
7. **Campus Event Announcements** – Clubs and organizations post updates.
8. **Discussion Forums** – Themed channels for collaboration and Q&A.

2.3 User Classes and Characteristics

- **Students (Primary Users):**
 - Looking for jobs, side hustles, peer tutoring, and material exchange.
 - Age range: 18–28 years.
 - Tech-savvy, accustomed to using social media and digital tools.
- **Recruiters/Employers (Secondary Users):**

- University faculty, student clubs, and external recruiters.
- Require easy job posting and application review tools.
- **Admins:**
 - Responsible for system moderation, security, and dispute resolution.

2.4 Operating Environment

- **Frontend:** React 18, TypeScript, Tailwind CSS.
- **Backend:** Node.js, Express, MySQL.
- **Real-time Communication:** Socket.IO.
- **Platforms Supported:**
 - Mobile App (Android/iOS).
 - Web Application (cross-browser).
- **Network Requirements:** Stable internet connection (minimum 2 Mbps).

2.5 Design and Implementation Constraints

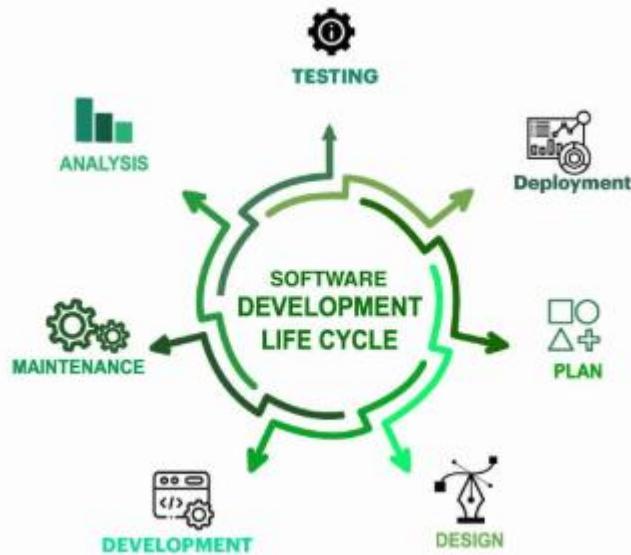
- Limited to university-verified users (authentication via student email/ID).
- Budget and timeline restrictions (student-led project).
- Must comply with data privacy and security policies.
- Payment features depend on third-party providers.
- Limited development timeframe within a semester.

2.6 Assumptions and Dependencies

- Students have regular internet access and smartphones.
- Universities permit student-driven platforms for communication and commerce.
- Payment gateway APIs will remain stable and available.
- Success depends on adoption by at least 50% of the surveyed student body.
- Sufficient server resources will be provided for peak usage (e.g., semester start, exam season).

2.7 SDLC Approach

To make sure Earn & Learn delivers on its promise, we're following a structured plan called the Software Development Life Cycle. Here's how it works:



1. **Requirement Gathering:** Collected inputs through student surveys, interviews, and competitor analysis to identify essential features like jobs, skill-sharing, and material exchange.
2. **System Design:** Designed ERD, architecture, and UI mockups with a simple, student-friendly dashboard for learning and earning.
3. **Analysis:** Performed feature comparison and gap analysis to finalize a student-focused feature list ensuring affordability, trust, and easy access.
4. **Development:** Implemented frontend (React.js), backend (Node.js/Express), and database (MongoDB/PostgreSQL) for smooth and secure operations.
5. **Testing:** Conducted unit, integration, usability, and security testing to guarantee reliability and safety for student users.
6. **Deployment:** Deployed on cloud hosting with mobile-friendly access for flexible campus usage.
7. **Maintenance:** Regular updates, bug fixes, and enhancements to keep improving features and student experience

3. Requirements Elicitation & Analysis

3.1 Benchmark Analysis:

The benchmark for Earn & Learn outlines a comprehensive comparison of features, functionalities, and performance metrics against similar platforms. This benchmark serves as a guide to ensure Earn & Learn meets and exceeds industry standards while providing a user-focused, engaging, and innovative platform for personal growth and well-being. Here is our benchmark:

Feature Area	Upwork	Fiverr	Freelancer	Handshake	Braintrust	Earn-N-Learn
Job/Internship Listings	<input checked="" type="checkbox"/> Advanced (bidding, contracts)	<input checked="" type="checkbox"/> Gig-based	<input checked="" type="checkbox"/> Bidding, contests	<input checked="" type="checkbox"/> Curated by employers	<input checked="" type="checkbox"/> Curated freelance roles	<input checked="" type="checkbox"/> Yes (with detailed info, filters)
Skill Sharing / Service Offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Professionals offer services	<input checked="" type="checkbox"/> Peer-to-peer services (e.g., tutoring)
Material Exchange (Textbooks, Tools)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Buy/sell/rent items
Project Tracking with Milestones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> (single delivery)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> With client updates	<input checked="" type="checkbox"/> Milestone-based progress tracking
Escrow Payment System	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Smart contract-based escrow	<input checked="" type="checkbox"/> Built-in, secure wallet
Group Chats for Projects	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Integrated collaboration	<input checked="" type="checkbox"/> Real-time team chats
Campus Community Posts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Career-focused updates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Campus-specific forums, events
Earnings/Savings Goals	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Financial tracking & goals

 Smart Job Recommendations	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> (search-based)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> AI-driven matching	<input checked="" type="checkbox"/> Based on user profile/interests
---	-------------------------------------	--	-------------------------------------	-------------------------------------	--	---

From the benchmark analysis table, it's clear that existing platforms like Upwork, Fiverr, and Freelancer are mainly designed for global freelancing and gig work. They offer features such as bidding systems, escrow payments, and project tracking, but they don't have campus-specific functions like academic material exchange, peer tutoring, or localized student networking.

In contrast, Handshake concentrates on curated internships and jobs, linking students with employers. However, it is limited to career opportunities and does not offer peer-to-peer services or financial tools. Braintrust has smart contracts and built-in collaboration tools, but it is focused on professionals instead of students.

Earn-N-Learn, on the other hand, fills this gap by combining the best aspects of freelancing platforms with features aimed at students. It offers a peer-to-peer skill-sharing system, academic material exchange, a secure in-app wallet, group chats, campus forums, and personalized financial tracking. This makes it a one-stop ecosystem designed for students' academic, professional, and social needs.

3.2 Requirements Analysis

Purpose: To identify what students need most in a campus marketplace.

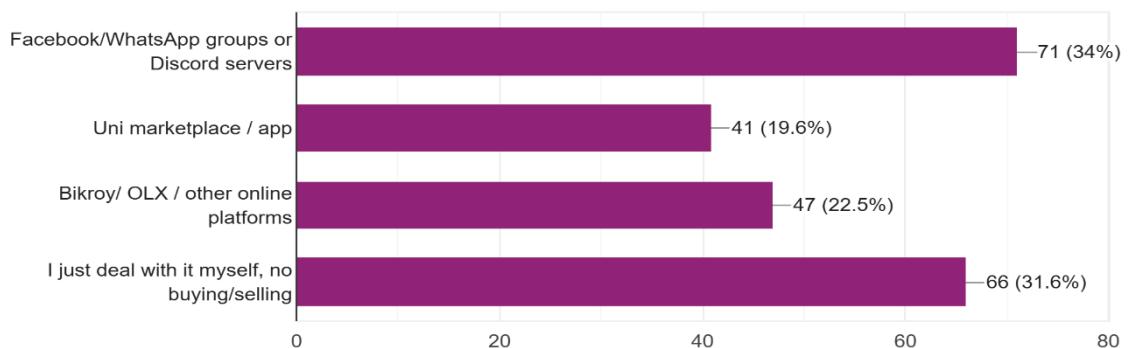
Scope: Jobs, gigs, tutoring, material exchange, payments, and community engagement.

- **Information Collected:**
 - Student habits (jobs, material exchange, peer help).
 - Processes (job applications, skill-sharing, transactions).
 - Campus social interactions.
- **Sources of Information:**
 - Surveys (Google Forms, **209 respondents**)
 - Documents (existing university apps, portals)
 - Observations (daily communication/workflows)
 - Informal discussions with peers
 - Online research on competing platforms

Among all these the majority voting from the Google Forms survey gave us direct, data-driven evidence of student frustration with current methods and a strong demand for a centralized platform. This not only validated the idea but also inspired and motivated us to move forward with building a system that addresses these gaps. Here are some of the responses that we collected through the Google Forms:

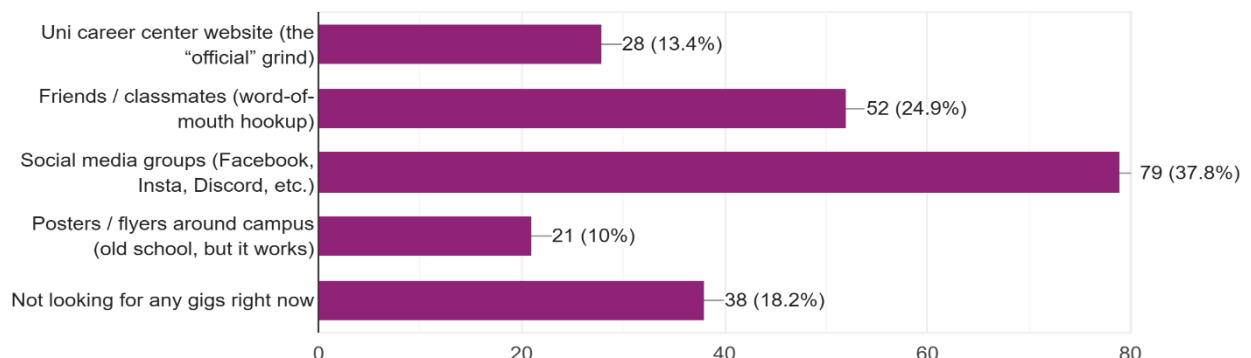
How do you usually buy or sell used textbooks & academic stuff?

209 responses



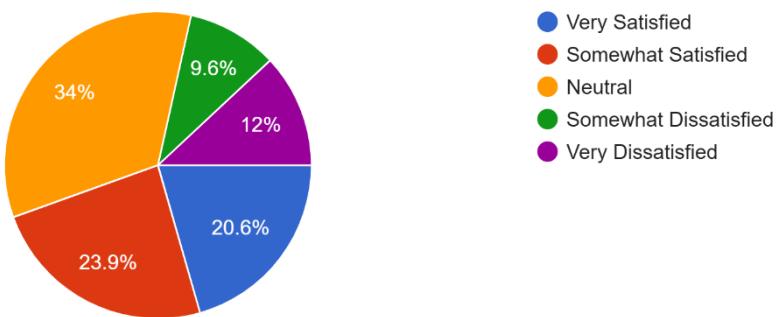
How do you usually hunt for part-time or on-campus side hustle?

209 responses



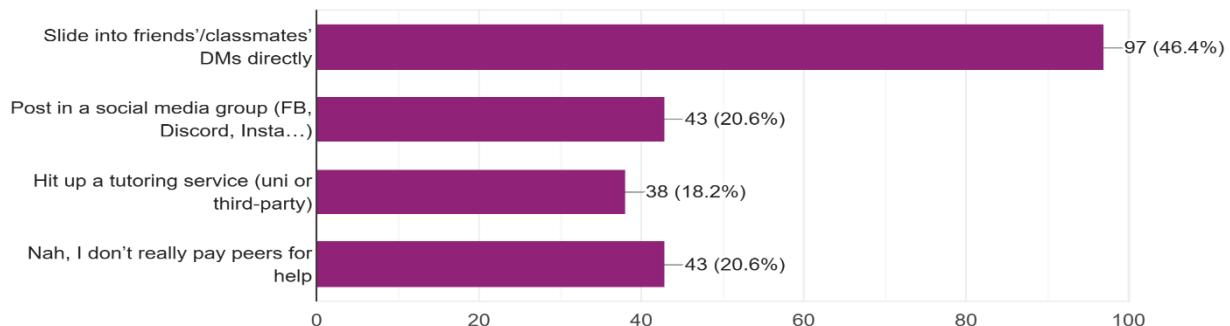
How satisfied are you with your current methods for the activities above?

209 responses



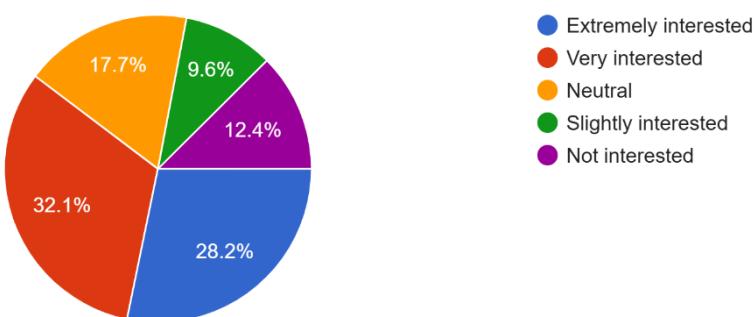
How do you usually find a friend to help you?

209 responses



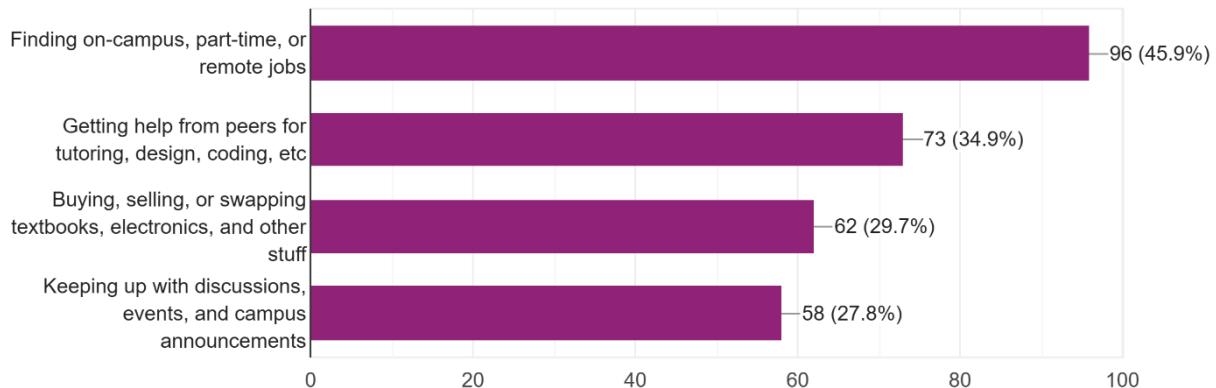
How interested would you be in using a secure, all-in-one campus app for buying/selling, jobs, and skill-sharing?

209 responses



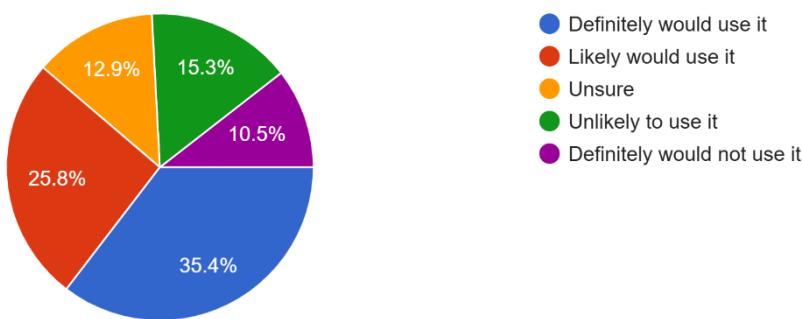
Imagine you had an all-in-one campus app that could help you out with campus life. Which of these would make your life easier? Pick your top 2

209 responses



If this app were available, how likely would you be to use it?

209 responses



The Google Forms survey played a crucial role in motivating us to create Earn-N-Learn. By collecting responses from **209 participants (90% students, gender balanced)**, we could clearly see majority voting trends in their needs and preferences.

- A large portion of students indicated that they rely on fragmented sources (Facebook, LinkedIn, university bulletins, etc.) for jobs, gigs, and events — showing the need for an all-in-one platform.
- Job opportunities and peer-to-peer help consistently ranked as top priorities, motivating us to design core features around these.
- Over 60% expressed high interest in using an all-in-one campus app, confirming feasibility and potential adoption.
- Students also voted strongly in favor of a secure in-app wallet, which encouraged us to include financial tracking and safe transactions.

