

---

# Software Requirements Specification

for

## IITKart

Version 1.01

Prepared by

### Group 5:

Hardik Tiwari	230437
Pranjali Deshpande	240336
Mahi Mittal	240608
Mayukh Chowdhury	240642
Palak Bandhu	240719
Praveen	240790
Shreshthraj Bhidodiya	240996
Sneha Kumari	241024
Virendra Kala	241172
Yash Dabi	241195

### Group Name: Thunderbolts

[hardikt23@iitk.ac.in](mailto:hardikt23@iitk.ac.in)  
[deshpandep24@iitk.ac.in](mailto:deshpandep24@iitk.ac.in)  
[mahii24@iitk.ac.in](mailto:mahii24@iitk.ac.in)  
[cmayukh24@iitk.ac.in](mailto:cmayukh24@iitk.ac.in)  
[palakb24@iitk.ac.in](mailto:palakb24@iitk.ac.in)  
[praveenas24@iitk.ac.in](mailto:praveenas24@iitk.ac.in)  
[sbhidodiya24@iitk.ac.in](mailto:sbhidodiya24@iitk.ac.in)  
[snehak24@iitk.ac.in](mailto:snehak24@iitk.ac.in)  
[virendrak24@iitk.ac.in](mailto:virendrak24@iitk.ac.in)  
[yashd24@iitk.ac.in](mailto:yashd24@iitk.ac.in)

Course: CS253

Mentor TA: George T L

Date: 24/01/2026

**Course:** CS253

## Table of Contents

<b>CONTENTS.....</b>	<b>II</b>
<b>REVISES.....</b>	<b>II</b>
<b>1 INTRODUCTION.....</b>	<b>4</b>
1.1 PRODUCT SCOPE.....	4
1.2 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	4
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	5
1.4 DOCUMENT CONVENTIONS.....	6
1.5 REFERENCES AND ACKNOWLEDGMENTS.....	6
<b>2 OVERALL DESCRIPTION.....</b>	<b>7</b>
2.1 PRODUCT OVERVIEW.....	7
2.2 PRODUCT FUNCTIONALITY.....	8
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	8
2.4 ASSUMPTIONS AND DEPENDENCIES.....	9
<b>3 SPECIFIC REQUIREMENTS.....</b>	<b>10</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	10
3.2 FUNCTIONAL REQUIREMENTS.....	23
3.3 USE CASE MODEL.....	27
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS.....</b>	<b>47</b>
4.1 PERFORMANCE REQUIREMENTS.....	47
4.2 SAFETY AND SECURITY REQUIREMENTS.....	47
4.3 SOFTWARE QUALITY ATTRIBUTES.....	48
<b>APPENDIX A – DATA DICTIONARY.....</b>	<b>49</b>
<b>APPENDIX B - GROUP LOG.....</b>	<b>55</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.00	Thunderbolts	The initial draft was prepared,	22/01/26
1.01	Thunderbolts	The draft was reviewed and final changes were made, along with formatting optimization.	24/01/26

## 1. Introduction

### 1.1. Product Scope

The campus community often faces challenges in accessing essential services such as food, stationery, and groceries due to the absence of a structured delivery system. Our project aims to address this gap by bringing all on-campus service providers onto a single, centralised delivery platform, allowing consumers to conveniently discover and access a wide range of goods and services.

Our primary goal is to provide a user-friendly interface for consumers, vendors, and delivery partners to streamline daily campus logistics. The website comprises three independent interfaces for each stakeholder, enabling users to browse services using category-based filters, receive recommendations based on prior activity, and provide feedback through a rating system. Users are further incentivised through a loyalty mechanism that rewards them with coins upon placing a certain number of orders. Vendors can list the services and their respective delivery requirements on their interface, where they can track and analyse overall metrics and order history. An integrated vendor-rider matching system would facilitate efficient coordination between vendors seeking delivery personnel and riders looking for opportunities. Delivery partners are provided with a dedicated interface to view active delivery requests, manage ongoing orders, and track performance metrics such as earnings and delivery history.

### 1.2. Intended Audience and Document Overview

This document is intended to provide interested stakeholders with a detailed overview of the proposed model for the IITKart platform. It outlines the user requirements identified by the team through discussions and analysis of the campus delivery ecosystem at IIT Kanpur, involving students, vendors, and delivery partners. The document further presents the corresponding system specifications designed to address these requirements and ensure efficient, reliable, and user-centric service delivery.

1. **Introduction** : This section outlines the product scope and provides foundational information such as document conventions, abbreviations, and definitions. While experienced readers may opt to skip this section, it serves as a useful resource to resolve potential uncertainties encountered while reviewing the document.
2. **Overall Description** : This section offers a high-level summary of the software, covering its primary objectives, functionality, and assumptions. It is recommended to read this section to gain an initial understanding of the system's purpose and context before diving into the details.
3. **Specific Requirements** : This section presents a comprehensive breakdown of the software's features and requirements, illustrated with supporting diagrams. It is a critical resource for developers during the implementation phase and provides end-users with a clear understanding of the system's capabilities.
4. **Other Non-Functional Requirements** : This section focuses on the non-functional requirements of the software, detailing performance, security, and usability standards.

Developers will find this section particularly helpful for ensuring the system meets its operational and quality expectations.

5. **Other Requirements:** This section specifies additional system requirements considering database management, legal and compliance constraints, system assumptions, and reusability or extensibility objectives to ensure completeness of the specification.
6. **Appendices :** This section includes additional resources, such as appendices, to support readers with supplementary information, references, or data relevant to the project.

### 1.3.Definitions, Acronyms and Abbreviations

- **Admin (Administrator):** A privileged user responsible for managing users, vendors, riders, and overall system governance.
- **API (Application Programming Interface):** A set of protocols and tools allowing different software components to communicate with each other.
- **Consumer:** The end-user (student, faculty, or staff) registered on the platform to browse products and place orders.
- **GPS (Global Positioning System):** A satellite-based navigation system used to track the real-time location of delivery partners.
- **GSTIN (Goods and Services Tax Identification Number):** A 15-digit unique identification number to identify and verify businesses registered under the Goods and Services Tax (GST) framework.
- **HTTPS (Hypertext Transfer Protocol Secure):** An extension of HTTP used for secure communication over a computer network, crucial for protecting user data during transmission.
- **KPI (Key Performance Indicator):** Quantifiable metrics (e.g., Total Earnings, Order Volume) used to evaluate the success and performance of Vendors and Riders.
- **MFA (Multi-Factor Authentication):** An electronic authentication method in which a user is granted access only after successfully presenting two or more pieces of evidence.
- **OTP (One Time Password):** A system-generated code valid within a small timeframe for authentication of the user during registration or password reset.
- **PAN (Permanent Account Number):** A unique alphanumeric identification number issued by the Income Tax Department of India, required for Vendor and Rider verification.
- **RBAC (Role-Based Access Control):** A restriction method that regulates access to computer or network resources based on the roles of individual users (e.g., preventing a Rider from accessing Admin settings).
- **SRS (Software Requirements Specification):** A document that describes the nature of a project, software, or application.
- **UI (User Interface):** The visual elements of the application through which users interact with the system (e.g., dashboards, menus, forms).
- **UPI (Unified Payments Interface):** An instant real-time payment system developed by the National Payments Corporation of India, used for facilitating inter-bank peer-to-peer and person-to-merchant transactions.

## 1.4.Document Conventions

- Arial font size 11 has been maintained throughout the text.
- Arial font sizes of 14 and 18 are used for subheadings and headings, respectively.
- The headings and subheadings of all sections are written in **bold**.
- The document text is single-spaced
- 1" margin has been maintained throughout the document.
- *Italics* have been used for writing comments.
- Bullet point ordering has been used as a listing typesetting tool.

## 1.5.References and Acknowledgments

### References:

- Style Guide (for this document): [IEEE Software Requirements Specification Template](#).
- [Figma](#) was used for designing the UI and flowcharts.
- [Freeform](#) was used for designing the Use Case Models.

### Acknowledgements:

The fundamental principles of our system design were established through the lectures and mentorship provided by Professor Dr. Indranil Saha. Alongside, our mentor TA, George TL provided valuable guidance and support throughout the development process.

## 2. Overall Description

### 2.1. Product Overview

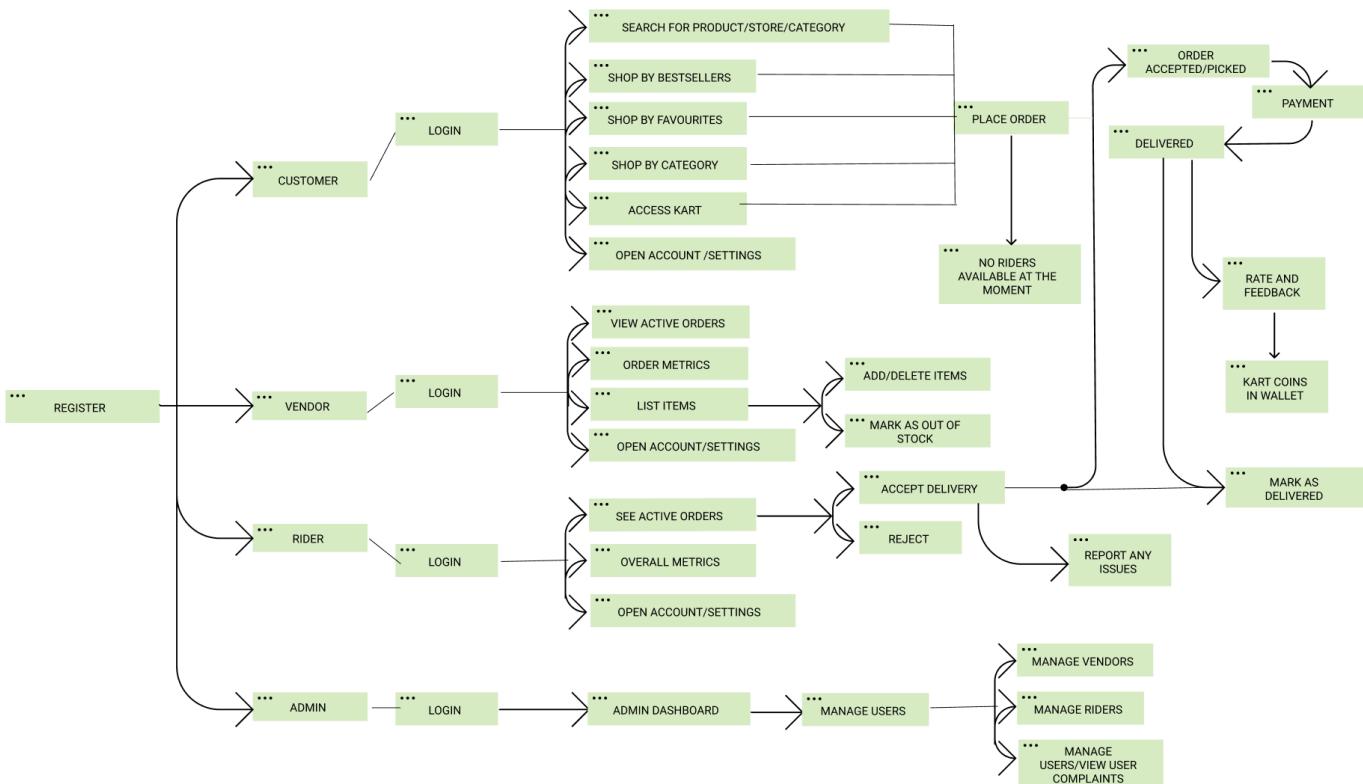
IITKart is aimed to serve as a one-stop online delivery management platform for the IIT Kanpur campus community. A self-contained software system, it is designed to streamline the end-to-end ordering and delivery of on-campus essential products.