3.2 Gap Analysis

Gaps in Current Solutions (What Others Provide, We Identified as Missing Needs)

1. **No Unified Platform for Students**
 - Existing solutions like Upwork, Fiverr, and LinkedIn only focus on freelancing or professional networking, but none consolidate jobs, gigs, academic exchange, and campus life into a single app.
2. **Lack of Peer-to-Peer Material Exchange**
 - Competitors such as Handshake and Upwork do not support second-hand academic material buying/selling, a common student need revealed in our survey.
3. **Limited Campus-Oriented Features**
 - Platforms like Freelancer or Fiverr serve global markets, but they lack localized campus-specific posts, forums, and event sharing.
4. **Absence of Low-Cost, Student-Friendly Services**
 - Existing freelancing sites are high-budget or professional-focused. They fail to provide affordable, small-scale peer help (like tutoring, assignment guidance, or project collaboration).
5. **Minimal Financial Tools for Students**
 - Competitors may offer escrow (Upwork, Fiverr) or crypto wallets (Braintrust), but none focus on financial literacy for students, such as savings goals and easy micro-transactions.
6. **No Safe, Controlled Communication System**
 - Most platforms rely on open messaging or external contact (LinkedIn, Facebook groups), which raises privacy risks. In-app, secure chat tailored for campus interactions is missing.
7. **Fragmented Payment Integration**
 - Freelancing platforms support international payments but not local, student-accessible methods like mobile banking or instant wallet transfers.
8. **Lack of Small-Scale Project Collaboration**
 - Competitors offer either large, contractual project management (Upwork, Braintrust) or one-off gigs (Fiverr), but not milestone-based, small collaborative tasks among peers.

Targeted Gap Analysis (What Earn-N-Learn Offers That Others Lack):

1. **All-in-One Campus Ecosystem**
 - Unlike scattered platforms, Earn-N-Learn integrates jobs, gigs, skill-sharing, material exchange, events, and payments into a single student-centered solution.
2. **Peer-to-Peer Material Marketplace**
 - Unique buy/sell/rent model for textbooks, electronics, and academic tools — absent in other gig or career platforms.

3. **Affordable, Micro-Gigs for Students**
 - Provides small, cost-effective tasks (tutoring, design, coding help) suitable for students' budgets, unlike professional-only platforms.
4. **Campus-Specific Forums & Event Spaces**
 - Builds community engagement through localized posts, event sharing, and discussions relevant to student life.
5. **Integrated Financial Literacy Tools**
 - In-app wallet not only secures transactions but also helps track savings and earnings, addressing financial management challenges unique to students.
6. **Privacy-Centric Communication**
 - Real-time in-app chat ensures safe, moderated interaction compared to risky direct contact on social platforms.
7. **Student-Centric Payment Options**
 - Supports multiple methods (bank, cards, wallets, mobile payments), tailored for accessibility in student communities.
8. **Milestone-Based Collaboration for Students**
 - Enables tracking and accountability for group projects and gigs — a feature usually restricted to enterprise-level platforms.
9. **Adoption Feasibility Validated by Students**
 - Backed by survey data showing 60%+ strong interest in an all-in-one app, making it both needed and widely adoptable.
10. **Holistic Value Addition to Campus Life**
 - Goes beyond just jobs or freelancing — addressing day-to-day campus needs from earning to learning, saving, and connecting.

3.4 Feature List Fixation

Based on survey findings, benchmark analysis, and feasibility evaluation, the following core features have been finalized for the Earn-N-Learn system. These features directly address the identified gaps and provide unique value tailored for students:

1. **Secure User Registration & Enhanced Profile Creation**
 - University email ID-based registration with verification.
 - Rich profile creation including skills, certifications, portfolio, and endorsements.
 - Public profile pages link directly to job applications, marketplace listings, and peer services.
2. **Job & Gig Posting with Application Tracking**
 - Students and recruiters can post part-time jobs, freelancing gigs, and campus opportunities.
 - Applicants can browse, filter, and apply directly through the platform.

- Posters have an application tracking dashboard to review, shortlist, interview, hire, or reject candidates.

3. Material Exchange Listing & Secure Transaction

- Marketplace for buying, selling, or renting textbooks, equipment, and academic tools.
- Listing includes images, pricing, condition, and delivery method.
- Escrow-based transaction ensures trust and transparency between buyer and seller.

4. Peer-to-Peer Skill Sharing

- Students can offer or request tutoring, project support, or creative/technical services.
- Listings include availability, pricing (free/paid), and ratings.
- Booking system with confirmation, calendar integration, and transaction history.

5. Escrow-Secured Campus Transaction System

- Built-in wallet with escrow holding ensures safe payments for jobs, gigs, and exchanges.
- Payment only released after service completion and buyer approval.
- Dispute resolution integrated into the system.

6. Real-time Direct & Group Messaging

- Instant text, file sharing, and multimedia messaging using socket-based communication.
- Group chat for project teams, clubs, or course groups.
- Optional voice/video call support for tutoring and interviews.

7. Campus Social Network

- Topic-based groups (e.g., clubs, academic fields, textbook exchanges).
- Personalized campus feed with posts, likes, comments, and discussions.
- Strengthens student community engagement beyond just transactions.

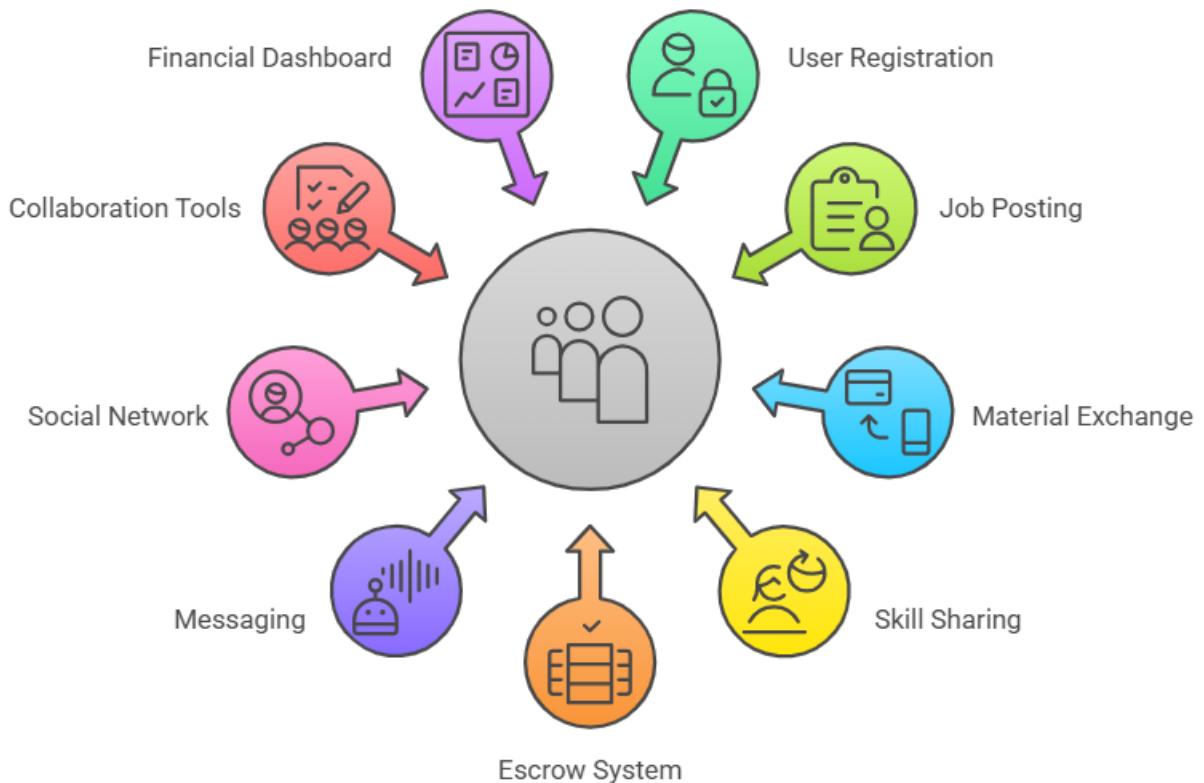
8. Project Collaboration & Task Management

- Dedicated project workspace with task boards, milestones, and deadlines.
- Resource sharing (documents, links, files) and integrated project chat.
- Progress tracking visible to all collaborators, linked to user profiles for portfolio credibility.

9. Financial Savings & Performance Dashboard

- Students can set savings goals (e.g., laptop, tuition) and allocate earnings toward them.
- Dashboard with real-time tracking of wallet balance, transaction history, and goal progress.
- Promotes financial literacy among students, a unique differentiator from competitors.

Earn and Learn Feature List Fixation



These nine features represent the core scope of Earn-N-Learn. Each was validated by student survey responses, competitor gap analysis, and feasibility studies. Together, they create a secure, student-centric, all-in-one platform that combines earning, learning, collaboration, and financial empowerment.

4. Functional Requirements

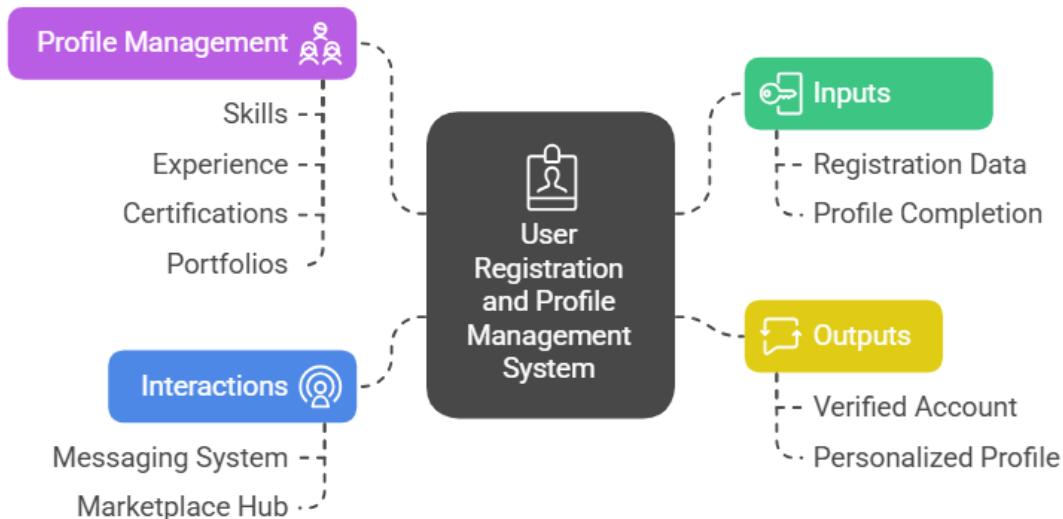
This section outlines the core functional requirements of our system Earn and Learn defining each key system feature, its purpose, inputs, outputs, and system interactions.

4.1 User Registration & Profile Management

Detailed Description:

Our system allows new users to register using their University ID and institutional email to ensure authenticity. Once registered, users can create a detailed profile that highlights their skills, experience, certifications, and portfolios. The profile serves as a central identity hub across all modules of the platform.

User Registration and Profile Management System



Inputs:

- Registration Data: Full Name, University ID, Email, Course Area, Password.
- Profile Completion: Profile Picture, Skill Tags, Experience Description, Portfolio Files/Links, and Social/Website URLs.

Outputs:

- System: Verified user account with a secure authentication token.
- User: A personalized public profile page displaying all credentials and achievements.

Interactions:

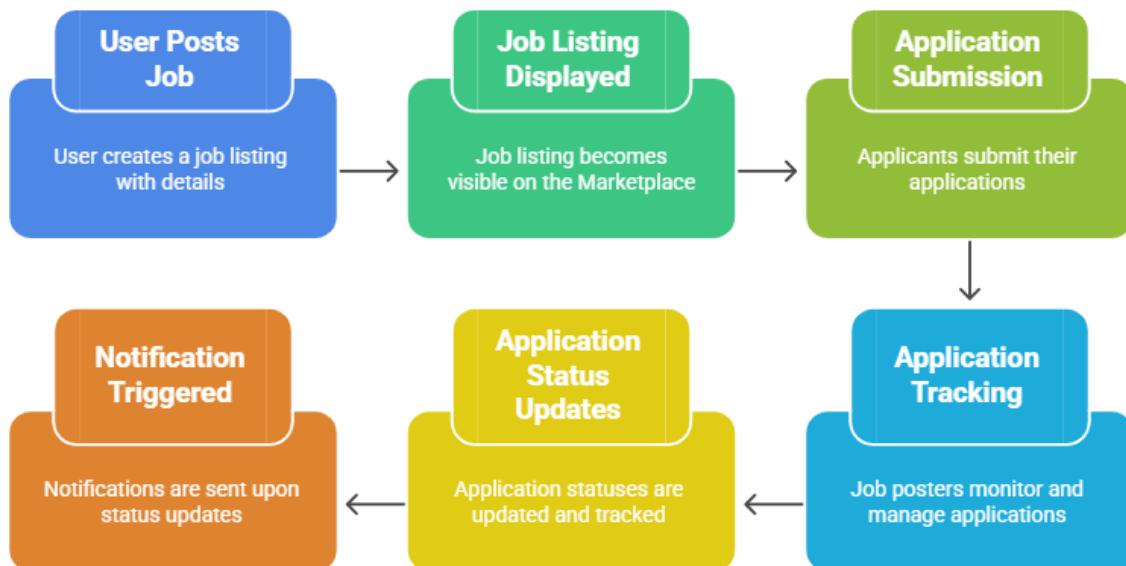
- Integrates with the Messaging System for profile linking.
- Connects with the Marketplace Hub to display user information on job postings and applications.

4.2 Job and Gig Posting & Application System

Detailed Description:

This feature enables verified users to post freelance jobs, gigs, or part-time opportunities on the Marketplace Hub. It includes an Application Tracking System that allows job posters to monitor, filter, and manage applicant submissions efficiently.

Job Posting and Application Tracking System



Inputs:

- Job Posting: Title, Description, Payment/Budget, Required Skills, Deadline, and Project Type.
- Application Tracking: Application submission, status changes (“Reviewing,” “Interview,” “Hired,” “Rejected”).

Outputs:

- **System:** A publicly viewable job listing in the Marketplace.
- **User:** A dashboard summarizing all applications with applicant profiles and status indicators.

Interactions:

- Integrates with Enhanced Profiles to display applicant credentials.
- Triggers notifications through the Messaging System upon application status updates.

4.3 Material Exchange System

Detailed Description:

This feature allows users to buy, sell, or rent academic materials such as textbooks, lab equipment, and study aids. All listings include item condition, pricing, and delivery options, creating a trusted campus-based marketplace.



Inputs:

- Listing Creation: Material Name, Category, Condition, Price, Description, Photos, Delivery/Pickup Details.
- Transaction Flow: Buyer initiates purchase/rental; Seller confirms transaction.

Outputs:

- **System:** Searchable and filterable marketplace listings with detailed item previews.

- User: Secure transaction record with escrow-protected payment (if applicable).

Interactions:

- Linked with Escrow Payment System for secure fund handling.
- Connected to Messaging System for buyer–seller communication.

4.4 Skill Sharing System

Detailed Description:

This feature supports peer-to-peer learning and tutoring by allowing students to offer or request skill-based services. Users can specify expertise, rate, and availability, fostering a collaborative campus learning ecosystem.



Inputs:

- Service Posting: Skill Offered, Rate, Availability, Location.
- Search/Filter: Skill Category, Price Range, Ratings.
- Booking: Request submission and Provider confirmation.

Outputs:

- System: Searchable skill listings with tutor profiles.
- User: Booking confirmation and updated skill exchange history.