The platform caters to three classes of users as described below -

- Campus residents (students, faculty, staff, etc): an efficient alternative to manual shopping. Centralization eliminates the need to visit multiple stores for different products, especially in emergent situations.
- Campus vendors: Enables organized product listings, enhancing accessibility and reach across the campus.
- Delivery partners: Provides a reliable opportunity to earn through structured delivery tasks.

In addition to these three categories of users, we will have one more interface for the Administrator(admin) who will manage the overall control , moderation and governance of the system.

The architecture and design of the system allow it to function as a centralized platform for smooth and reliable campus-wide e-commerce management.



## 2.2. Product Functionality

The platform offers the following functionalities:

- **Centralized delivery platform** for students, faculty, and staff that allows users to browse various products and services offered by vendors across the campus, place orders, and track their status.
- **Search and filter functionality** that allows consumers to search for services based on categories, availability, and relevance.
- **Recommendation mechanism** that suggests relevant services to consumers based on their activity and order history.
- **User Feedback and Loyalty System** that allows consumers to rate services, provide feedback based on their experience, and earn rewards to incentivize repeated usage.
- **User Account Management** that allows consumers to view order history and manage account information, including profile details (phone number, profile picture, email address), saved addresses, wallet settings, account logout, and account deletion.
- **Vendor Service and Logistics Management** that allows vendors to list available services under different categories, update availability, and specify delivery requirements.
- **Vendor-Delivery Partner Matching Mechanism** that facilitates matching between vendors and delivery partners based on delivery requirements and availability, allowing respective parties to view, accept, and manage delivery requests.
- **Performance and Earnings Overview** that allows vendors and delivery partners to view summary metrics such as cumulative earnings and activity-related statistics for accountability and analysis.

## 2.3. Design and Implementation Constraints

### Hardware Requirements:

- A functional desktop, laptop, tablet, or mobile device with a stable internet connection is required to access the IITKart web platform.
- End-user devices should have sufficient processing capability and memory to support modern web browsers and basic client-side rendering. For mobile devices, a minimum of **2 GB RAM** is recommended to ensure smooth interaction with the platform.
- Vendors and delivery partners will require continuous internet connectivity during active order handling to ensure real-time updates and notifications.

### Software Requirements:

- The IITKart platform is designed as a web-based application and is compatible with devices running operating systems that support modern web browsers.

- The application supports commonly used browsers including **Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera**, and performs optimally on their latest stable versions.
- JavaScript and **cookies** must be enabled in the browser to support dynamic content rendering, session management, and secure authentication.

## 2.4.Assumptions and Dependencies

### Assumptions:

- All users have access to internet-enabled devices (smartphones, tablets, or computers) for accessing the platform.
- Campus vendors are willing to participate and maintain updated product inventories on the platform.
- Delivery partners have their own vehicles (bicycles/bikes) and are available during operational hours .
- Users have email addresses for registration and authentication.
- The application will not handle more than 500 concurrent users during peak hours initially.
- Campus network infrastructure is stable and provides adequate internet connectivity across all halls and vendor locations.
- Payment integrations (UPI, Card, Wallet) are available and functional for transaction processing.
- Vendors have the capability to prepare orders within 15-30 minutes of acceptance.
- Delivery partners can complete deliveries within campus in 15-20 minutes on average.
- The platform operates within IIT Kanpur campus boundaries only.
- Users provide accurate delivery addresses (Hall number, Room number).
- The Kart Coins loyalty system is managed internally without external financial regulations.
- Admin accounts have full system access and cannot be deleted or compromised.
- Vendors and delivery partners undergo verification before account activation.

### Dependencies:

- **React** - JavaScript library for building user interfaces and component-based architecture
- **TypeScript** - Typed superset of JavaScript for enhanced code quality and development experience
- **Vite** - Fast build tool and development server for modern web applications
- **Tailwind CSS v4** - Utility-first CSS framework for styling responsive user interfaces
- **React Router** - Client-side routing library for navigation between different interfaces
- **Lucide React** - Icon library for consistent UI iconography across the platform
- **Shadcn UI Components** - Pre-built accessible component library for rapid UI development
- **ORM (Object-Relational Mapper)**: The system uses **Prisma** to interface between the Node.js backend and the database, ensuring type-safe database queries and automated schema migrations.
- **Sonner** - Toast notification library for user feedback and alerts
- **Recharts** - Charting library for analytics and data visualization dashboards
- **Browser Support** - Modern browsers (Chrome, Firefox, Safari, Edge) with ES6+ support
- **Node.js Runtime** - JavaScript runtime environment for development and build processes
- **Git** - Version control system for project management and collaboration
- **Package Manager (pnpm/npm)** - Dependency management and package installation
- **Web APIs** - Local Storage for client-side data persistence, Fetch API for future backend integration
- **MongoDB/PostgreSQL** - Database for backend data storage
- **Express.js/Node.js** - Backend server framework
- **JWT** - Authentication and authorization tokens

## Potential Future Dependencies:

- **Payment Gateway APIs** - For UPI, Card, and Wallet integrations
- **SMS/Email APIs** - For order notifications and OTP verification
- **Google Maps API** - For location tracking and delivery route optimization