Interactions:

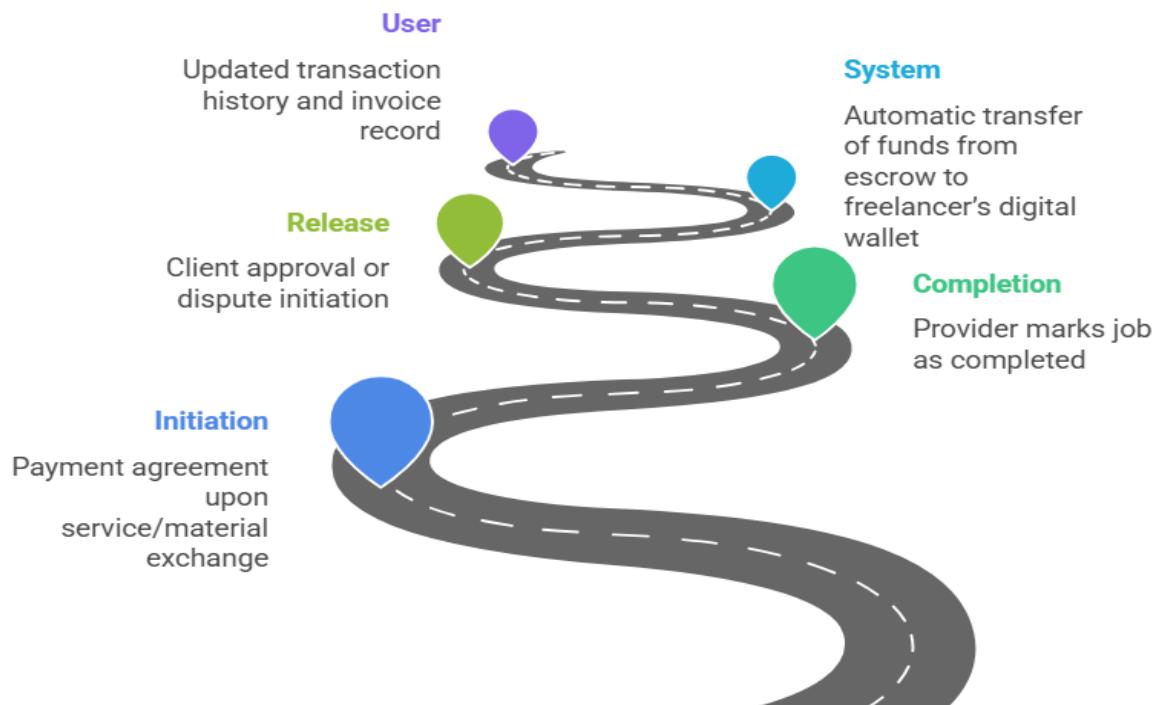
- Updates user Profile Activity and Skill Verification.
- Links to Messaging System for scheduling and communication.

4.5 Escrow Payment System

Detailed Description:

Implements a secure escrow mechanism to hold payments until project completion. Funds are released only after client approval, ensuring fairness and trust in all financial transactions.

Escrow Payment System Sequence



Inputs:

- Initiation: Payment agreement upon service/material exchange.
- Completion: Provider marks job as completed.
- Release: Client approval or dispute initiation.

Outputs:

- System: Automatic transfer of funds from escrow to freelancer's digital wallet.
- User: Updated transaction history and invoice record.

Interactions:

- Integrates with SSLCommerz for payment processing.
- Links to Work Management System for job-payment synchronization.

4.6 Real-Time Messaging & Notification System

Detailed Description:

Provides instant, real-time communication between users via Socket.IO, supporting private chats, group discussions, file sharing, and video/voice calls. It also manages system-wide notifications for updates and alerts.

Real-Time Communication System



Inputs:

- User Inputs: Messages, images, documents, voice/video calls.
- System Inputs: Read receipts, typing indicators, delivery confirmations.

Outputs:

- System: Instant message delivery and persistent chat history.
- User: Interactive chat interface with real-time notifications.

Interactions:

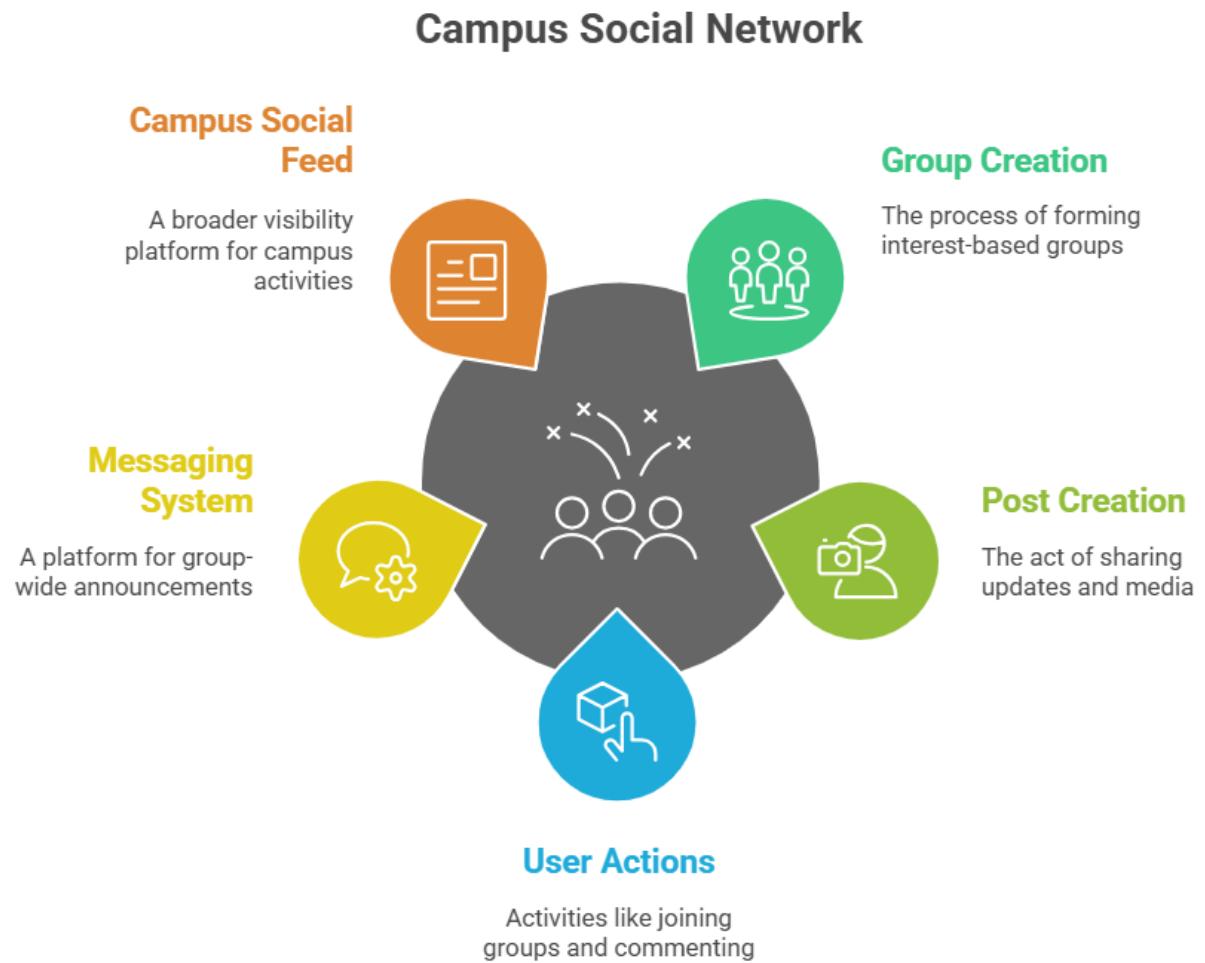
- Central hub connecting Job Applications, Projects, and Group Chats.

- Sends automatic notifications for payments, updates, and system alerts.

4.7 Campus Social Network

Detailed Description:

Facilitates a student community hub where users can join interest-based groups, share updates, and post announcements. The system maintains an organized, filterable campus feed for dynamic engagement.



Inputs:

- Group Creation: Group Name, Description, Category, Privacy Setting.
- Post Creation: Text, Images, Media Links.
- User Actions: Join/Leave group, Comment, Moderate posts.

Outputs:

- **System:** Dedicated group pages with categorized feeds.
- **User:** Personalized feed reflecting followed groups and activities.

Interactions:

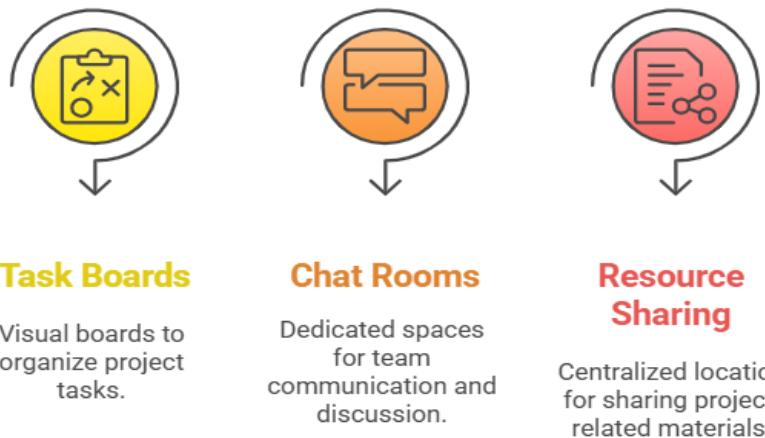
- Integrates with **Messaging System** for group-wide announcements.
- Feeds into **Campus Social Feed** for broader visibility.

4.8 Project & Task Management

Detailed Description:

Enables structured project coordination for freelance or academic collaborations. It provides task boards, chat rooms, and resource-sharing capabilities to enhance teamwork and accountability.

Project Coordination Features



Inputs:

- Task Details: Task Name, Description, Assigned Member, Deadline, Status.
- Project Updates: Progress tracking and team chat entries.

Outputs:

- System: Task board visualization and project timeline.
- User: Notifications for assignments, progress updates, and due dates.

Interactions:

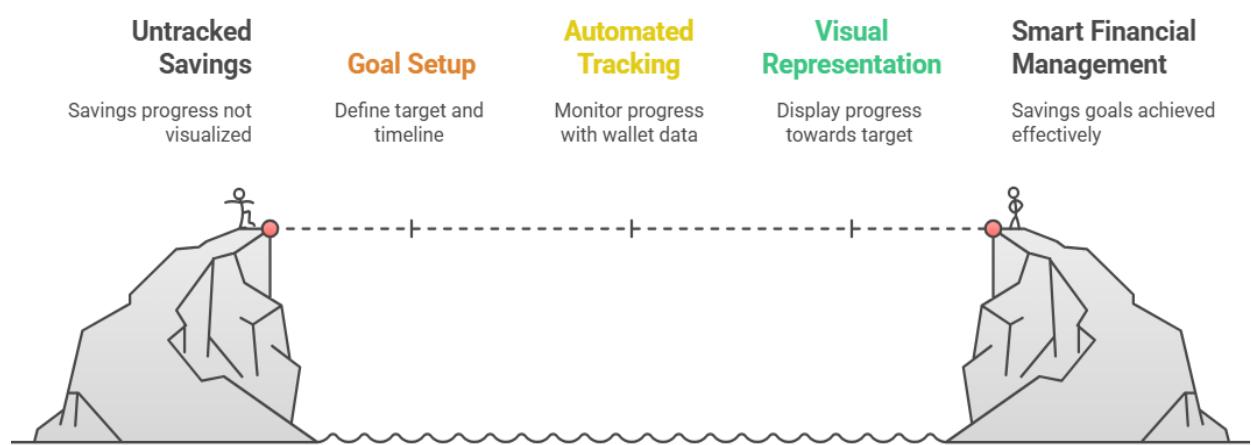
- Links with **Messaging System** for project-based discussions.
- Updates reflected in **User Profiles** to showcase project contributions.

4.9 Financial & Savings Management

Detailed Description:

Allows users to set **financial goals** and track savings progress using integrated wallet data. The system visually represents savings toward specific targets and encourages smart financial management.

Achieve Financial Goals



Inputs:

- Goal Setup: Goal Name, Target Amount, Target Date.
- User Actions: Assign portion of earnings to goal.

Outputs:

- **System:** Real-time visual tracker showing goal achievement percentage.
- **User:** Updated savings summary and performance analytics.

Interactions:

- Draws data from Wallet & Transaction History.
- Displays insights in the Analytics Dashboard for user awareness.

5. External & Non-Functional Requirements

5.1 User Interfaces

Design Philosophy:

The system will deliver a **modern, responsive, and user-friendly interface** built using **React and Tailwind CSS**, supporting both **dark and light modes** for accessibility. The design follows a **mobile-first** strategy, optimized for desktop use, ensuring seamless experience across all devices. A consistent design language will be maintained throughout every module.

Core Interface Modules:

1. Landing Page

- Engaging hero section highlighting the platform's value proposition.
- Key features and benefits displayed concisely.
- Prominent call-to-action buttons for user registration and login.



2. Dashboard Layout

- Sidebar navigation with collapsible menus.
- Header containing profile, notifications, and wallet overview.
- Central workspace for contextual information and actions.
- Modules: *Browse, My Work, Campus Hub, Messages, Profile, Wallet, Log out.*

3. Browse Marketplace

- Tab-based navigation for *Jobs*, *Skills*, and *Materials*.
- Advanced search, filters, and card-style listings.
- Quick action buttons for *Apply*, *Contact*, and *View Details*.

The screenshot shows the 'Campus Marketplace' interface. On the left, there is a sidebar with the 'Earn-n-Learn' logo and a 'Browse' button. The main area has a 'Search' bar and a user profile icon. Below the search bar are three tabs: 'Explore opportunities', 'Post opportunities', and 'Job marketplace' (which is currently selected). Underneath are three more tabs: 'Skill market place' and 'Material marketplace'. The central part of the screen is titled 'Job Marketplace' and displays four job listings for 'Python Programming Assignment'. Each listing includes a brief description, budget (\$150-300), due date (2023-12-15), and a status indicator ('Approaching'). Each listing also has 'View Details' and 'Apply now' buttons.

4. My Work Section

- Tabs for *Applications*, *Received*, *Projects*, *Invoices*.
- Table views with sorting and filtering options.
- Detailed modal pop-ups and visual project status indicators.

My Applications

The screenshot shows a list of applications with the following details:

- Graphic Design Intern**
Posted by user2
Reading
• Remote \$500 Applied: Aug 29, 2025
Buttons: View Details, Edit, Go to project, Applicants
- Frontend Developer**
Posted by user4
Accepted
• On-campus Free Applied: Aug 29, 2025
Buttons: View Details, Edit, Go to project, Applicants

5. Campus Community

- Social feed with infinite scrolling and post creation editor.
- Poll creation, comment threads, and reactions to engage students.

The screenshot shows a sidebar on the left and a main content area on the right.

Sidebar (Left):

- Earn-n-Learn
- Browse
- Campus hub (highlighted)
- My Work
- Wallet
- Message
- Profile
- Log Out

Main Content Area (Right):

- Search bar, User Name, View profile
- Community Groups (Section)
Discover and join groups based on your interests and skills
- Showing 6 groups
- Sort button
- Groups listed:
 - Web Development Experts**
342 members Discussion
A group for professional web developers to share tips, tricks, and job opportunities.
#webdev #programming #javascript
Join Group
 - UI/UX Design Community**
169 members Collaboration
Share your design work, get feedback, and discuss the latest trends in UI/UX design.
#design #ui #ux
Join Group
 - Startup Founders Network**
76 members Crowdfunding
A private group for startup founders to share experiences, challenges, and resources.
#startup #entrepreneurship #funding
Join Group

6. Messaging System

- Split-pane layout with live chat updates via WebSockets.
- File sharing, group chats, and real-time typing indicators.

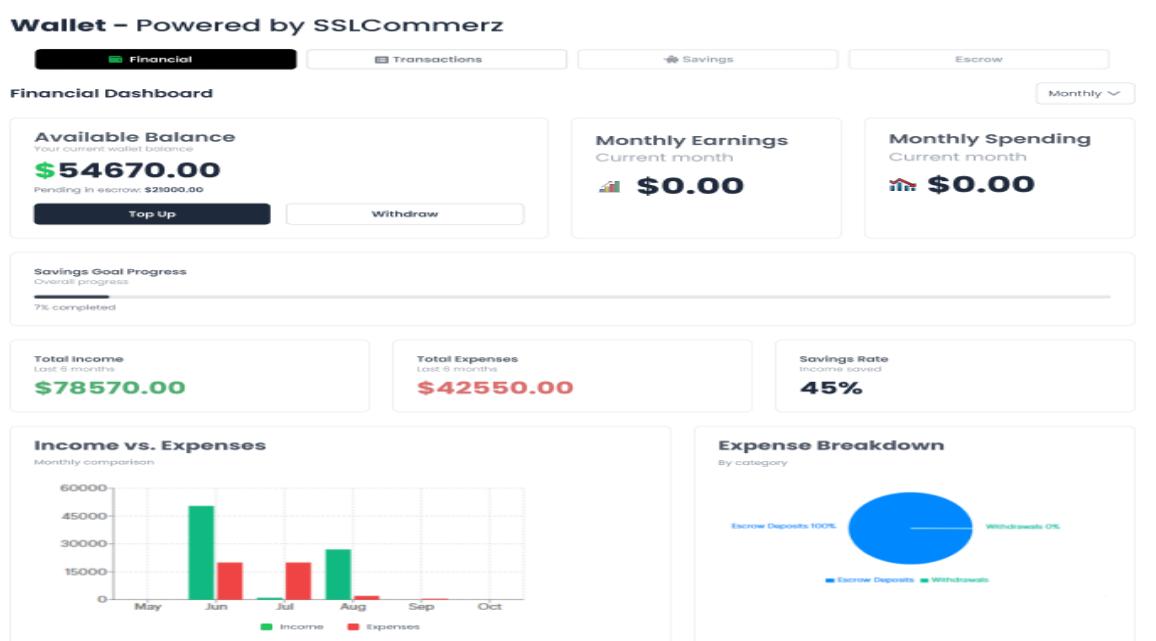
The screenshot shows a communication center interface. On the left is a sidebar with navigation links: Browse, Campus hub, My Work, Wallet, Message (highlighted in teal), Profile, and Log Out. The main area is titled "Communication Center" and contains a "Direct Messages" tab, a "Group chat" tab, and a "Notifications" tab. A search bar at the top right has the placeholder "Search". Below it is a list of messages in a conversational format:

- Sarah Johnson** (10:32 AM): Hey, do you have time to discuss the group project today?
- Mike Thompson** (Yesterday): Sure, I'm free after 3 PM. What specifically did you want to go over?
- Alex Rodriguez** (Monday): I'm struggling with the research section and wanted to see if we could divide the work differently.
- Sarah Johnson** (10:25 AM): Also, do you know if the deadline is still next Friday or did Professor Wilson extend it?
- Mike Thompson** (10:26 AM): Sure, I'm free after 3 PM. What specifically did you want to go over?

At the bottom of the message list is a file attachment icon labeled "project_report.pdf". Below the messages is a text input field with the placeholder "Type a message..." and a toolbar with icons for smiley face, camera, link, and attachment.

7. Wallet & Financial Dashboard

- Overview cards showing *Balance*, *Earnings*, *Spending*, and *Savings*.
- Transaction filtering, secure top-ups, and withdrawals.
- Escrow management for trusted in-app transactions.



8. Project Management

- Kanban-style boards for task tracking.
- Timeline and milestone management.
- Integrated collaboration and time tracking tools.

Navigation & Prototyping:

- Smooth routing using React Router, breadcrumbs, and back navigation.
- Quick access shortcuts for frequently used actions.
- Wireframes and prototypes developed in Figma/Adobe XD.
- Fully responsive across breakpoints: Mobile (≥ 320 px), Tablet (≥ 768 px), Desktop (≥ 1024 px).

5.2 Software Interfaces

Backend Integration:

- Node.js/Express RESTful API architecture.
- JWT-based authentication and Multer for secure file uploads.
- Socket.IO for real-time communication and notifications.

Database Layer:

- **MySQL** relational database using mysql2 connection pooling.
- **Prepared statements** for SQL injection prevention.

Payment Gateway:

- **SSLCommerz** integration for online payments.
- Payment validation, webhook confirmations, and transaction tracking.

Real-Time Communication:

- **Socket.IO** events for messaging, typing indicators, and group rooms.

External APIs & Services:

- Email delivery via SendGrid or Nodemailer.
- Cloud file storage through AWS S3 or Cloudinary.
- Optional SMS notifications for future updates.

File Management:

- Structured local storage under an /uploads directory.
- Static serving through Express middleware.

5.3 Performance Requirements

Speed:

- Page load time < 2 seconds.
- API response time < 500 ms.

- Real-time messaging latency < 100 ms.
- Optimized frontend using lazy loading and code splitting.

Scalability:

- Supports 1,000 + concurrent users.
- Horizontal scaling with load balancers and database replicas.
- CDN integration for static resources.
- Future clustering with Redis adapter for Socket.IO.

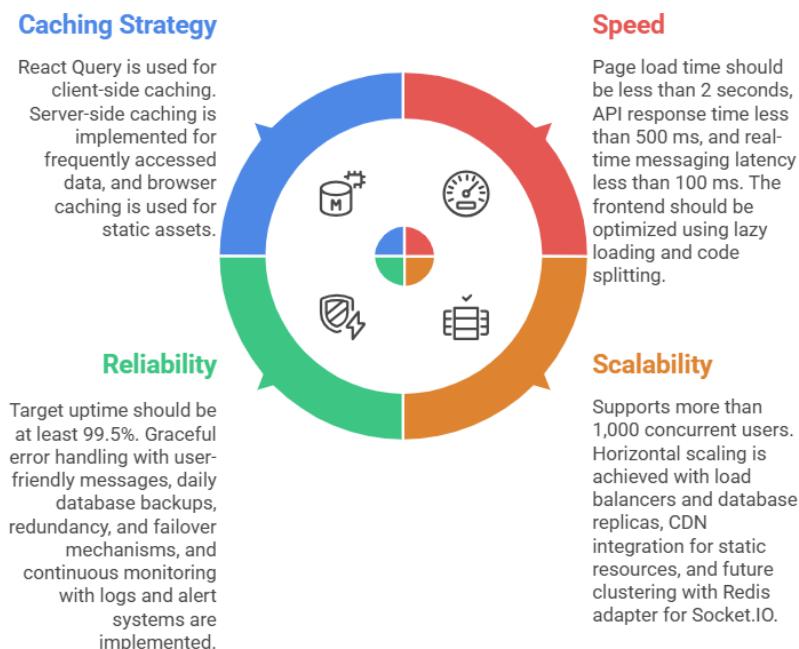


Figure: Earn and Learn Performance Requirements

Reliability:

- Target uptime $\geq 99.5\%$.
- Graceful error handling with user-friendly messages.
- Daily database backups, redundancy, and failover mechanisms.
- Continuous monitoring with logs and alert systems.

Caching Strategy:

- React Query for client-side caching.
- Server-side caching for frequently accessed data.
- Browser caching for static assets.

5.4 Security Requirements

Authentication:

- Secure registration via email and strong password validation.
- JWT tokens for session management and expiry handling.
- bcrypt password hashing and optional multi-factor authentication.

Data Protection:

- End-to-end encryption using HTTPS/TLS.
- Sanitization to prevent XSS and SQL injection.
- File validation (type & size) before upload.



Figure: Earn and Learn Security Requirements

Access Control:

- Role-based permissions (Student, Recruiter, Admin).
- Middleware for authorization and activity auditing.
- API rate limiting to prevent spam or brute-force attacks.

Compliance:

- GDPR-compliant data handling and user consent mechanisms.
- Clear retention and deletion policies for personal data.

5.5 Maintainability

Code Quality & Structure:

- Developed in TypeScript for type safety.
- ESLint and Prettier ensure consistent style.
- Modular, component-driven architecture using atomic design principles.

Documentation:

- Detailed inline documentation for complex logic.
- Comprehensive README for setup and deployment.
- API reference and database schema documentation.

Version Control:

- Git-based workflow with feature branching and pull requests.
- Conventional commit messages for traceability.
- Future CI/CD pipeline integration.

Testing:

- Automated unit, integration, and end-to-end testing suites.
- Manual QA for acceptance testing.

Extensibility:

- Plugin-ready modular architecture.
- API versioning for backward compatibility.
- Environment-based configuration management.
- Regular dependency maintenance and patch updates.

6. Feasibility & Project Management Analysis

This section evaluates the overall viability of the system Earn & Learn — from technical, financial, operational, human resource, market/legal, and scheduling perspectives. It also presents a detailed SWOT analysis to understand our project's internal strengths and weaknesses, as well as external opportunities and threats.

6.1 Technical Feasibility

Technology Stack

- **Frontend:** React, TypeScript, Tailwind CSS, Vite
- **Backend:** Node.js, Express.js
- **Database:** MySQL
- **Real-time Communication:** Socket.IO
- **Payment Gateway:** SSLCommerz

Assessment:

- Mature and widely adopted technologies ensure long-term support and community assistance.
- The stack is highly scalable, suitable for both MVP and production-level deployment.
- Strong in-house expertise in React and Node.js accelerates development.
- TypeScript introduces better type safety and maintainability but may require minor team training.



Figure: Earn and Learn technical feasibility

Infrastructure:

- Local development environment for testing and iteration.
- Hosting via VPS or cloud platforms (AWS, Render, or Heroku).
- Managed MySQL database hosting for reliability and backup.
- File storage via cloud or local storage depending on deployment scale.

So, we can say that our system is technically feasible, using a stable and scalable technology stack that aligns with modern development practices and available expertise.

6.2 Economic / Financial Feasibility

Development Costs:

- Team salaries (developers, designers, QA testers).
- Minimal license fees due to open-source technologies.
- Hosting expenses: approximately **\$50–\$200/month**.
- Payment gateway fees: **1.5–2%** via SSLCommerz.

Expected Revenue Streams:

- **5–10% commission** on completed jobs.
- Premium subscription tiers for advanced users.
- Advertising and sponsored posts.
- Freemium model to attract new users.

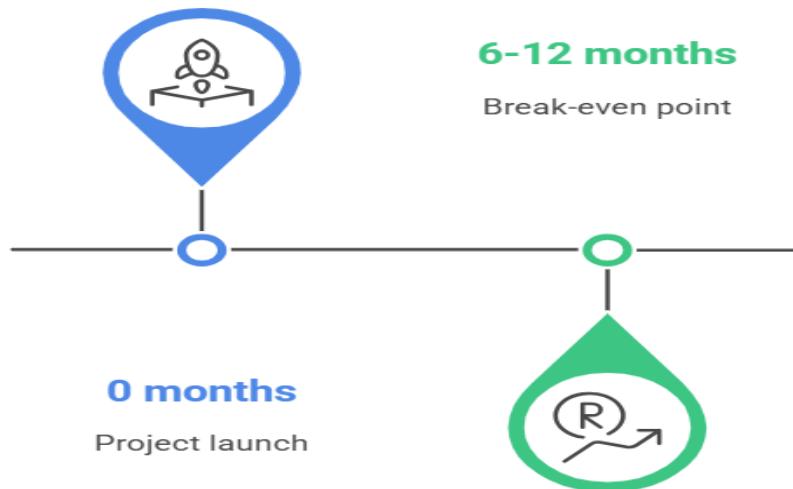


Figure: Earn and Learn Economic Feasibility

ROI Analysis:

- Estimated break-even point: **6–12 months** post-launch.
- Low marketing costs through campus and online promotions.

- High lifetime user value through continuous engagement.

This makes our project financially viable, offering sustainable profit potential and manageable operational expenses.

6.3 Operational Feasibility

System Operations:

- Simple and intuitive onboarding process.
- In-app moderation and content management tools.
- Customer support through help desk or AI chatbot.
- Automated payment handling and analytics dashboards.

Stakeholder Benefits:

- **Students:** Gain flexible, skill-based work opportunities.
- **Recruiters:** Access verified and diverse student talent.
- **Universities:** Promote skill development and employment readiness.

Operational Challenges:

- Ensuring authenticity and quality of job postings.
- Managing user disputes and preventing spam/fraudulent activities.

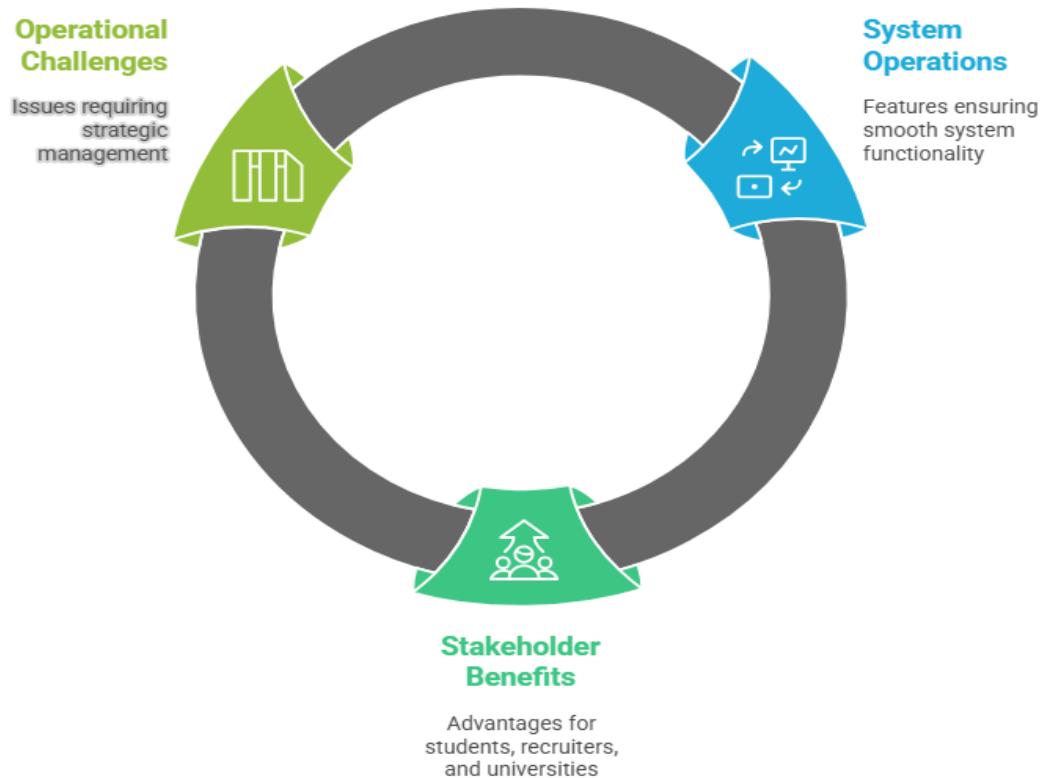


Figure: Earn and Learn Operational feasibility

We can therefore say that this system is operationally feasible, with streamlined processes that ensure usability and scalability in a real-world educational and professional environment.

6.4 Human Resources Feasibility

Required Skills:

- **Frontend:** React, TypeScript, Tailwind CSS
- **Backend:** Node.js, Express, MySQL
- **UI/UX:** Design and prototyping with Figma
- **QA:** Manual and automated testing
- **Project Management:** Agile and sprint-based coordination
- **DevOps:** Deployment and continuous monitoring

Resource Availability:

- Talent can be sourced from in-house student developers or freelance contributors.
- Access to online learning resources ensures continuous skill enhancement.

Estimated Timeline:

- **Development Phase:** 3–6 months
- **Maintenance Phase:** 1–2 developers for long-term support

Human resources are readily available, and the team possesses the technical and managerial skills necessary to execute and maintain the system effectively.

6.5 Market, Legal, and Political Feasibility

Market Overview:

- Expanding global student user base.
- High demand for flexible part-time and freelance work.
- Minimal direct competition in the student-focused freelance niche.

Legal and Regulatory Compliance:

- GDPR and PCI-DSS compliance for data and payment security.
- Clear Terms of Service and Privacy Policy.
- Intellectual Property and content ownership protections.



Figure: Earn and Learn Market, Legal and Political feasibility

Political and Institutional Factors:

- Government support for digital entrepreneurship.
- Favorable socio-economic climate promoting gig work.
- Requires coordination with university policies and student affairs offices.

Risks:

- Regulatory changes in payment or freelance laws.
- Dependency on third-party payment gateways.
- Potential restrictions within certain academic institutions.

That makes Earn & Learn **market-ready and legally sound**, with favorable external conditions for adoption and growth.