## 3. Specific Requirements

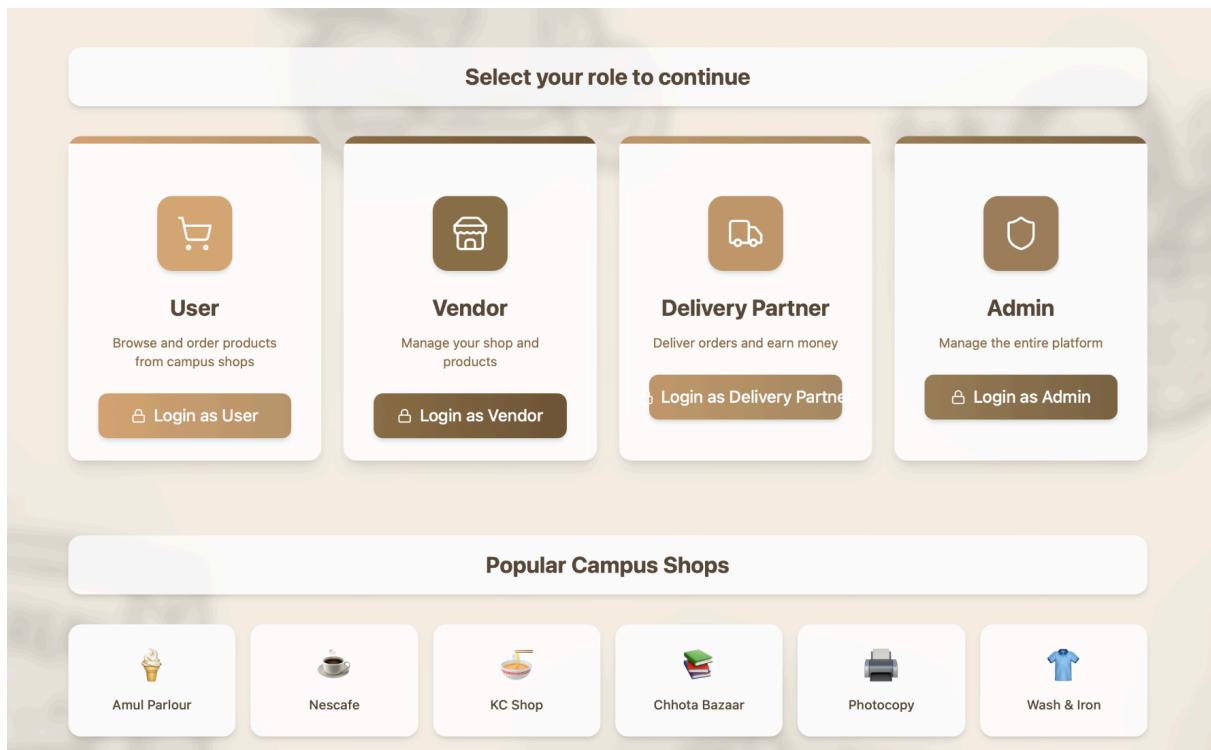
### 3.1.External Interface Requirements

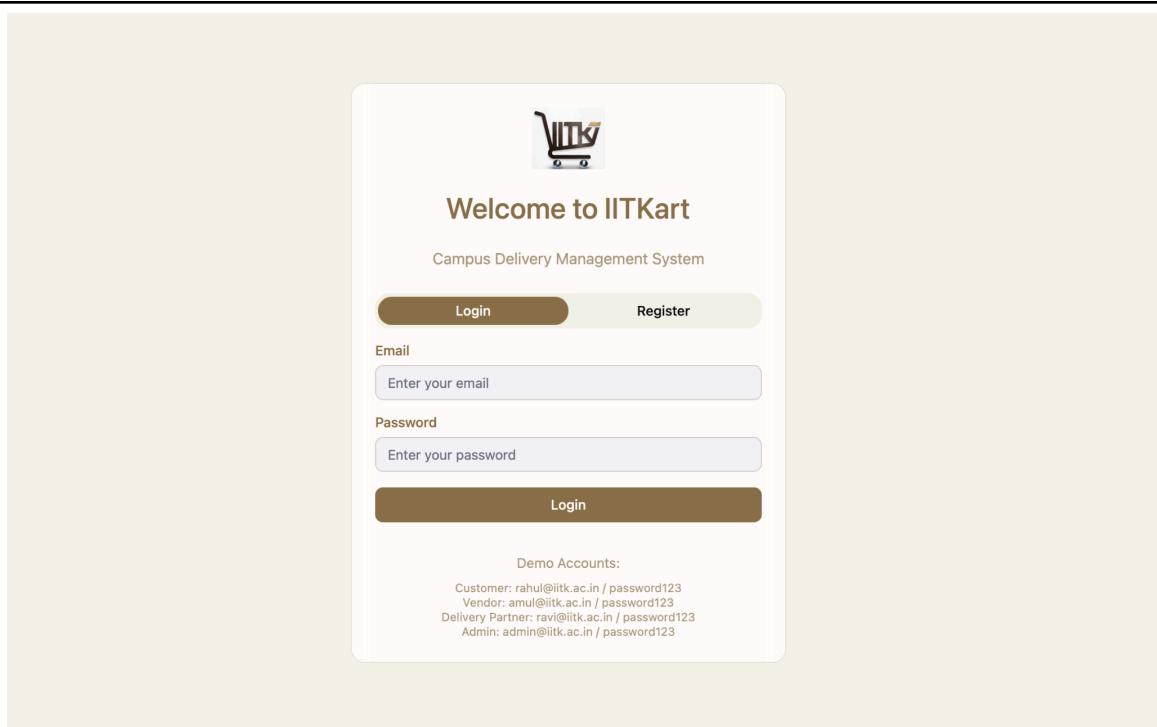
#### 3.1.1.User Interfaces

IITKart is a web-based on campus delivery system developed for the IIT Kanpur campus. It allows students, shop owners, delivery personnel, and administrators to use a single platform for all delivery-related activities. Depending on the type of account a user has, the system shows different pages and features for placing orders, managing shops, delivering orders, or handling overall system control.

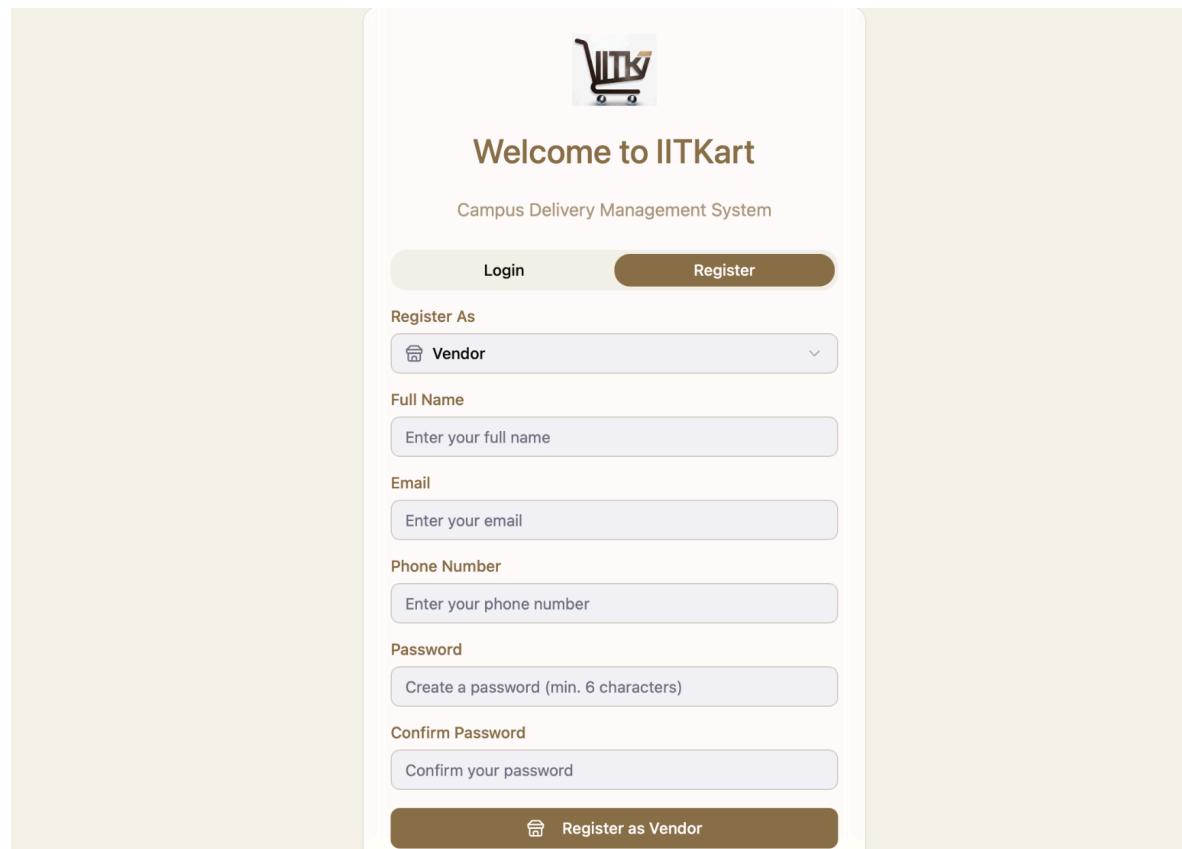
- **The Login Page**

The login page provides a simple and secure authentication interface for all users of the system. Users can register by entering their mobile number and email ID, followed by OTP verification and password setup. During registration, users can select their role as User, Vendor, Courier, or Admin. Once registered, users can log in using their email or mobile number and password. After successful authentication, the system redirects users to their respective dashboards based on their selected role.





- **The Registration Page**



## User Module

- **Home Page**

The screenshot shows the IITKart home page. At the top, there is a navigation bar with links for Browse, My Orders, Wallet, and Settings. Below the navigation bar is a search bar labeled "Search products..." and a "Cart" button. There are three dropdown menus: "All Categories", "All Shops", and "All Products". A location indicator shows "Hall 2, Room 201".

A food listing from "Amul Parlour" is displayed, rated 4.5 stars. The listing includes:

- Veg Sandwich**: Fresh veg sandwich with cheese, ₹50. Includes a "Review" button.
- Cold Coffee**: Chilled cold coffee, ₹40. Includes a "Review" button.
- Ice Cream**: Amul ice cream 100ml, ₹60. Includes a "Review" button.
- Cheese Burger**: Delicious cheese burger, ₹80. Includes a "Review" button.
- Milkshake**: Thick milkshake, ₹70. Includes a "Review" button.

- **Order History Page**

The screenshot shows the "My Orders" section of the IITKart app. It displays two completed orders and one pending order.

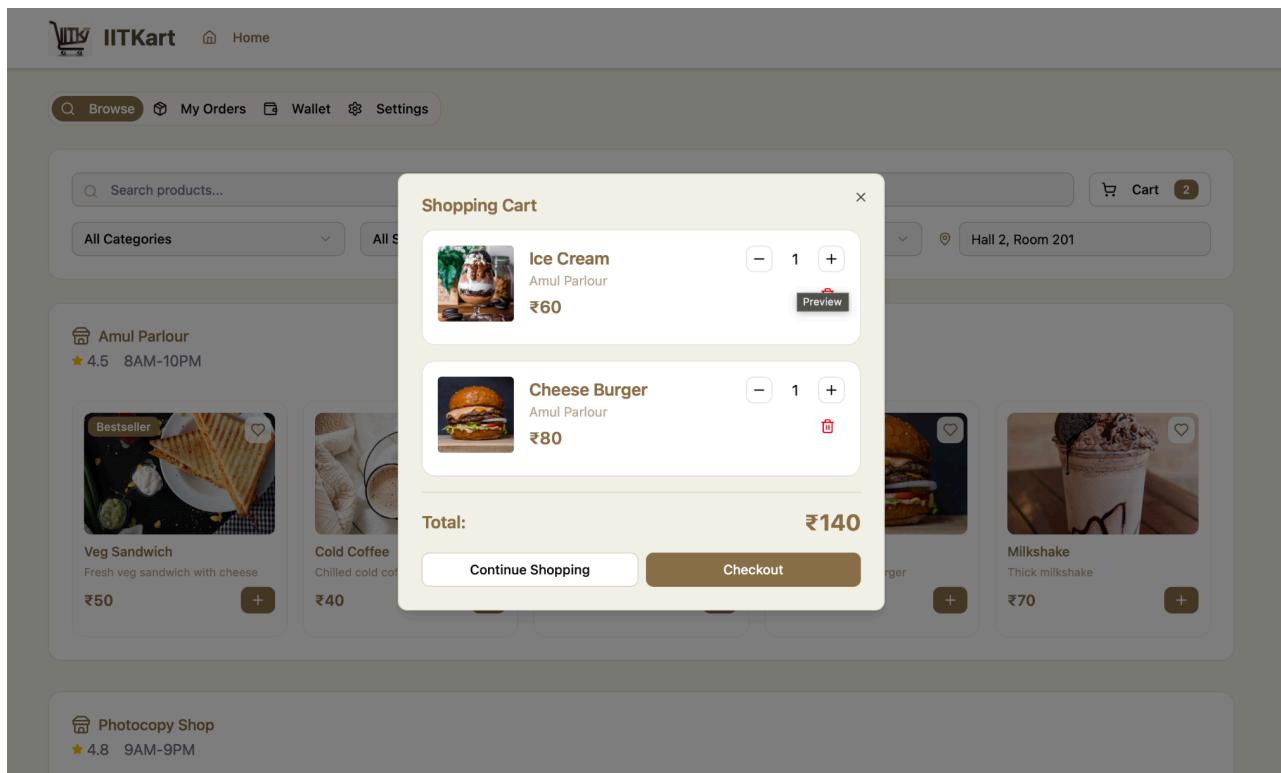
**Completed Orders:**

- Order #ORD001**: Placed (Order confirmed) → Accepted (Shop confirmed) → On the way (Out for delivery) → Delivered (Order complete). Total amount: ₹150, +15 coins. Status: delivered. Message: "Order delivered successfully! Hope you enjoyed your purchase".
- Order #ORD005**: Placed (Order confirmed) → Accepted (Shop confirmed) → On the way (Out for delivery). Status: pending. Message: "Your order has been placed and is awaiting shop confirmation".