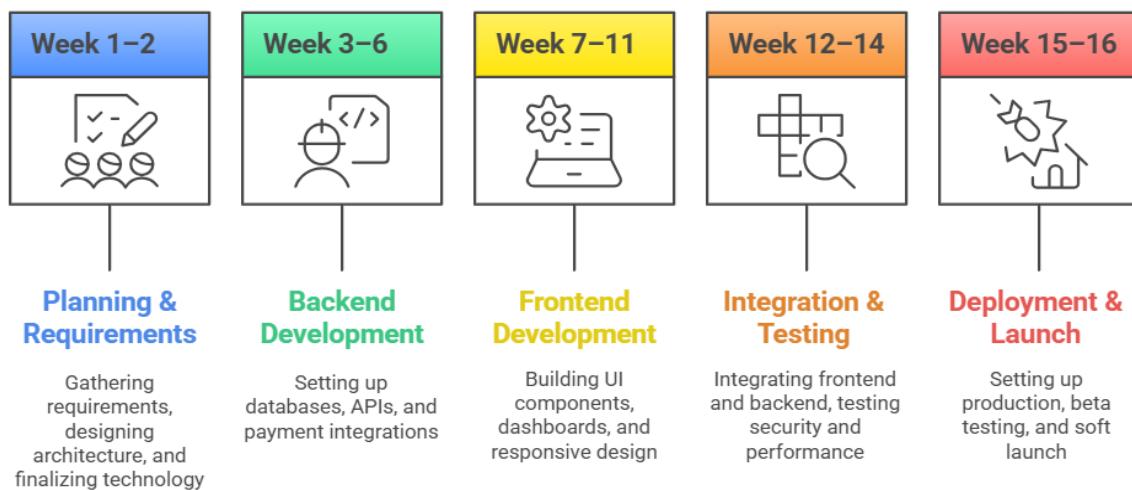
6.6 Schedule Feasibility

Project Timeline Overview (16 Weeks / 4 Months):

Phase	Duration	Key Activities
Phase 1: Planning & Requirements	Week 1–2	Requirement gathering, system architecture, database design, UI/UX prototypes, technology finalization.

Phase 2: Backend Development	Week 3–6	Database setup, authentication, API development (users, jobs, skills, materials, payments), real-time setup, SSLCommerz integration.
Phase 3: Frontend Development	Week 7–11	UI components, dashboards, marketplace, project management, chat system, wallet, responsive design.
Phase 4: Integration & Testing	Week 12–14	Frontend-backend integration, real-time and payment testing, security and performance optimization, user acceptance testing.
Phase 5: Deployment & Launch	Week 15–16	Production setup, SSL configuration, monitoring tools, final bug fixes, beta testing, and soft launch.

Here is the visual representation of the timeline:



Resource Allocation:

- 2–3 Full-stack Developers
- 1 UI/UX Designer
- 1 Project Manager / QA Tester

Critical Path Items:

- Database schema finalization before backend coding.
- Authentication and payment gateway setup as prerequisites.
- Real-time communication requires advanced testing and optimization.

Risk Mitigation:

- Built-in **2-week buffer** for unforeseen delays.
- Parallel module development to save time.
- Agile sprints and weekly progress evaluations.

6.7 SWOT Analysis

The SWOT (Strengths, Weaknesses, Opportunities, and Threats) Analysis provides a strategic evaluation of our Student Freelancer Marketplace System (Earn and Learn), identifying internal and external factors that influence its success. This analysis helps in understanding the system's competitive position, areas of improvement, and future potential within the student freelancing ecosystem.

- **Strengths** represent the system's internal advantages, such as its comprehensive feature set, scalability, and user-focused design.
- **Weaknesses** highlight the internal challenges like technical complexity or dependency on third-party integrations.
- **Opportunities** explore future growth possibilities, including the rising student freelancing trend and integration with educational institutions.
- **Threats** address external risks, such as market competition, regulatory issues, and technological vulnerabilities.

This analysis serves as a foundation for strategic planning and decision-making, guiding the enhancement, sustainability, and long-term scalability of the platform.

In the following table we tried to emphasize the analytical points of the SWOT analysis:

Earn & Learn SWOT analysis

Strengths	Weaknesses
<p>1. Earn and Learn is a comprehensive, all-in-one freelancing ecosystem specifically designed for students.</p> <p>2. Our system used a modern and scalable technology stack, ensuring future growth and flexibility.</p> <p>3. This system focused on a user-centric design with intuitive dashboards for easy navigation and better user experience.</p> <p>4. We implemented an integrated wallet and secure payment system to ensure smooth financial transactions.</p> <p>5. We included strong security measures like JWT authentication, bcrypt encryption, and role-based access control.</p>	<p>1. This system has a large scope, which increases development complexity and project management challenges.</p> <p>2. Our platform requires continuous maintenance and skilled developers to keep it running smoothly.</p> <p>3. The system depends on a third-party payment gateway (SSLCommerz), which could create reliability or integration issues.</p> <p>4. The platform has limited offline functionality, which may affect accessibility in low-connectivity areas.</p> <p>5. Acquiring initial users might be challenging because the platform's success depends on active student participation (network effect).</p>
Opportunities	Threats
<p>1. The growing gig economy and student freelancing trend create a perfect opportunity for platform adoption.</p> <p>2. More educational institutions are adopting digital work and freelancing platforms, which I can collaborate with.</p> <p>3. There's a noticeable market gap for campus-based freelancing systems like mine.</p> <p>4. We can expand the system with AI-driven project matching and a mobile app in the future.</p> <p>5. We can use platform data analytics to provide insights to universities and improve engagement.</p>	<p>1. Established freelancing platforms like Fiverr or Upwork could expand into the student market.</p> <p>2. There may be regulatory or compliance challenges when handling user data and online payments.</p> <p>3. Economic downturns could reduce freelancing demand among students.</p> <p>4. Security risks and technical vulnerabilities may arise due to reliance on third-party services.</p> <p>5. New users may face trust issues or potential fraud concerns, affecting platform credibility.</p>

7.1 Data Flow Diagram (DFD)

7.1.1 Overview

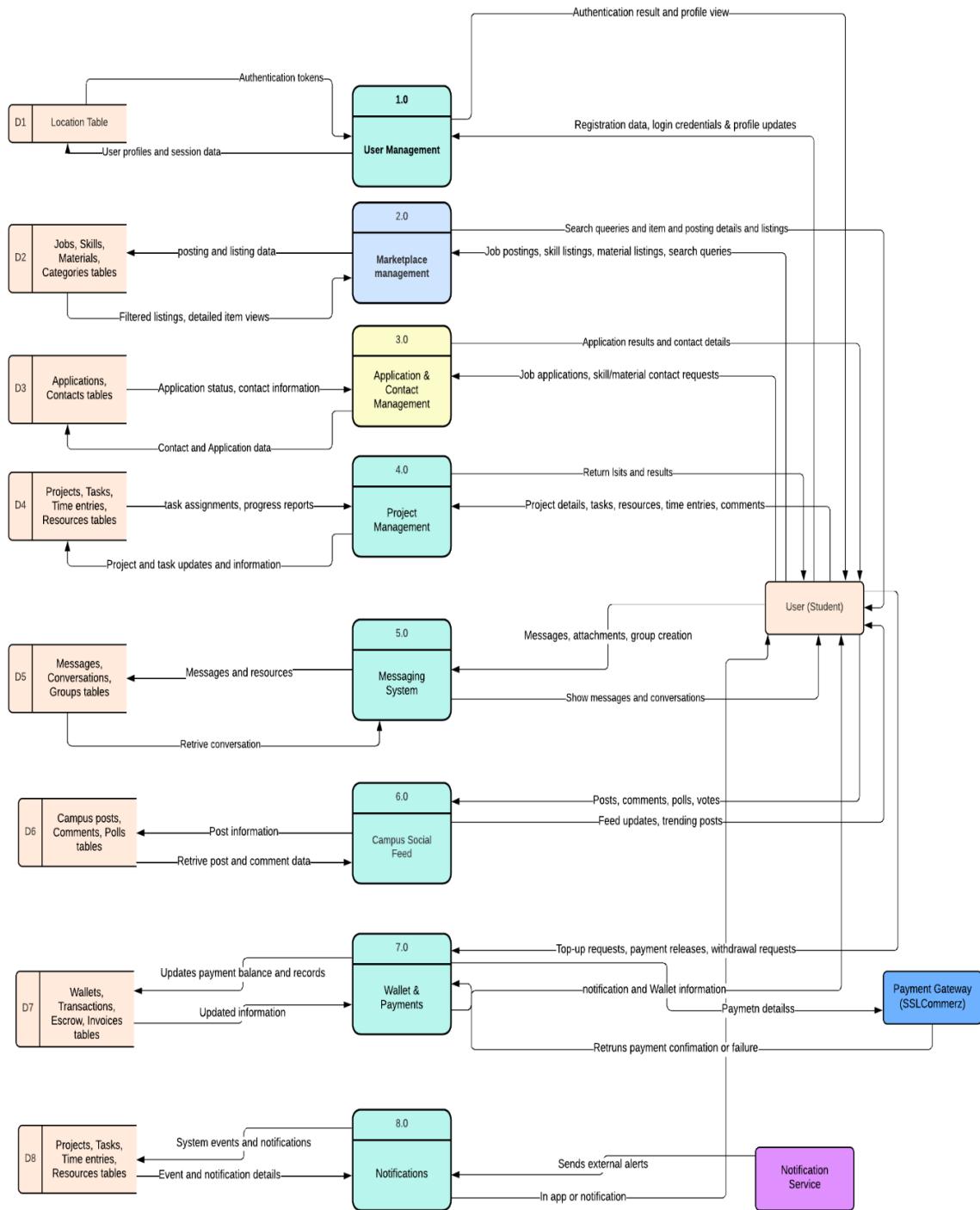
The Data Flow Diagram (DFD) provides a high-level view of the Student-Freelancer Marketplace platform, showing how data moves between users, system processes, and data stores. It helps stakeholders and developers understand the interactions between core system components, external entities, and internal data storage mechanisms.

The diagram breaks the system into eight major processes:

1. **User Management** – Handles registration, authentication, and profile management.
2. **Marketplace Management** – Manages job postings, skills, and material listings.
3. **Application & Contact Management** – Facilitates job applications, skill/material contact requests, and status notifications.
4. **Project Management** – Oversees project creation, task assignments, resource allocation, time tracking, and progress reporting.
5. **Messaging System** – Supports real-time communication between users with message storage and delivery notifications.
6. **Campus Social Feed** – Handles posts, comments, polls, and voting, updating users' feeds in real-time.
7. **Wallet & Payments** – Manages user wallets, top-ups, payments, withdrawals, and escrow transactions securely.
8. **Notifications** – Generates and delivers in-app and email notifications based on system events.

7.1.2 DFD Diagram

In the following is the DFD diagram of our system Ern and Learn which can give a visual illustration of our system's data flow:



7.1.3 DFD Explanation

The Level 1 DFD highlights the following key interactions:

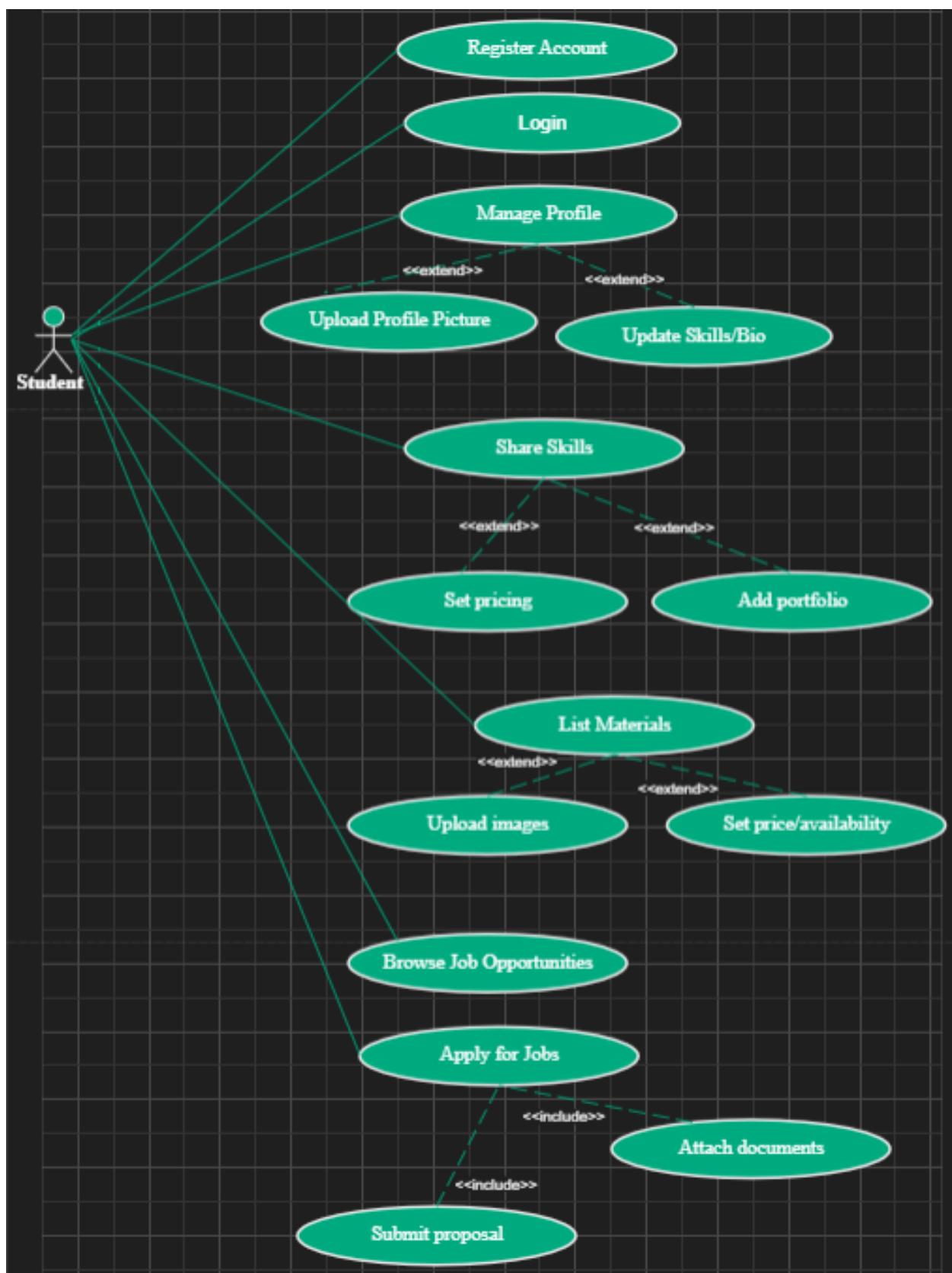
- **User Management (Process 1):** User data flows from registration, login, and profile updates to the Users data store. Authentication tokens and session data are generated to manage user sessions.
- **Marketplace Management (Process 2):** Job postings, skills, and materials are submitted by users and stored in Jobs, Skills, Materials, and Categories tables. Search queries retrieve filtered listings for users.
- **Application & Contact Management (Process 3):** Job applications and contact requests are stored in Applications and Contacts tables. Users receive updates and notifications regarding application statuses.
- **Project Management (Process 4):** Project data, tasks, resources, and time entries are recorded in respective tables (Projects, Tasks, Resources, Time entries). Users can track progress and receive updates.
- **Messaging System (Process 5):** Messages and attachments are stored in Messages, Conversations, and Groups tables. Real-time delivery is managed via Socket.IO with read receipts.
- **Campus Social Feed (Process 6):** Posts, comments, polls, and votes are managed in Campus posts, Comments, and Polls tables. Feed updates and trending posts are broadcast to followers.
- **Wallet & Payments (Process 7):** Handles top-ups, payment releases, and withdrawals. Data is stored in Wallets, Transactions, Escrow, and Invoices tables. The payment gateway ensures secure transactions and updates balances.
- **Notifications (Process 8):** System events trigger notifications stored in the Notifications table. Users receive in-app and email alerts for important activities such as messages, applications, or payments.

The DFD provides a comprehensive view of how data flows across the platform, ensuring alignment between functional requirements, database design, and system processes. It also serves as a blueprint for developers and testers to implement and validate system features accurately.

7.2 Use Case Diagram

The Use Case Diagram provides a high-level, user-focused view of the system, illustrating how the Student actor interacts with various functionalities. It captures key processes such as authentication and profile management (registering, logging in, updating profile, and uploading profile picture) and service offerings (sharing skills, listing materials, browsing job opportunities, and applying for jobs). Optional actions, like setting prices, adding portfolio items, or attaching documents, are represented as extensions, while mandatory steps, such as submitting proposals, are included use cases. This diagram emphasizes the system's capabilities from the user's perspective, helping stakeholders understand the core interactions without detailing internal implementation.

Earn and Learn Use Case Diagram (for Authentication & Skill sharing)



7.3 Class Diagram

The Class Diagram provides a structured overview of our system's core entities, their attributes, methods, and interrelationships. It highlights the object-oriented design, showing how users interact with jobs, skills, materials, applications, projects, tasks, messages, wallets, and notifications. Key relationships are captured, such as one-to-many (e.g., a user can post multiple jobs or skills), one-to-one (e.g., each user has a wallet), and many-to-many associations (e.g., users and groups, or users in conversations). This diagram serves as a blueprint for both database schema design and backend implementation, ensuring clarity of data flow and interactions between objects within the system. It also helps developers understand responsibilities of each class and supports maintainability, scalability, and integration of additional features in future iterations.

Purpose and Benefits:

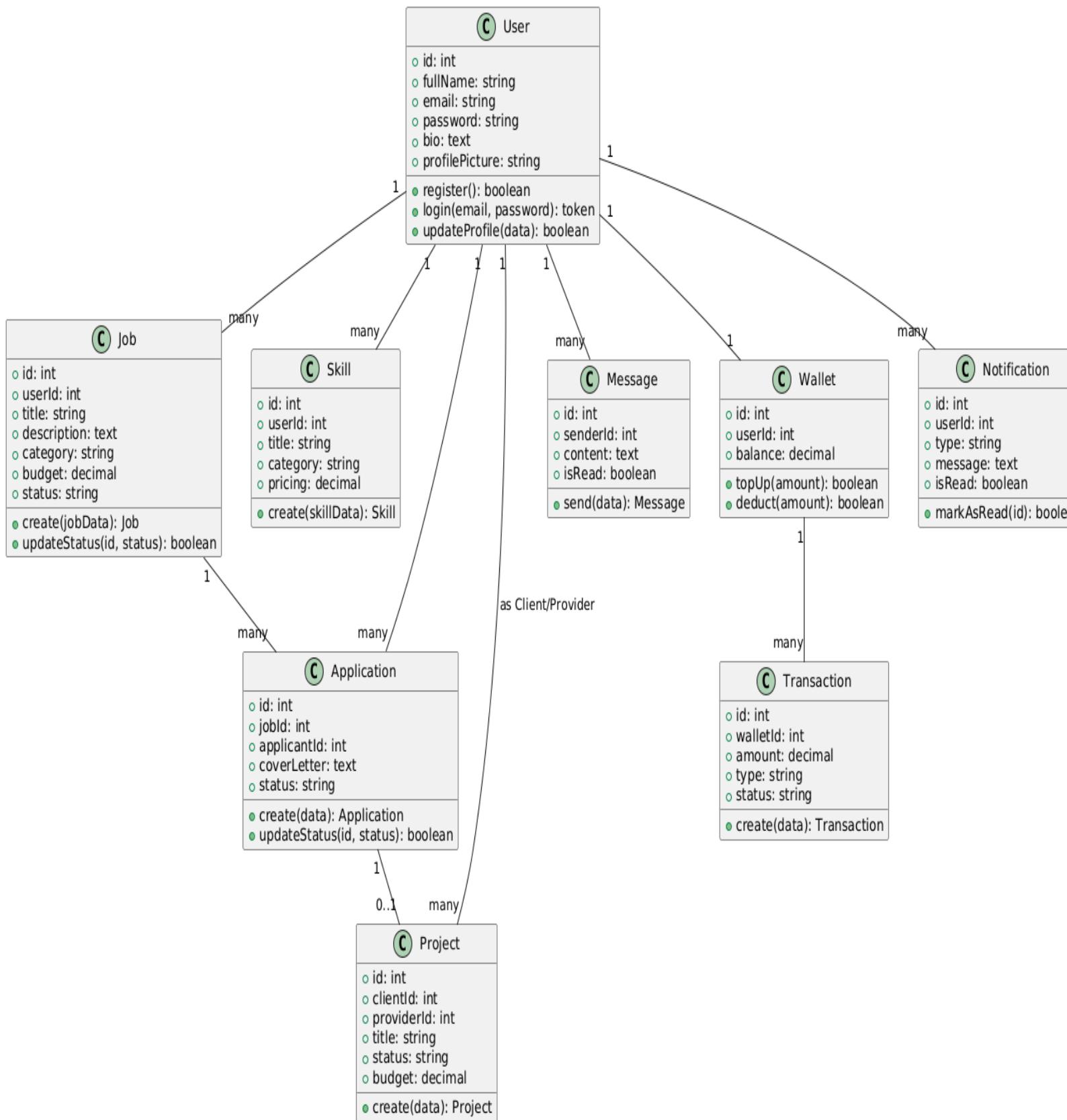
- Provides a visual overview of the system's structure.
- Clarifies the data model, making database design easier.
- Helps developers understand object interactions before coding.
- Ensures **consistency** between the design and actual implementation.

Key Classes and Their Roles

1. **User** – Manages student profiles, registration, login, and profile updates.
2. **Job** – Handles posting, updating, and managing freelance jobs or gigs.
3. **Skill** – Represents skills offered by users, with creation and search functionalities.
4. **Material** – Manages marketplace listings for textbooks, equipment, or study items.
5. **Application** – Tracks user applications to jobs/projects, including status updates.
6. **Project** – Connects clients and providers through accepted applications, managing tasks and progress.
7. **Task** – Handles creation, updates, and tracking of individual project tasks.
8. **Message** – Supports real-time messaging, including sending, reading, and retrieval.
9. **Wallet & Transaction** – Manages user wallets, payments, top-ups, and transaction history.
10. **Notification** – Sends alerts for system events like applications, messages, and payments.

Here is the visual description of our Class Diagram:

Earn and Learn - Class Diagram



7.4 Sequence Diagram

The Sequence Diagram illustrates the chronological flow of interactions between users and various system components in our system (Earn and Learn). It captures how students, freelancers, and clients interact with the frontend (React), backend (Express), database (MySQL), and external services like Email, Socket.IO, and Payment Gateways during different workflows.

By visualizing these sequences, the diagrams provide insight into how the system handles data flow, manages dependencies, and coordinates multiple modules to ensure seamless user experience. They also help developers and testers understand timing, interactions, and potential bottlenecks, ensuring that each user action triggers the expected responses across the system. Overall, sequence diagrams are essential for verifying that the platform's dynamic behavior aligns with functional requirements, providing both clarity and guidance for system development and maintenance.

Purpose

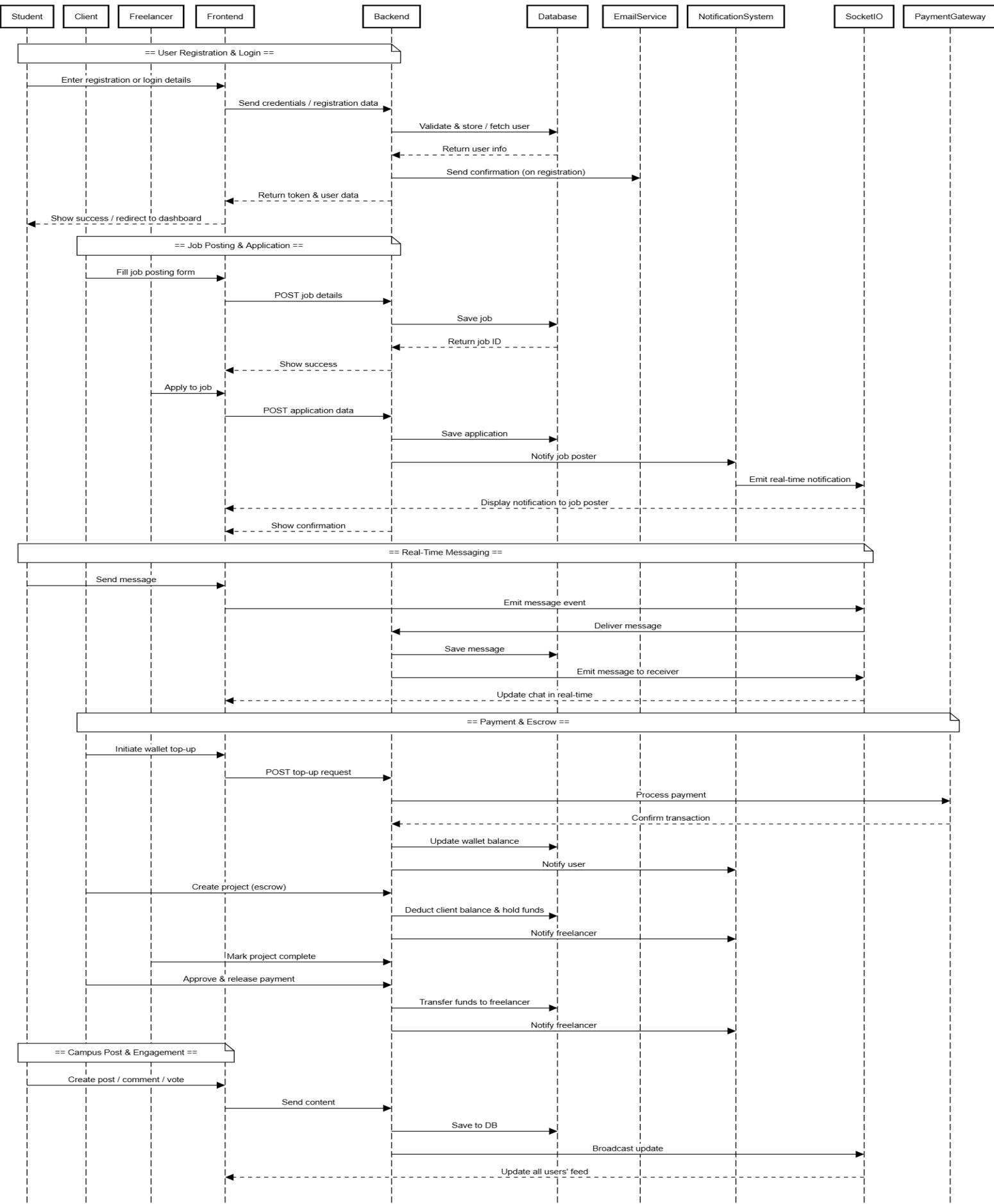
The purpose of the Sequence Diagram is to visualize the order of interactions between users and system components in the Student Freelancer Marketplace platform. It helps in understanding how data and messages flow over time, ensuring correct execution of workflows like user registration, job posting, applications, messaging, and payments.

Key Components

1. **Actors/Users:** Student, Freelancer, Client – entities interacting with the system.
2. **System Components/Objects:** Frontend (React), Backend (Express), Database (MySQL), External Services (Email, Socket.IO, Payment Gateway).
3. **Messages/Interactions:** Requests, responses, notifications, and updates exchanged between users and system modules.
4. **Lifelines:** Represent the existence of objects over the interaction period.
5. **Sequence of Actions:** The chronological order in which operations occur to complete workflow

Here is the visual description of our sequence diagram:

Sequence Diagram for Earn and Learn



8. Prototype Design

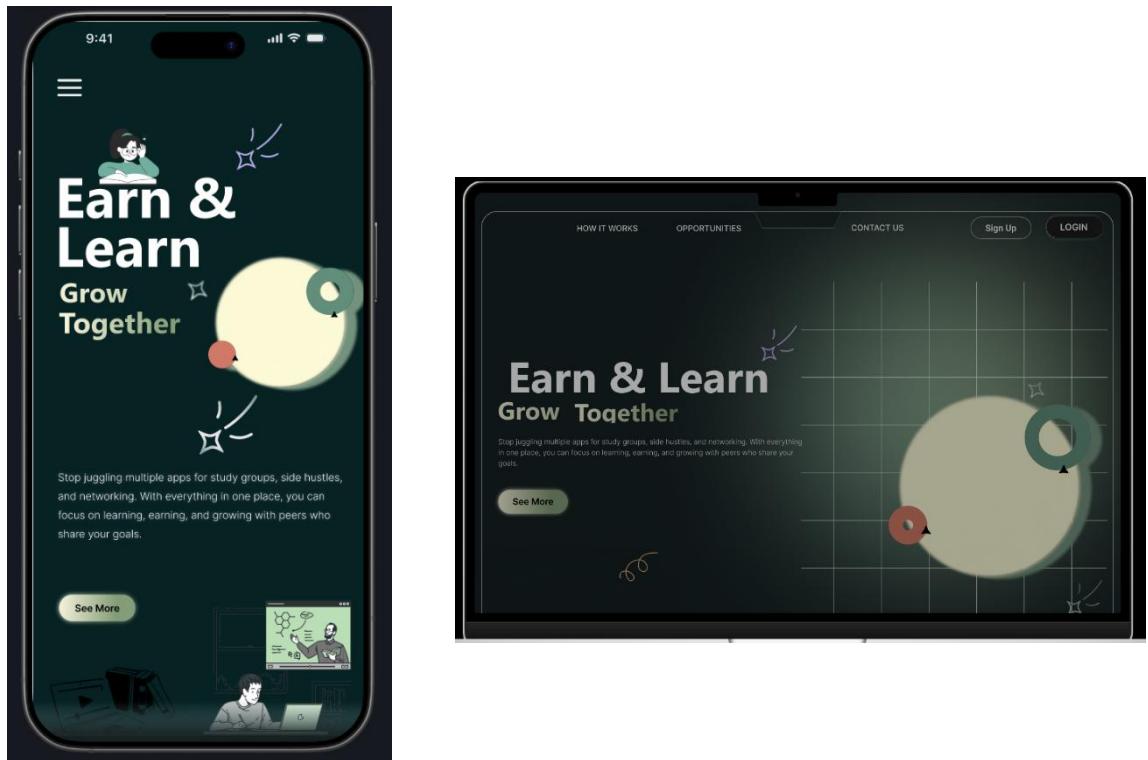
8.1 Overview

The prototype design illustrates the visual layout and user interface (UI) of the Student Freelancer Marketplace system, showcasing **both** web and mobile views. It helps visualize how users interact with the platform's main modules — including registration, marketplace, job applications, messaging, and wallet management — before full system implementation. The prototypes were created using **Figma**, following a user-centered design approach.