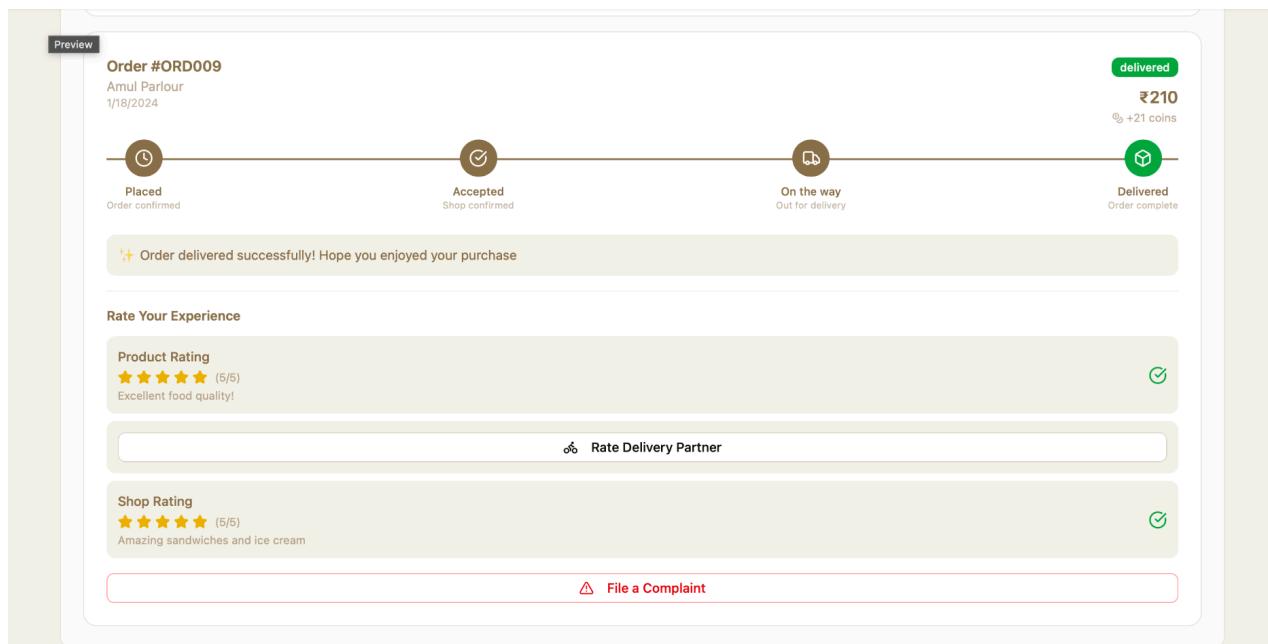
**Pending Order:**

- Order #ORD005**: Placed (Order confirmed) → Accepted (Shop confirmed) → On the way (Out for delivery). Status: pending. Message: "Your order has been placed and is awaiting shop confirmation".

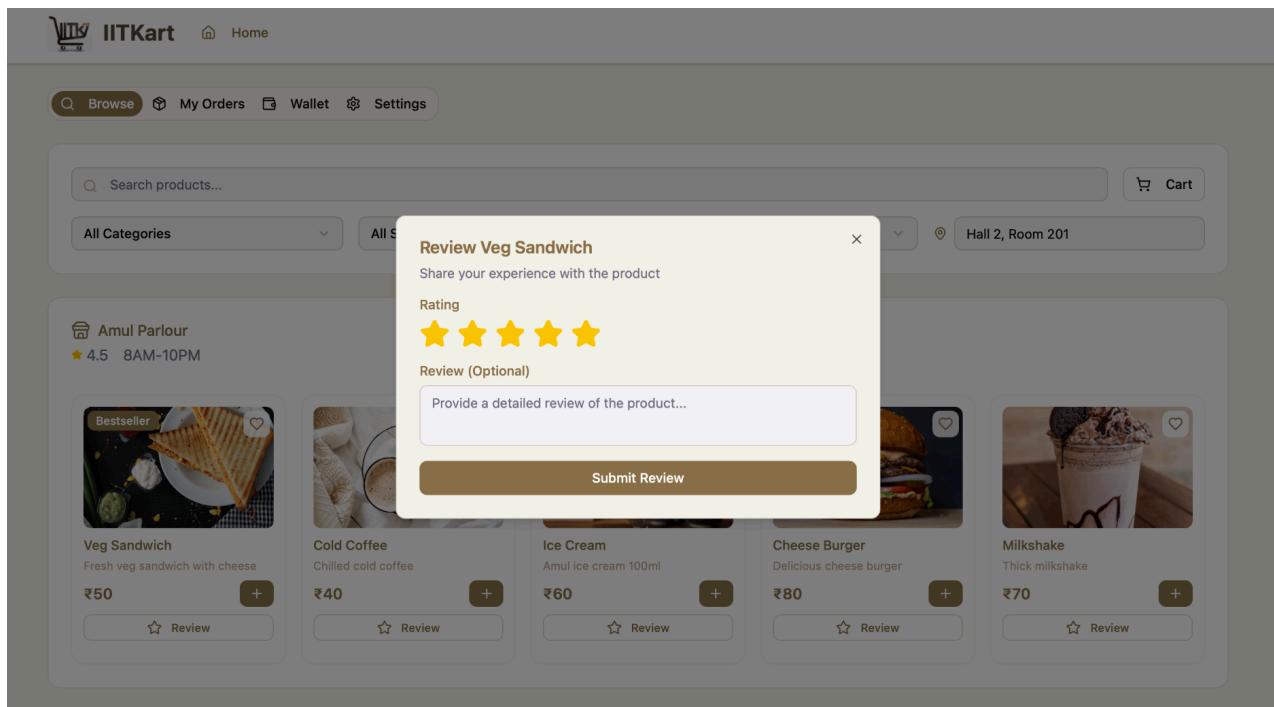
- Kart



- Complaint and Rate delivery Partner option



- **Review Order**



## Vendor Module

- **Vendor Dashboard**

The screenshot shows the vendor dashboard for "Amul Parlour". At the top, there is a header with the logo and "Home" link. Below the header, the dashboard title is "Amul Parlour" and "Vendor Dashboard". There are four summary cards: "Total Earnings ₹45600" with a dollar sign icon, "Total Orders 156" with a shopping cart icon, "Average Rating 4.5★" with a star icon, and "Active Products 5" with a box icon. Below these cards is a navigation bar with tabs: "Orders" (selected), "Inventory", "Reviews", and "Settings". The main content area is divided into two sections: "Active Orders" and "Order History". The "Active Orders" section shows "5 pending orders" and details for "Order #ORD004" (accepted, 1/20/2024, 9:00:00 AM, 1 items - ₹90, Hall 7, Room 410). It includes a "Delivery Partner Details" card for "Suresh Rider" (Name, Contact Number 9876543231, Experience 1 year, Total Deliveries). The "Order History" section shows "2 completed orders" with details for "Order #ORD001" (Delivered, 1/19/2024, 2 items - ₹150) and "Order #ORD009" (Delivered, 1/18/2024, 2 items - ₹210).

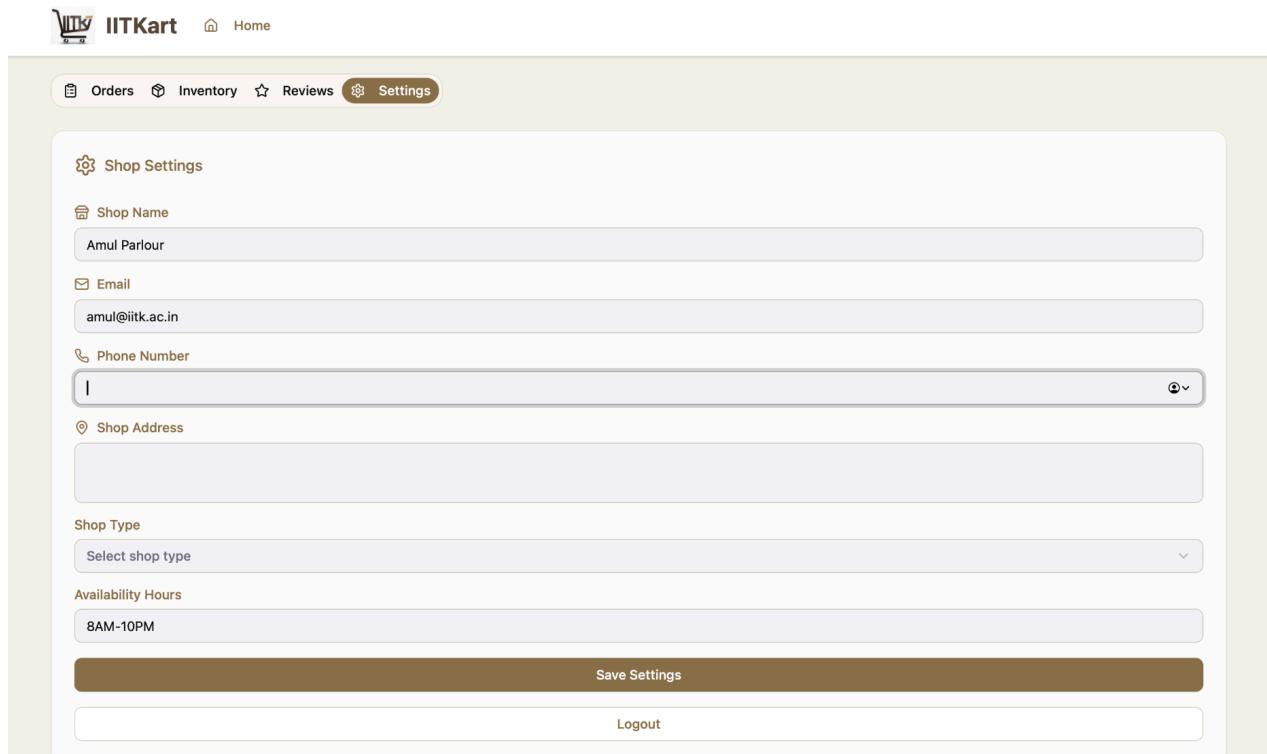
- Inventory( Products Management Page)

The screenshot shows the IITKart Vendor Dashboard for 'Amul Parlour'. At the top, there are four summary cards: Total Earnings (₹45600), Total Orders (156), Average Rating (4.5★), and Active Products (5). Below these are navigation tabs: Orders, Inventory (which is selected and highlighted in brown), Reviews, and Settings. The main section is titled 'Product Inventory' and displays three products: 'Veg Sandwich' (Food, ₹50, In Stock), 'Cold Coffee' (Beverage, ₹40, In Stock), and 'Ice Cream' (Food, ₹60, In Stock). Each product card includes an 'Edit' button (with a pencil icon) and a 'Delete' button (with a trash bin icon).