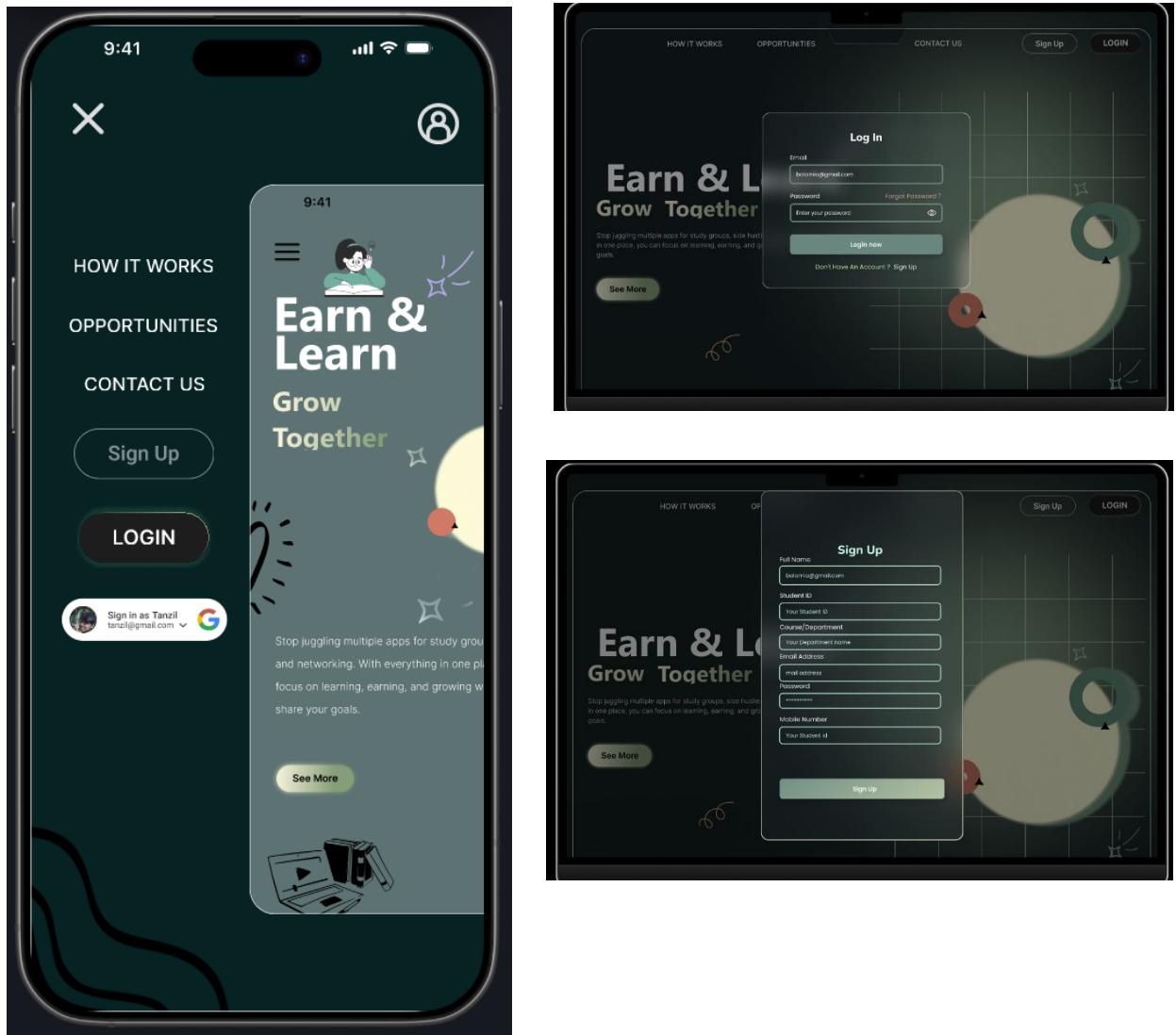
8.2 Screenshots of Prototypes

Here are the screenshots of our web and mobile application prototype:

8.2.1 Earn and Learn (Landing Page):



8.2.2 Earn and Learn (User Sign up and Log in Page):



8.2.3 Earn and Learn (Browse):

Campus Marketplace

Explore opportunities

64 jobs found Sorted by Newest ▾

Job Marketplace

Coding Freelance

Python Programming Assignment
Need help with a Python data analysis project.....
Budget: \$150-300
Due: 2023-12-15

Python Programming Assignment
Need help with a Python data analysis project that involves cleaning data, creating visualizations, and building predictive models.
Budget: \$150-300
Due: 2023-12-15

64 jobs found Sorted by Newest ▾

Skill Marketplace

Teaching Freelance

Web Development Mentorship
Offering one-on-one mentorship in modern web
Availability: Weekends Duration: 1-2 hrs/session

Teaching Freelance

Web Development Mentorship
Offering one-on-one mentorship in modern web ...
Availability: Weekends Duration: 1-2 hrs/session

Earn-n-Learn

Browse Campus hub My Work Wallet Message Profile Log Out

Campus Marketplace

Welcome User

Explore opportunities **Post opportunities**

Job marketplace **skill market place** **Material marketplace**

Job Marketplace

Coding Intermediate Freelance

Python Programming Assignment
Need help with a Python data analysis project that involves cleaning data, creating visualizations, and building predictive models.
Budget: \$150-300
Due: 2023-12-15

Coding Intermediate Freelance

Python Programming Assignment
Need help with a Python data analysis project that involves cleaning data, creating visualizations, and building predictive models.
Budget: \$150-300
Due: 2023-12-15

Coding Intermediate Freelance

Python Programming Assignment
Need help with a Python data analysis project that involves cleaning data, creating visualizations, and building predictive models.
Budget: \$150-300
Due: 2023-12-15

8.2.4 Earn and Learn (Post):

Campus Marketplace

Post opportunities

Post a Job

Post Job **Share a skill** **List a material**

Job Title
e.g., Java Developer

Description
Describe your skill, experience level, and you can offer....

Category
Academic Help

Educational support, tutoring, research assistance

Job Type **Payment/ Compensation**

Select job type \$15/hr

Application Deadline
mm/dd/yyyy

Location

Earn-n-Learn

Browse Campus hub My Work Wallet Message Profile Log Out

Campus Marketplace

Welcome User

Explore opportunities **Post opportunities**

Post Job **Share a skill** **List materials**

Job Title
e.g., Java Developer

Description
Describe your skill, experience level, and you can offer....

Category
Academic Help

Job Type **Payment/Compensation**

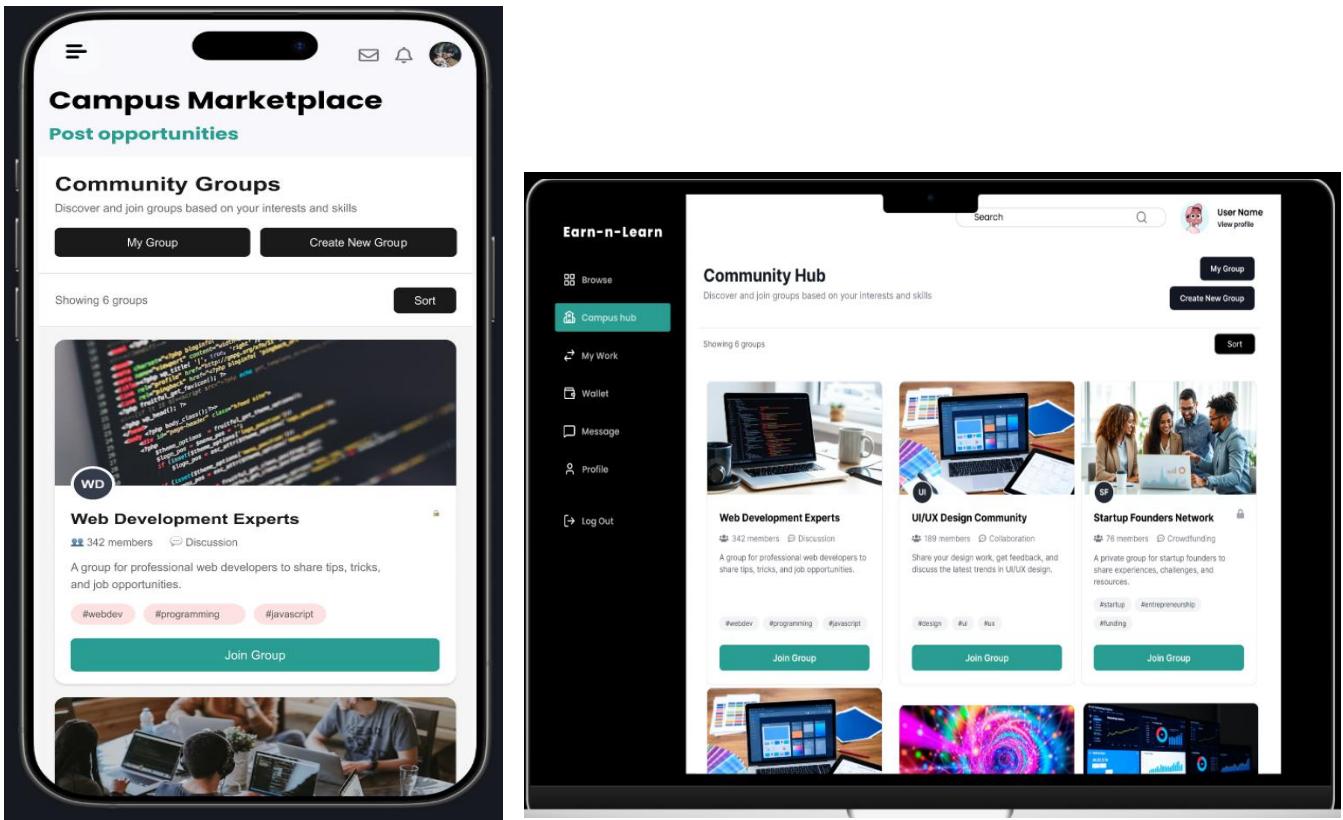
Select job type \$15/hr

Application Deadline **Location**

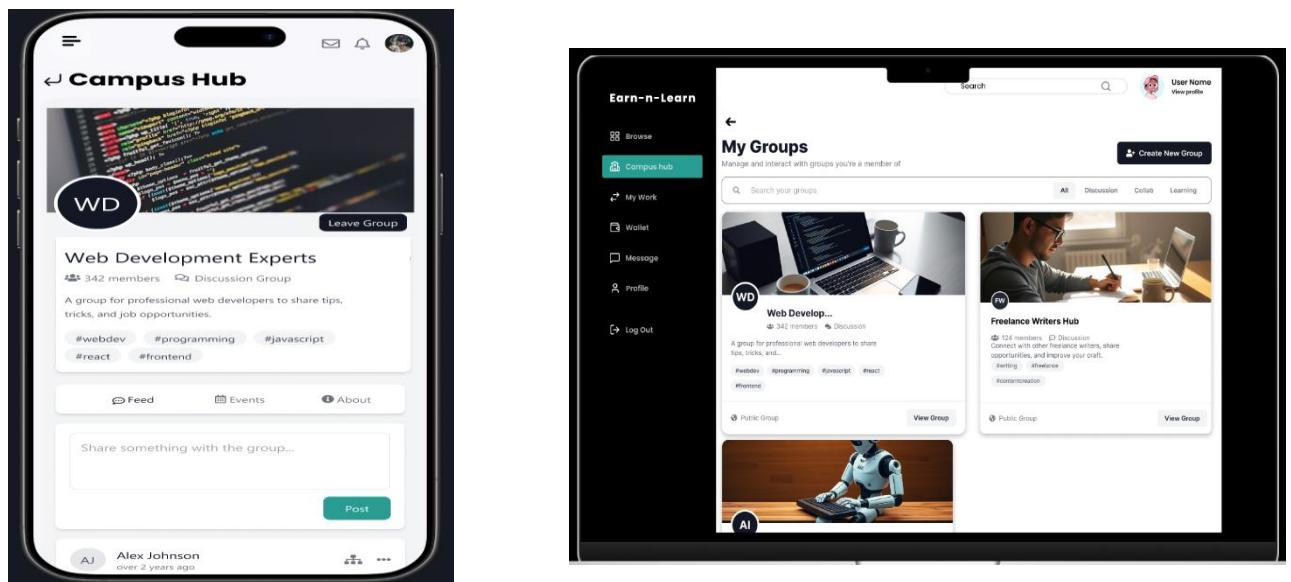
mm/dd/yyyy

Requirements (Optional)

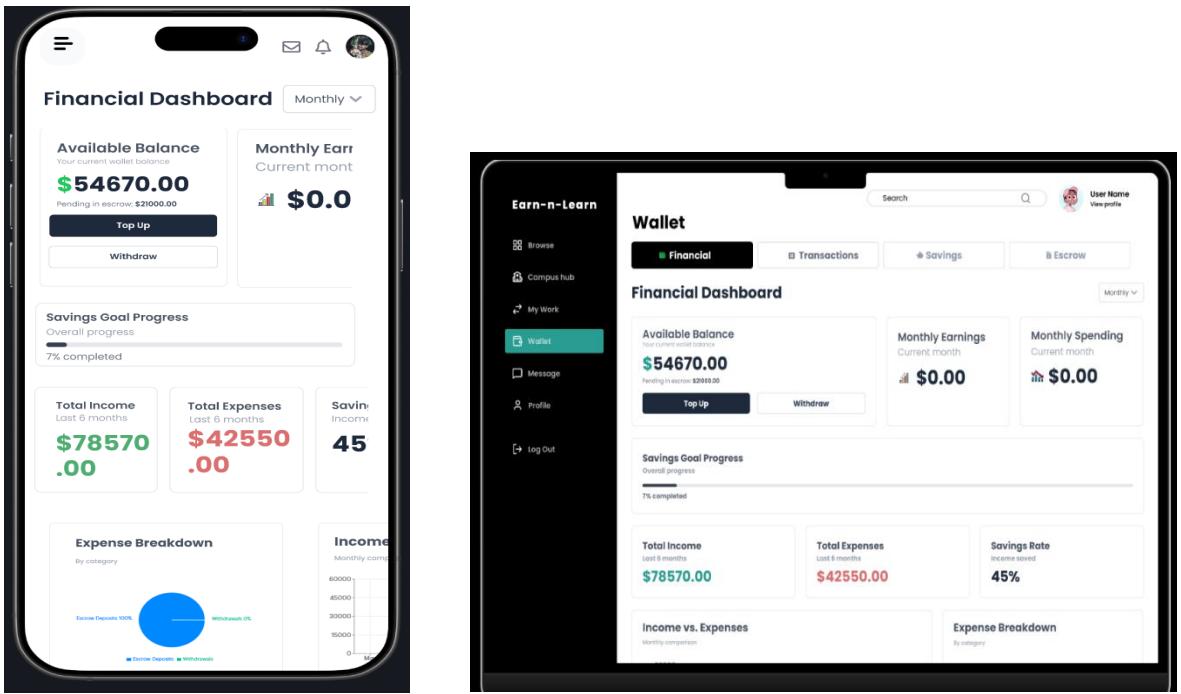
8.2.5 Earn and Learn (Community Hub):



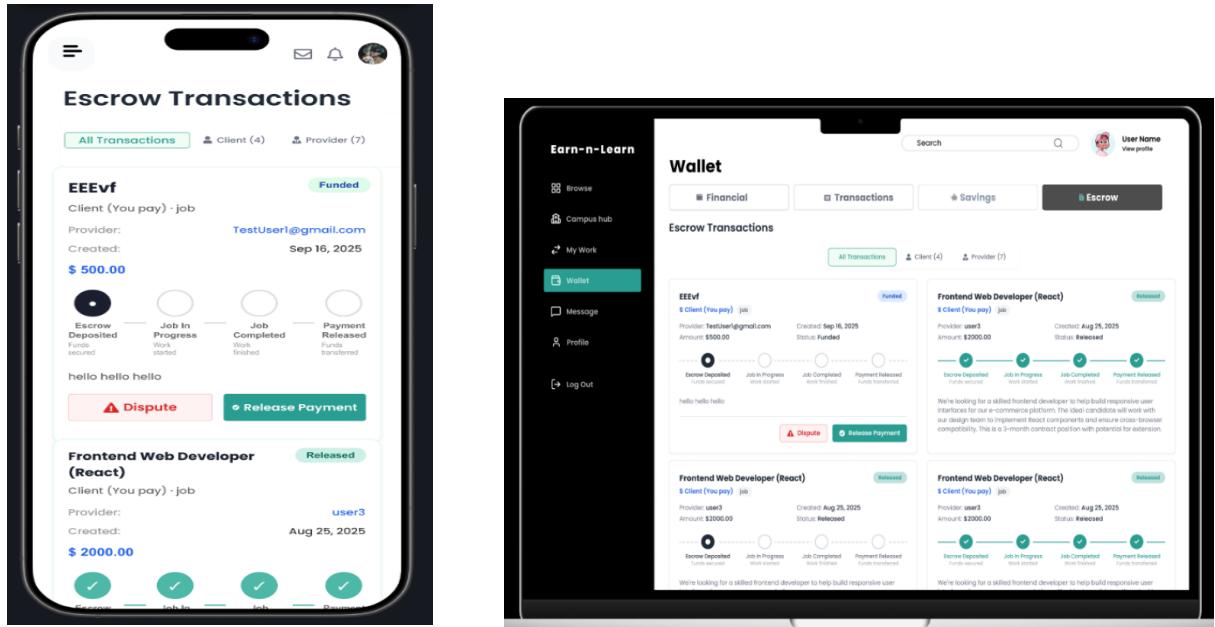
8.2.6 Earn and Learn (Campus Group):