- Review and Feedback Page

The screenshot shows the IITKart Vendor Dashboard for 'Amul Parlour'. It features the same top summary cards and navigation tabs as the previous page. The main section is titled 'Customer Reviews' and displays a summary: Average Rating (4.0★) based on 1 review. Below this, a specific review is shown for 'Order #ORD001' from 1/19/2024, which states 'Food was good' and has a rating of 4/5.

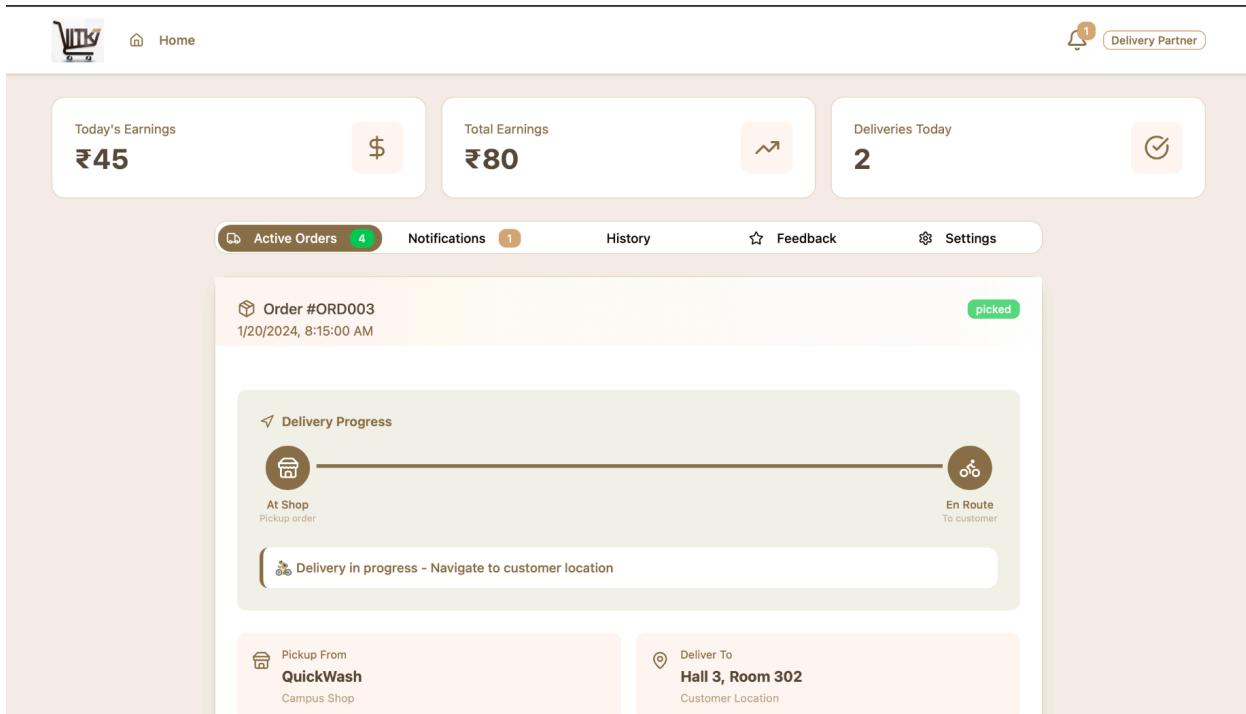
- **Settings Page**



The screenshot shows the 'Shop Settings' section of the IITKart Settings page. It includes fields for Shop Name (Amul Parlour), Email (amul@iitk.ac.in), Phone Number (empty), Shop Address (empty), Shop Type (Select shop type dropdown), Availability Hours (8AM-10PM), and a Save Settings button.

## Rider(Courier) Module

- **Rider Dashboard**



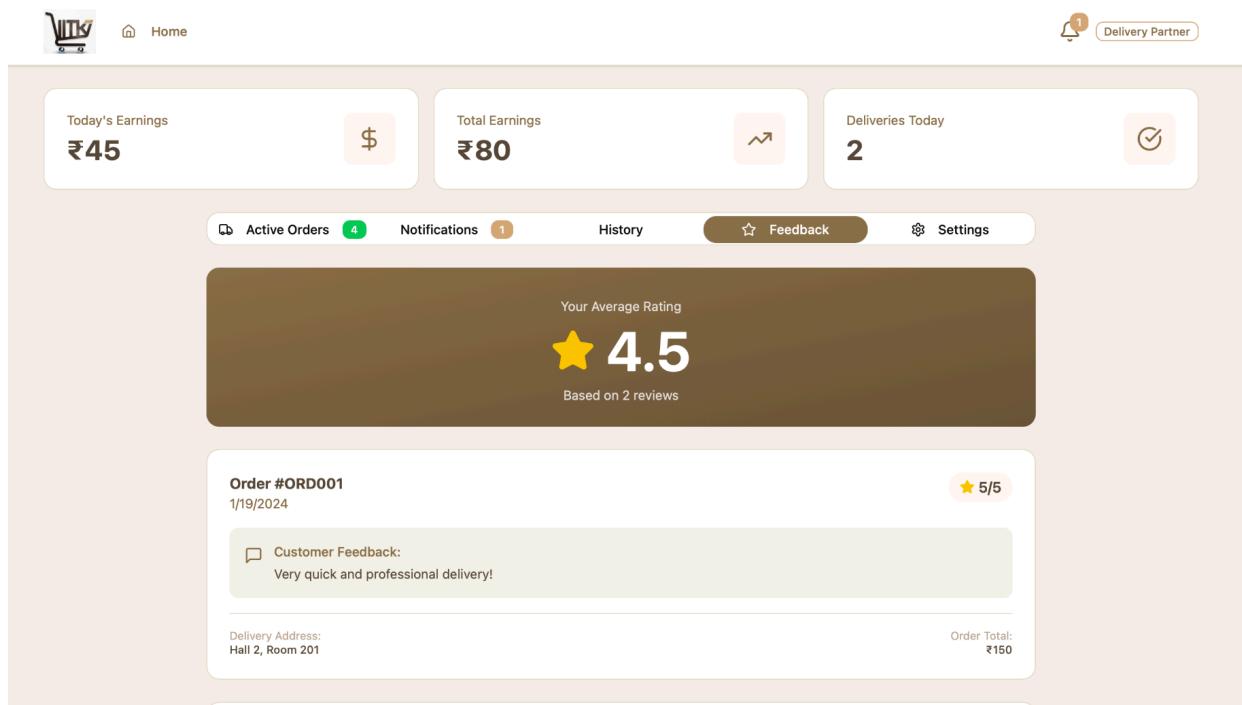
The Rider Dashboard displays key metrics: Today's Earnings (₹45), Total Earnings (₹80), and Deliveries Today (2). It also shows a bell icon with 1 notification and a 'Delivery Partner' status. The main area shows Order #ORD003 details (1/20/2024, 8:15:00 AM) with a green 'picked' status. The 'Delivery Progress' section shows the rider moving from 'At Shop' (Pickup order) to 'En Route' (To customer). Below this, it shows 'Delivery in progress - Navigate to customer location'. At the bottom, it lists 'Pickup From QuickWash Campus Shop' and 'Deliver To Hall 3, Room 302 Customer Location'.

- Order History Page

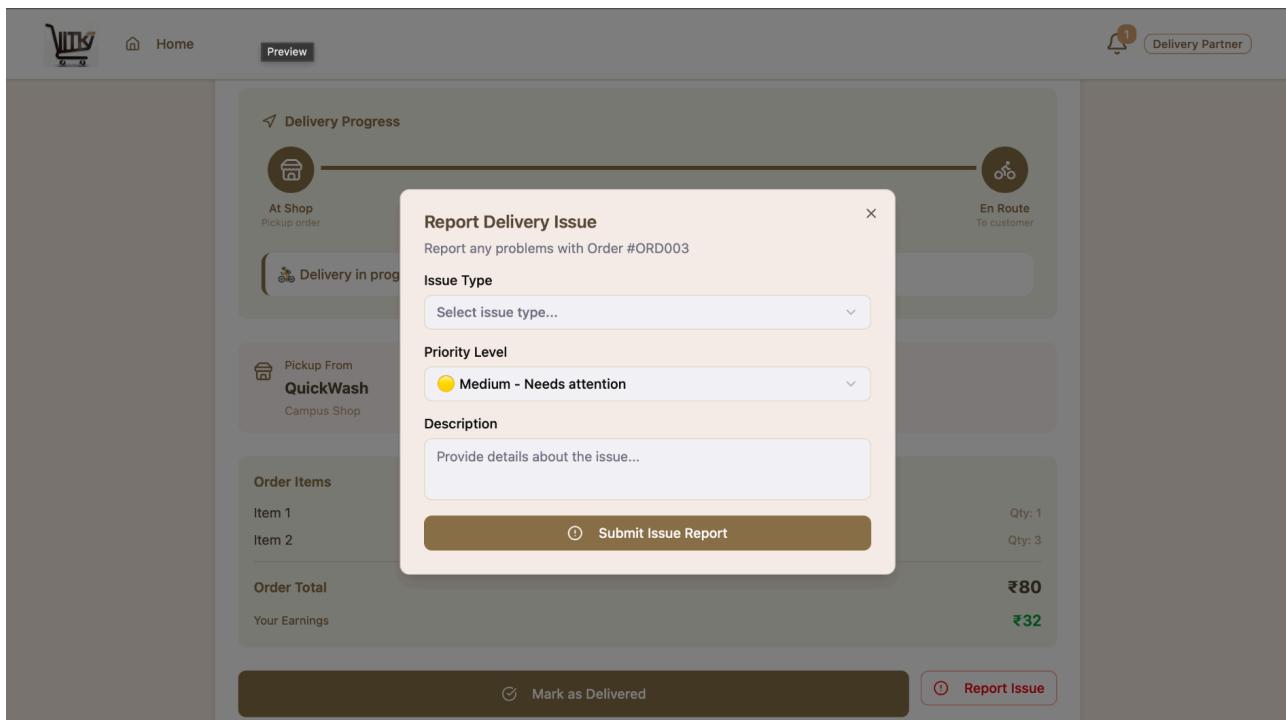
The screenshot displays the Order History Page of a delivery application. At the top, there are three summary boxes: 'Today's Earnings ₹45' with a dollar sign icon, 'Total Earnings ₹80' with an upward arrow icon, and 'Deliveries Today 2' with a checkmark icon. Below these are navigation tabs: Active Orders (4), Notifications (1), History (selected), Feedback, and Settings. The main content area shows two completed orders:

- Order #ORD001** (Completed) - Delivered 2 hours ago from Campus Cafe to Hall 2, Room 201. Distance: 1.2 km, Items: 3. Earnings: ₹45.
- Order #ORD002** (Completed) - Delivered 5 hours ago from Stationery Store to Hall 4, Room 105. Distance: 0.8 km, Items: 2. Earnings: ₹35.

- Reviews and Feedback Page



- Report issue and Emergency



## Admin Module

- Admin Dashboard

The screenshot shows the Admin Dashboard of the IITKart platform. At the top, there are several summary cards:

- Total Orders: 10 (Today: 0)
- GMV: ₹1310 (Commission: ₹197)
- Active Users: 4
- Vendors: 6 (Riders: 2)

Below these are two performance indicators:

- Success Rate:** 30.0% (Orders delivered successfully) - Green checkmark icon
- Failure Rate:** 0.0% (Orders cancelled) - Red error icon

Two line graphs show trends over the last 7 days:

- Order Trends (Last 7 Days):** Shows a flat line at 1 order per day.
- Revenue Trends (Last 7 Days):** Shows a flat line at ₹12340 per day.

- Vendor Management Page

The screenshot shows the Vendor Management page. It lists four vendors with their details:

Vendor Name	Email	Status
Amul Parlour	amul@iitk.ac.in	active
Photocopy Shop	photocopy@iitk.ac.in	active
Wash & Iron	laundry@iitk.ac.in	active
Chhota Bazaar	bazaar@iitk.ac.in	active

Each vendor card includes the following information:

- Total Orders
- Earnings
- Rating (with star icon)
- Fulfillment Rate
- Availability
- A "Suspend Vendor" button

- User Management Page

**User Management**

User Details	Kart Coins
Rahul Kumar rahul@iitk.ac.in 9876543210  Address: Hall 2, Room 201 Orders: 3	150
Priya Singh priya@iitk.ac.in 9876543211  Address: Hall 5, Room 105 Orders: 3	220
Amit Sharma amit@iitk.ac.in 9876543212	80

- Order Status Page

**Live Order Status**

Order Details	Status
Order #ORD003 1/20/2024, 8:15:00 AM  Vendor: KC Shop Customer: Amit Sharma Total: ₹80  Delivery Address: Hall 3, Room 302  Cancel Order   Mark Delivered	picked
Order #ORD004 1/20/2024, 9:00:00 AM  Vendor: Amul Parlour Customer: Neha Gupta Total: ₹90  Delivery Address: Hall 7, Room 410  Cancel Order   Mark Delivered	accepted

- Feedback Page

The screenshot shows the Feedback Portal section of the IITKart admin dashboard. It displays three feedback entries:

- Nescafe** (Order #ORD002) - 4 stars, 1/19/2024: "Good"
- Amul Parlour** (Order #ORD001) - 5 stars, 1/19/2024: "Great service!"
- Amul Parlour** (Order #ORD009) - 5 stars, 1/18/2024: "Excellent food quality!"

- Complaints Management

The screenshot shows the Complaints Management section of the IITKart admin dashboard. It displays two open complaints:

- Delayed Delivery** (pending)  
 Complaint ID: CMP001  
 User: Rahul Kumar (rahul@iitk.ac.in)  
 Submitted: 1/21/2024, 2:00:00 PM  
 Description: My order was delayed by 2 days.  
 Buttons: Mark as Resolved Close
- Incorrect Product** (resolved)  
 Complaint ID: CMP002  
 User: Priya Singh (priya@iitk.ac.in)  
 Submitted: 1/22/2024, 10:00:00 AM  
 Description: Received a different product than ordered.

### 3.1.2.Hardware Interfaces

- **Client Devices:**

**Supported Device Types:**

- Desktop Computers (Windows, macOS, Linux)
- Laptops (Windows, macOS, Linux)
- Smartphones and Tablets (Android, iOS)

**Nature of Interaction:**

- The IITKart platform is accessed through a web browser and communicates with backend services using standard internet protocols such as **HTTP/HTTPS**.
- The user interface is designed to be **responsive**, allowing seamless interaction across devices with varying screen sizes and resolutions.
- Consumers, vendors, and delivery partners require internet-enabled devices to access their respective interfaces for browsing products, managing orders, and handling deliveries.

- **Web server:**

**Logistical Characteristics:**

- The web server hosts the IITKart application and processes client requests including authentication, order management, vendor listings, and delivery coordination.
- Communication between client devices and the server occurs through secure API calls with appropriate authentication and authorization mechanisms.

**Physical Characteristics:**

- The backend services are hosted on **cloud-based or on-premise servers** running modern operating systems such as Linux or Windows.
- Server hardware must have sufficient **CPU, memory, and storage capacity** to handle concurrent users, transaction processing, and real-time order updates.

### 3.1.3.Software Interfaces

The software consists of three major components:

- **The front end :** A web-based interface through which users, vendors, delivery partners and admin can send requests to and receive responses from the server
- **The back end :** Handles application logic, request processing, authentication, and communication with the database

- **The database :** Stores system data including user accounts, vendor listings, order information, delivery records, and transaction-related details.

## 3.2. Functional Requirements

### 3.2.1. User Registration

The users are required to provide a username, mobile number, and email address for registration. After OTP verification, the user shall set a password to complete registration and secure the account. Vendors must additionally provide shop details such as shop name, address, category, operating hours, GSTIN number, and bank details while delivery partners are required to submit vehicle details and identification proofs (Aadhaar card or PAN card).

### 3.2.2. Search/Filter by Product/Store

Consumers can go through all listed items on the websites through the Search function of IITKart. They can search products/stores based on their requirement and also can filter the options on the basis of Price, Bestsellers, Availability and Offer deals.

### 3.2.3. Shop by Categories and Favourites

The system will provide the categories section on the platform in which the products are listed on the basis of similar interests. Consumers can shop products using such categories. Consumers can also list their most liked in their Favorites list by marking them as their favourites, this will help consumers to access their liked products with ease.

### 3.2.4. Recommendation System

The system will be using a Recommendation Mechanism for the consumers. This system will help to recommend the products to the consumers on the basis of their activities, previous orders, interests. This system will also mention the Best Sellers and Offer Deals on the platform.

### 3.2.5. User Feedback

Consumers can rate the services of the platform based on their experience. Consumers can also rate their purchased products on the platform which contributes to the rating of the product/items. These ratings on the product will help to build transparency for the other consumers while browsing.

### 3.2.6. Loyalty Rewards (Kart Coins)

The system shall reward the consumers on the basis of repeated usage. The reward will be in the form of Kart Coins in the consumer's wallet. Kart coins will be rewarded to the consumers after completion of an order. After a specific amount of Kart Coins that are collected in the Wallet, Consumer can avail special discounts and offers like free deliveries.

### 3.2.7.Consumer's My Account

The system will be containing a personal page for the consumer that will be containing the details of the Consumer(like Name, Address, Email, Contact no.). This personal page will also contain the consumer's order history, Account settings and Wallet details.

## VENDOR

### 3.2.8.Vendor Dashboard

The system shall display key metrics on the dashboard, including:

- Total earnings
- Total orders
- Average customer rating
- Number of active products

The system shall allow vendors to accept or reject incoming orders.

- The system shall update the order status immediately after vendor action and notify the customer and rider.

### 3.2.9.Order Management

- The system shall display all incoming orders to the vendor in real time.
- The system shall allow vendors to view order details, including item list, quantity, total price, and delivery status.
- The system shall allow vendors to accept or reject incoming orders.
- The system shall update the order status immediately after vendor action and notify the customer and rider.

### 3.2.10.Reviews Tab

- The system shall display customer reviews for completed orders..
- The system shall display the average rating of the vendor.
- The system shall associate each review with its corresponding order.
- The system shall allow vendors to view historical feedback.

### 3.2.11.Inventory and Product Management

- The system shall allow vendors to add new products to their store listing.

- The system shall allow vendors to edit product details such as name, description, price, and availability.
- The system shall allow vendors to mark products as "out of stock."
- The system shall prevent customers from ordering out-of-stock items.
- The system shall allow vendors to remove products from listings.

### **3.2.12. Earnings and Settlement Overview**

- The system shall provide a summary of total earnings, broken down by day, week, and month.
- The system shall display settlement history including payout status.

### **3.2.13. Store Details Management**

- The system shall allow vendors to set store opening and closing timings.
- The system shall allow vendors to edit store details, including store name, description, address, contact information, and category.
- The system shall reflect updated store information immediately on the customer interface.

## **DELIVERY PARTNER**

### **3.2.14 Rider Registration and Verification**

- The system shall allow delivery partners to register by providing personal details including name, mobile number, and email address.
- The system shall perform OTP-based verification during rider registration.
- The system shall require delivery partners to submit additional details such as vehicle type and valid identification proof (Aadhaar card or PAN card).
- The system shall allow the rider to set a secure password upon successful verification.
- The system shall restrict platform access to riders whose registration is incomplete or pending approval.