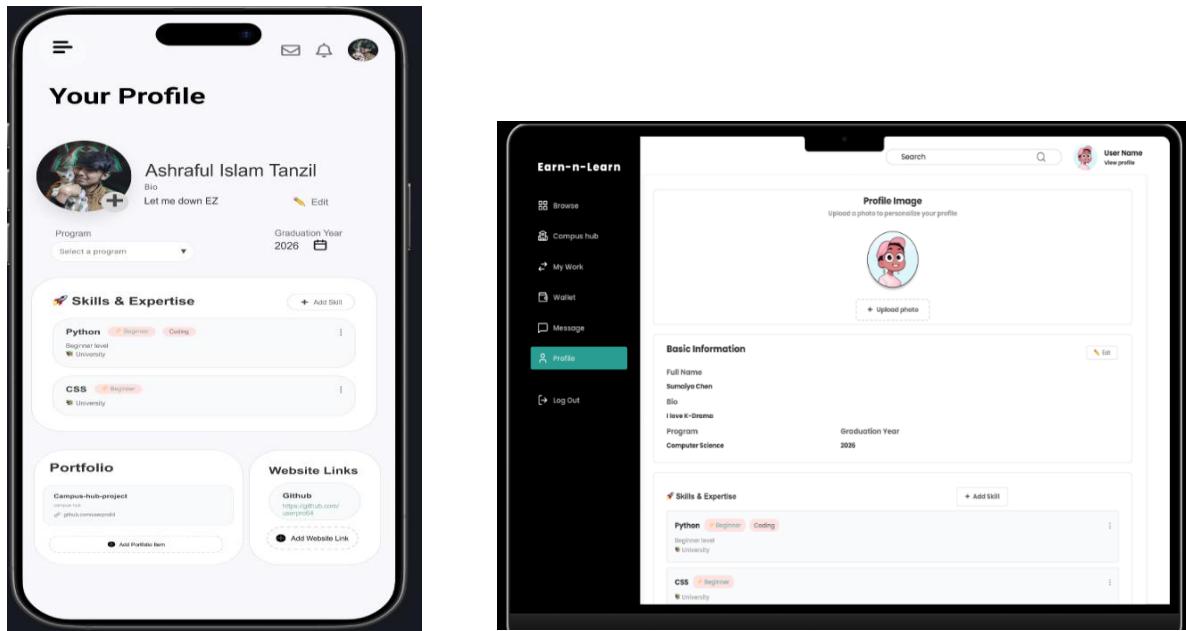
8.2.7 Earn and Learn (Financial Management):



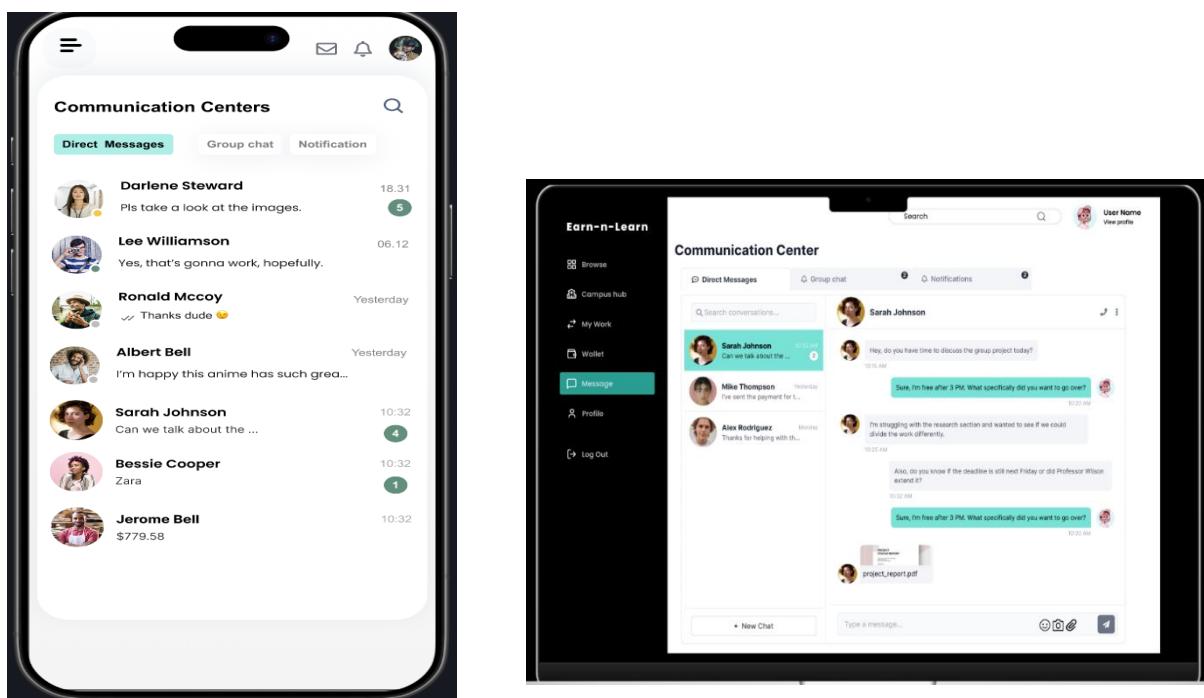
8.2.8 Earn and Learn (Escrow system):



8.2.9 Earn and Learn (User Profile):



8.2.10 Earn and Learn (Communication Centers):



8.2.11 Earn and Learn (My Work)

Mobile App Screenshot:

- Summary:** 2 Active Projects, 1 Completed, 2 My Posts, 1 Accepted Application.
- Active Projects:** Frontend Developer Internship (Tech Corp)
 - Progress: 45%
 - Total tasks: 20, Completed: 9, In Progress: 5, Pending: 6
 - Duration: Sep 1, 2025 - Dec 1, 2025
 - Supervisor: John Smith
 - Mid-term Review Due: Oct 15, 2025
 - [Open Project →](#)
 - [View Original Application](#)

Web Interface Screenshot:

- Summary:** 2 Active Projects, 1 Completed, 2 My Posts, 1 Accepted Application.
- Active Projects:** Frontend Developer Internship (Tech Corp) and Mobile App Development (StartupXYZ)
 - Frontend Developer Internship Progress: 45%
 - Total tasks: 20, Completed: 9, In Progress: 5, Pending: 6
 - Duration: Sep 1, 2025 - Dec 1, 2025
 - Supervisor: John Smith
 - Mid-term Review Due: Oct 15, 2025
 - [Open Project →](#)
 - [View Original Application](#)
 - Mobile App Development Progress: 78%
 - Total tasks: 15, Completed: 12, In Progress: 2, Pending: 1
 - Duration: Aug 15, 2025 - Nov 15, 2025
 - Supervisor: Sarah Lee
 - Final Deployment Due Nov 15, 2025
 - [Open Project →](#)
 - [View Original Post](#)

8.2.12 Earn and Learn (Project Management)

Mobile App Screenshot:

- Project Overview:** Frontend Developer Internship (Tech Innovations Ltd.)
 - Duration: Sep 1, 2025 - Dec 1, 2025
 - Members: 4
 - [Share](#) [+ New Task](#)
- Project Progress:** 45%
- Recent Tasks:**
 - Setup Development Environment (Completed, 2025-09-05)
 - Create User Authentication Module (In Progress, 2025-10-15)
 - Design Dashboard UI Components

Web Interface Screenshot:

- Project Overview:** Frontend Developer Internship (Tech Innovations Ltd.)
 - Duration: Sep 1, 2025 - Dec 1, 2025
 - Members: 4
 - [Share](#) [+ New Task](#)
- Project Progress:** 45%
- Recent Tasks:**
 - Setup Development Environment (Completed, 100%)
 - Create User Authentication Module (In Progress, 60%)
 - Design Dashboard UI Components (In Progress, 30%)
- Team Members:** John Smith (Supervisor), Sarah Lee (Designer), Mike Chen (Backend Dev), You (Frontend Intern)
- Quick Actions:** + New Task, + Unlink Resource

8.3 UI Design Principles – Guidelines Followed

This section outlines the key UI design principles applied throughout the Earn and Learn prototype (Figures 1–12). The design emphasizes usability, consistency, and accessibility across all web and mobile interfaces.

1. Consistent Design System

Maintained across all screens (Figures 1–12), ensuring harmony in colors, typography, and spacing. Semantic color tokens (primary, surface, background, success, warning) and a modular type scale were implemented for visual consistency.

2. Component Reusability

Applied in Sign-Up, Log-In, Post, Communication, and Profile screens (Figures 2, 4, 5, 9). Buttons, cards, inputs, and modals were designed as reusable components with variant states (default, hover, focus, disabled) for consistent design and easier maintenance.

3. Responsive and Adaptive Layouts

Used in Landing Page, Browse, Community Hub, and Campus Group (Figures 1, 3, 6, 7). Flexible grids ensure layouts adapt seamlessly to different screen sizes, maintaining readability and usability on both mobile and desktop.

4. Visual Hierarchy and Typography

Applied to Landing Page, My Work, and Project Management screens (Figures 1, 11, 12). Clear typographic hierarchy guides user attention, with bold headings and properly spaced body text for enhanced readability.

5. Color and Theme System

Implemented across all pages (Figures 1–12), supporting both light and dark themes. Colors meet AA contrast standards ($\geq 4.5:1$) and the emerald accent is used for CTAs, icons, and highlights.

6. Spacing and Layout Rhythm

Applied to Financial Management, Escrow, and Project Management screens (Figures 7, 8, 12). A standardized 4px/8px grid ensures balanced margins, padding, and white space for better content scanning.

7. Interactive and Feedback States

Used in Buttons, Form Inputs, Post Creation, and Navigation screens (Figures 4, 6, 7, 12). Hover, focus, active, and disabled states, along with smooth transitions (200–300 ms), communicate system responses effectively.

8. Accessibility Compliance

Applied to Navigation, Communication, Profile, and Form Screens (Figures 2, 9, 11). WCAG 2.1 AA standards were followed, with visible focus indicators, ARIA roles, high contrast, and descriptive labels for icons.

9. Standard Design Patterns

Implemented in Modals, Cards, Navigation, and Search Bars (all applicable screens). Consistent overlays, card elevation, padding, and navigation hierarchy improve usability and familiarity.

10. Brand Consistency

Maintained in Landing Page, Community Hub, and Financial Management screens (Figures 1, 6, 8). Emerald accent, minimalist icons, and balanced typography reinforce the platform's brand identity.

9. Conclusion

Earn and Learn, the Student Freelancing and Resource Exchange System was developed to address the growing need for a unified digital ecosystem that empowers students to showcase their skills, find freelance or part-time opportunities, collaborate on academic or professional projects, and exchange materials within their campus network. Through this system, we successfully designed an integrated platform that combines multiple functional modules — including User Registration & Enhanced Profiles, Marketplace Hub for Jobs and Materials, Skill Sharing System, Escrow-based Payment Integration, Real-time Messaging, Campus Social Network, Project Management, and Financial Tracking.

This project demonstrates how technology can foster a self-sustaining student economy, promote peer-to-peer collaboration, and enhance employability within the university environment. The modular architecture, secure payment gateway (SSLCommerz), and real-time interaction features ensure that the system is both scalable and user-friendly. Moreover, strong security measures like JWT authentication, bcrypt encryption, and role-based access control reinforce system reliability and user trust.

9.1 Future Enhancements

For future development, the system can be expanded with:

- AI-driven matching and recommendation systems to connect freelancers with the most suitable opportunities based on skills and history.
- Mobile app integration for enhanced accessibility and instant notifications.
- Advanced analytics and dashboards for universities to monitor student engagement and freelancing performance.
- Gamification elements to encourage participation and reward consistent contributors.
- Blockchain-based credential verification for secure validation of student achievements and work records.

In conclusion, this project lays the groundwork for a modern, inclusive, and intelligent student freelancing ecosystem, bridging the gap between academia and industry while nurturing entrepreneurial and technical growth within the student community.

10. References

1. *Figma Official Documentation* – Collaborative Interface Design Tool, <https://help.figma.com>
2. *Figma Design Systems Guide* – Creating and Managing Reusable Components, <https://www.figma.com/design-systems/>
3. *Figma for UX/UI Design – Designing Interactive Prototypes and Wireframes*, <https://www.figma.com/resources/learn-design/>
4. *Figma Community Resources* – Templates and UI Kits, <https://www.figma.com/community>

5. *Jira Software Documentation* – Agile Project Management & Issue Tracking, Atlassian, <https://support.atlassian.com/jira-software-cloud/>
6. *draw.io (diagrams.net) Documentation* – Diagramming and DFD Design, <https://www.diagrams.net/doc/>
7. *Lucidchart: UML & DFD Diagram Resources* – <https://www.lucidchart.com/pages/uml-diagram>
8. *sequencediagram.org* – Online UML Sequence Diagram Tool, <https://sequencediagram.org/>
9. *Socket.io Documentation* – Real-time Bidirectional Event-based Communication, <https://socket.io/docs>
10. *SSLCommerz Payment Gateway API Reference*, <https://developer.sslcommerz.com/>
11. *MongoDB Documentation* – NoSQL Database Guide, <https://www.mongodb.com/docs/>
12. *Node.js Documentation* – Server-side JavaScript Framework, <https://nodejs.org/en/docs>
13. *React.js Official Documentation* – Front-end Library for UI Development, <https://react.dev/>
14. *Express.js Guide* – Web Framework for Node.js, <https://expressjs.com/>
15. *JWT.io* – JSON Web Token Implementation Guide, <https://jwt.io/>
16. *BCrypt.js* – Secure Password Hashing Library for Node.js, <https://www.npmjs.com/package/bcrypt>
17. *GitHub Documentation* – Version Control and Repository Management, <https://docs.github.com/>
18. *freeCodeCamp* – Open Source Learning Resources for Full Stack Development, <https://www.freecodecamp.org/>
19. *Miro* – Online Collaborative Whiteboard for UX and System Planning, <https://miro.com/>
20. *Notion* – Team Documentation and Collaboration Tool, <https://www.notion.so/>

11. Appendices

The appendices provide supplementary information that supports the main content of this SRS. They include definitions of key terms, acronyms, and abbreviations, as well as references to tools, software, and diagrams used throughout the project. This section helps readers and developers understand specialized terminology, design references, and resources, ensuring clarity and consistency in system development and documentation.

Appendix A – Glossary

Term / Acronym	Definition
SRS	Software Requirements Specification – A document that describes the requirements for a software system.
UI	User Interface – The visual layout and interactive elements of the system.
UX	User Experience – The overall experience and satisfaction of users interacting with the system.
JWT	JSON Web Token – A secure token used for user authentication.
DFD	Data Flow Diagram – A diagram representing the flow of data in the system.
CRUD	Create, Read, Update, Delete – Standard database operations.
Escrow	A financial arrangement where funds are held by a third party until conditions are met.
Socket.IO	A library enabling real-time, bi-directional communication between clients and servers.
Portfolio	A collection of user's work, certifications, or skill demonstrations.
Marketplace Hub	Central platform where jobs, skills, and materials are listed and browsed.
Frontend	Client-side portion of the system, built using React.
Backend	Server-side portion of the system, built using Express.js and Node.js.
Database	MySQL database storing all persistent system data.
Notification System	Component managing in-app notifications, emails, and alerts for users.
SSLCommerz	Payment gateway integration for secure transactions.
Campus Feed	Social network feed displaying posts, polls, and announcements within the campus.
Task Management	Feature enabling users to create, assign, and track tasks within projects.

Appendix B – Tools & Software Used

Tool / Software	Purpose
Figma	UI/UX design, wireframes, and mockups of the system.
Jira	Project management and task tracking for development workflow.
PlantUML	Creating class diagrams, sequence diagrams, and system architecture diagrams.
VS Code	Development environment for coding frontend and backend.
MySQL Workbench	Database design and queries for system data.
Socket.IO	Real-time messaging implementation.
SSLCommerz	Payment gateway for wallet top-ups and escrow.
Canva / Adobe Illustrator	Designing illustrations, icons, and visual elements for reports and mockups.

Appendix C – References to Diagrams

Diagram	Purpose
Level 1 DFD	Represents major processes and data flow across the system.
Use Case Diagram	Illustrates interactions between actors and the system.
Class Diagram	Shows main entities, attributes, methods, and relationships.
Sequence Diagrams	Describes step-by-step interactions for registration, login, job posting, messaging, escrow, and campus posts.