### **3.2.15 Rider Dashboard**

- The system shall provide a dedicated rider dashboard displaying key metrics such as:
  - Earnings for the current day
  - Total earnings
  - Number of deliveries completed
  - Active delivery assignments
  - The system shall update rider dashboard metrics in real time.

### 3.2.16 Delivery Request and Order Details

- The system shall notify riders when new delivery requests are available.
- The system shall allow riders to view detailed information for each delivery request, including:
  - Vendor pickup location
  - Customer delivery location
  - Distance and estimated delivery time
  - Order value and rider earnings
- The system shall allow riders to accept or reject delivery requests.
- The system shall automatically reassign delivery requests if a rider rejects or does not respond within a specified time.

### 3.2.17 Active Order Management

- The system shall allow riders to view all active and ongoing delivery orders.
- The system shall display the current delivery status for each active order.
- The system shall prevent riders from accepting new delivery requests beyond system-defined delivery limits.

### 3.2.18 Delivery Completion

- The system shall allow riders to mark an order as *Delivered* upon successful handover to the customer.
- The system shall update the order status immediately after delivery confirmation.
- The system shall notify the customer and vendor once an order is marked as delivered.
- The system shall trigger the rating and feedback process for completed deliveries.

### 3.2.19 Rider Performance and Earnings

- The system shall allow riders to view their performance metrics, including:
  - Total deliveries completed
  - Average rating
  - Delivery history
- The system shall allow riders to view detailed earnings and settlement history.
- The system shall update rider earnings automatically after successful order completion.

### 3.2.20 Rider Account Management

- The system shall allow riders to view and update their personal profile information.

- The system shall allow riders to update vehicle details and contact information.
- The system shall provide account settings including password management.

### **3.2.21 Issue Reporting and Support**

- The system shall allow riders to report delivery-related issues during an active order.
- The system shall log reported issues with associated order details.
- The system shall notify the admin when a rider reports an issue requiring intervention.
- The system shall temporarily restrict order completion until the reported issue is resolved, if applicable.

### **3.2.22 Rider Feedback Visibility**

- The system shall allow riders to view ratings and feedback received from customers.
- The system shall associate each feedback entry with its corresponding order.
- The system shall compute and display an average rider rating.

## **ADMIN**

### **3.2.23 Admin Dashboard and System Monitoring**

- The system shall provide an Admin Dashboard displaying overall platform metrics such as total orders, active users, vendors, and riders.
- The system shall allow the admin to view real-time status of all active and completed orders.
- The system shall allow the admin to monitor system activity for smooth platform operation.

### **3.2.24 Vendor and Rider Management**

- The system shall allow the admin to view all registered vendors and delivery partners.
- The system shall allow the admin to approve, suspend, or remove vendors and riders.
- The system shall allow the admin to view performance details of vendors and riders, including order completion rates and complaints.

### **3.2.25 User Management and Complaints Handling**

The system shall allow the admin to view all registered users and their order history.

- The system shall allow the admin to handle user complaints and feedback.
- The system shall allow the admin to take corrective actions such as warning or suspending users, vendors, or riders when required.

### **3.2.26 Accounts and Financial Overview**

- The system shall allow the admin to view invoices and receipts generated for completed orders.
- The system shall allow the admin to track payments, settlements, and pending payouts.

- The system shall allow the admin to generate basic financial reports for administrative purposes.

## **PAYMENT**

### **3.2.27.Order Payment Processing**

- The system shall allow customers to make online payments for placed orders.
- The system shall support payment methods such as UPI, net banking, and cards through an external payment gateway.
- The system shall confirm the order only after successful payment verification.

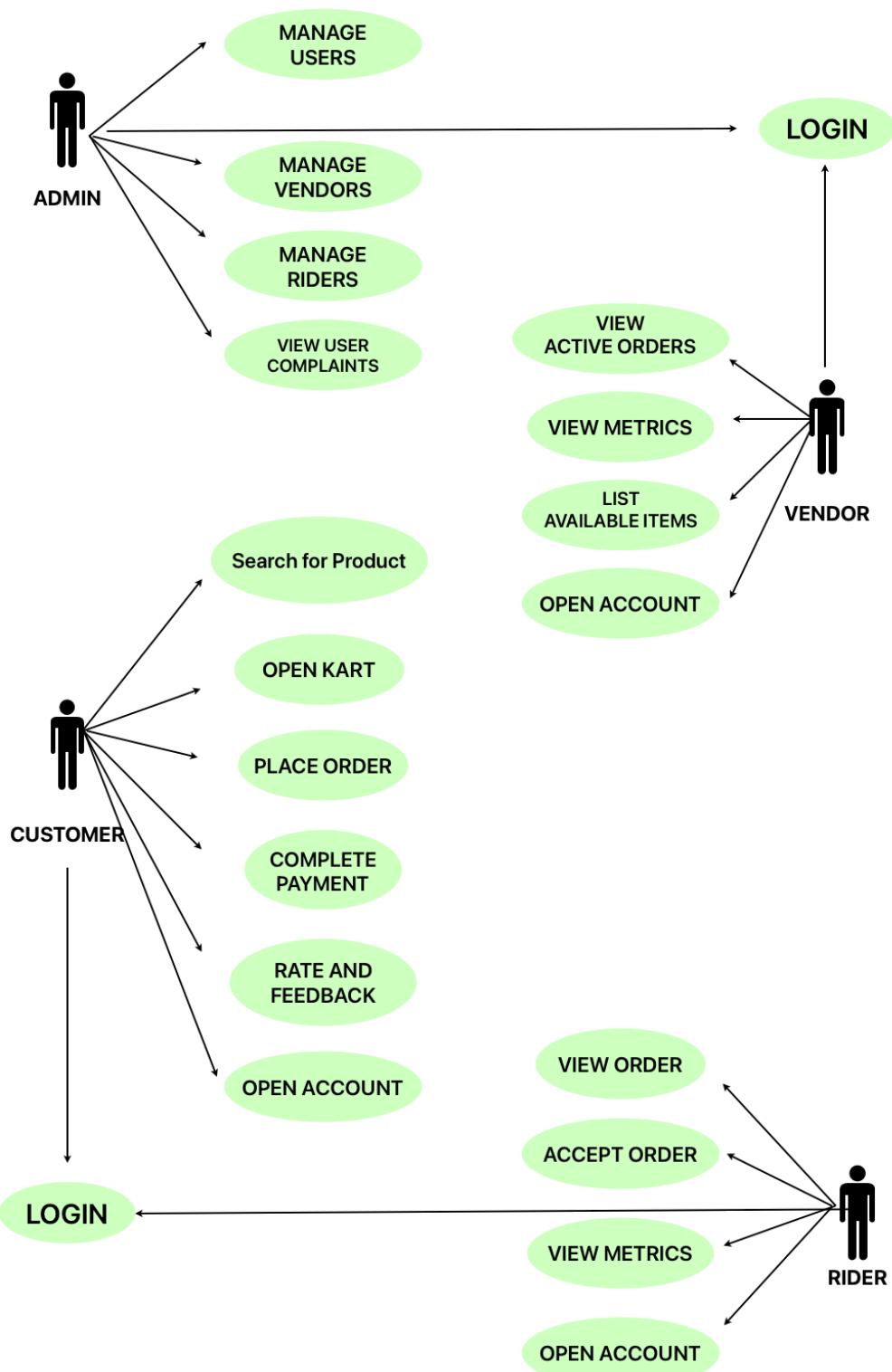
### **3.2.28.Payment Verification and Status Update**

- The system shall verify payment status received from the payment gateway.
- The system shall update the order status upon successful or failed payment.
- The system shall notify the customer regarding payment success or failure.

### **3.2.29.Refund and Transaction Records**

- The system shall support refunds for cancelled or failed orders.
- The system shall maintain records of all payment transactions.
- The system shall allow users and admin to view payment receipts and transaction history.

### 3.3. Use Case Model



### 3.3.1. U1-Customer Searches for product

**Author** – Mayukh

**Purpose** - The customer searches for whatever product he wishes to purchase. He/she can search on the basis of either products , shops, category , etc. The system may also display recommendations based on previous search and order history to assist the customer in discovering relevant items.

**Requirements Traceability** – Users shall be able to browse products and vendors.

**Priority** - High. This is a core functionality of the system , without an effective search feature customers cannot browse or discover products and stores.

**Preconditions** - Customers must be logged into the system and at least one vendor must have products listed for selling.

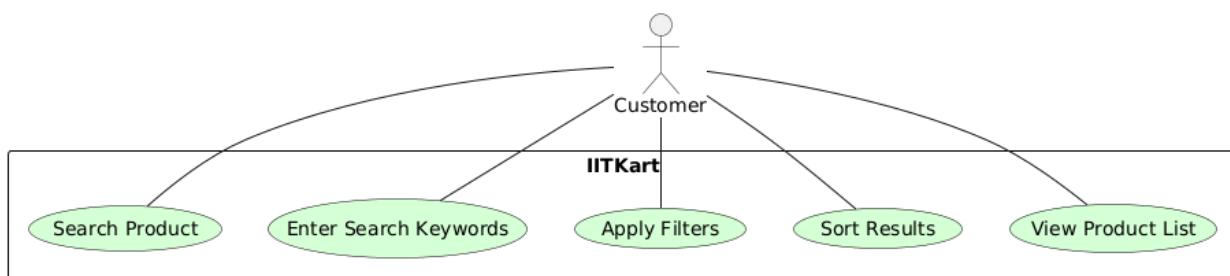
**Post conditions** - A list of available products will display upon search completion and customers may proceed to buy the product or add to Kart.

**Actors** – Customer

**Exceptions** - No matching products or vendors may be found.

**Includes** (other use case IDs)-None

**Notes/Issues** - Recommendation facilities need to be integrated using Machine Learning based techniques.



### 3.3.2. U2- Customer places an Order

**Author** – Mayukh

**Purpose** - The customer having searched/found the product he/she wants , now places an order for the product(s) through the IITKart platform. They may review items added/present in the cart and then proceed to purchase.

**Requirements Traceability** – Users shall be able to place orders ,

**Priority** - High. This is also a core functionality as it represents the primary focus of the system.The system cannot fulfil its intended purpose without a proper purchasing process integrated.

**Preconditions** - Customers must be logged into the system , the selected item(s) must be in stock.

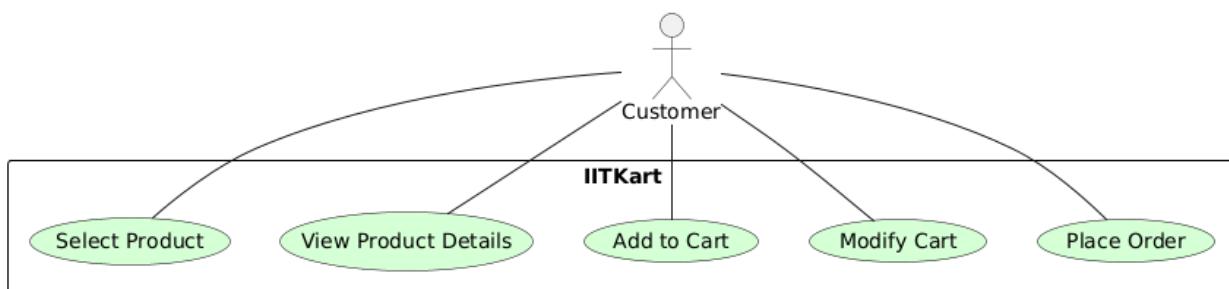
**Post conditions** - A new order is created and is now visible to the vendor.Once the customer has completed the payment ,the vendor is now responsible for handing over the order to the delivery rider who accepts the delivery assignment.

**Actors** – Customer (Vendor ,Rider)

**Exceptions** - No riders may accept the assignment . In this case the customer gets a “No riders available at the moment” status.

**Includes** (other use case IDs) -U3 :Customer completes payment

**Notes/Issues** - Future versions may include Order scheduling and cancelling followed by automatic payment refund.



### 3.3.3. U3 - Customer completes payment for order

**Author** – Mayukh

**Purpose** - Having finalised Kart items , the completion of payment makes the order placement finalised and up for further processing by vendor and rider.

**Requirements Traceability** – Users shall be able to complete payment for placed orders.

**Priority** - High. Payment completion is a critical use case since an order cannot be confirmed or processed without successful payment .

**Preconditions** - Customer must be logged in , a valid order must have been created already, and a rider must also have already accepted the delivery task.

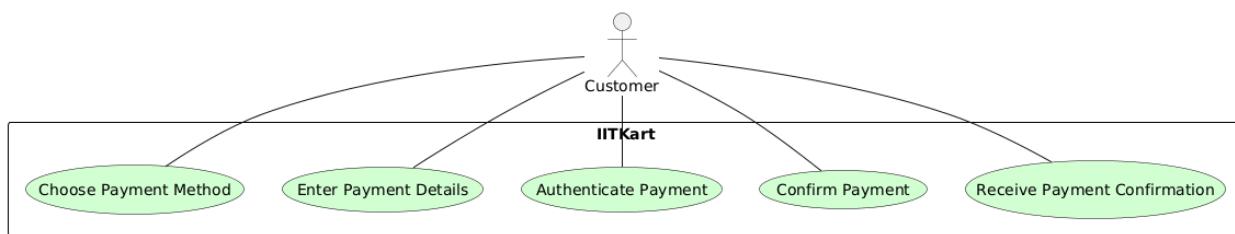
**Post conditions** - The vendor and rider get notified that the order placement is complete.A payment receipt/conformation message is displayed to the customer.

**Actors – Customer**

**Exceptions** - Payment may get declined by payment gateway( due to various reasons , for example , network connection may fail) .In that case order cannot be completed and customer gets informed.

**Includes** (other use case IDs)-None

**Notes/Issues** -Payment facility needs to be integrated in the system .Payment integration requires interaction with an external payment gateway (such as UPI, net banking, or card services), which may introduce additional security, reliability, and compliance requirements.



### 3.3.4. U4 - Customer will rate and review

**Author** – Mayukh

**Purpose** - This use case allows a customer to provide ratings and written feedback for a completed order, including both the vendor and the delivery rider. The customer may even lodge a complaint if required , which will then be handled by the Admin.

**Requirements Traceability** – Users shall be able to rate vendors and riders

**Priority** - Medium. Although this is not really a necessity , it is quite important that users be allowed to submit reviews and complaints for long term smooth functioning of the system.

**Preconditions** - Customers must be logged in to the system. The order must have been successfully completed.

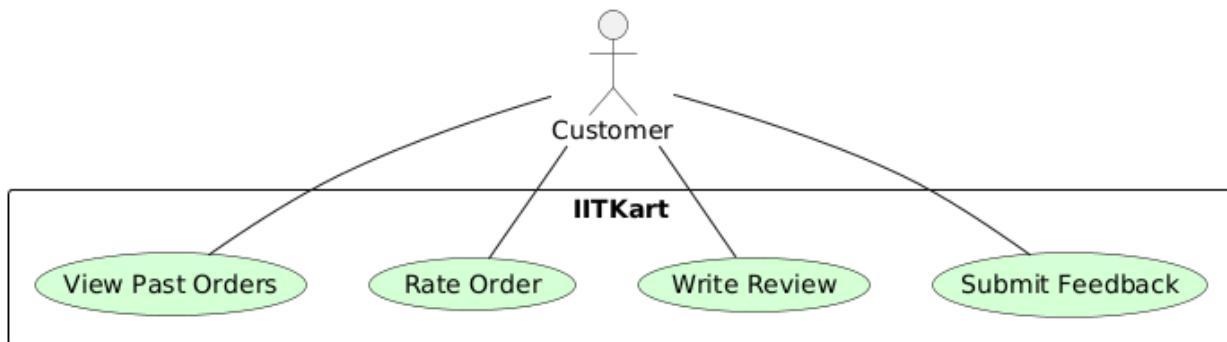
**Post conditions** - Admin receives complaints and grievances. The average ratings will be visible on profiles of vendors and riders.

**Actors – Customer**

**Exceptions** - None

**Includes** (other use case IDs)-None

**Notes/Issues** - Feedback moderation mechanisms may be required to prevent abusive or inappropriate content. Ratings should be averaged over time and will lead to rankings of vendors and/or products.



### 3.3.5. U5 - Viewing Account/Opening Settings

**Author** – Mayukh

**Purpose** - This allows a customer/vendor/rider (any user of the system) to open their account and view/make changes wherever applicable. Customers will also be able to access wallet/Kart coins henceforth.

**Requirements Traceability** – Users shall be able to view and manage their account details.

**Priority** - Medium. Important for account maintenance and data storage but not critical to core functioning of the system.

**Preconditions** - Users must be logged in to the system.

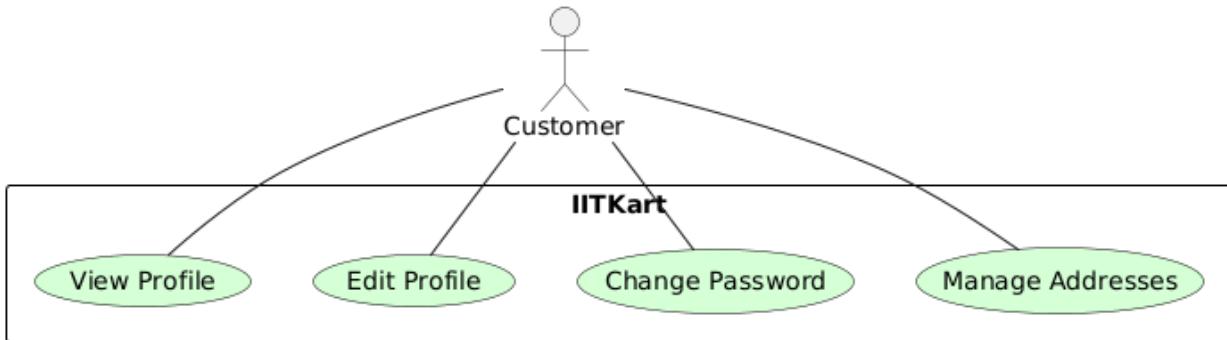
**Post conditions** - None.

**Actors** – Customer ,Vendor ,Rider

**Exceptions** - Users may enter incorrect /invalid information. They need to ensure that details submitted are correct to the best of their knowledge.

**Includes** (other use case IDs)-None

**Notes/Issues** - Sensitive operations such as password changes may require additional authentication .Profile picture uploads may be integrated too .



### 3.3.6. U6 - Admin Monitors Platform Activity

**Author**-Hardik

**Purpose** - This use case allows the admin to monitor overall platform metrics and live order status.

**Requirements Traceability** - Admin shall be able to view platform metrics and live order status.

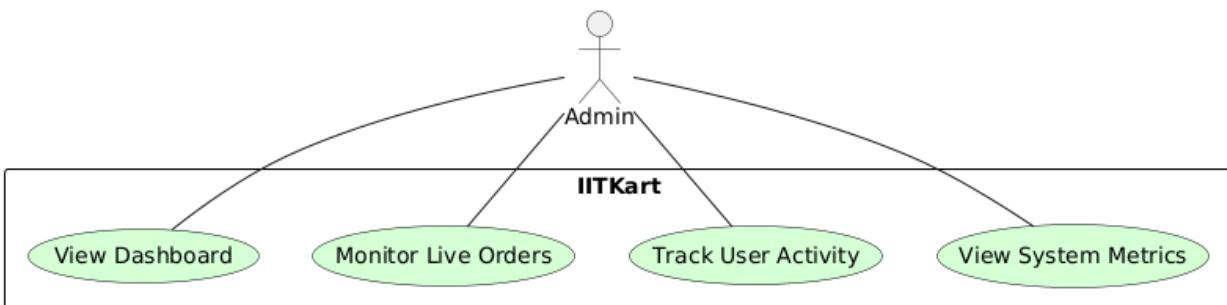
**Priority** - Medium.

**Preconditions** - Admin must be logged in to the system.

**Post conditions** - Admin gains real-time visibility into platform activity.

**Actors** - Admin

**Exceptions** - Temporary server unavailability.



### 3.3.7. U7 - Admin Manages Users, Vendors, and Riders

**Author** - Hardik

**Purpose** - Allows the admin to approve, suspend, or manage users, vendors, and riders.

**Requirements Traceability** - Admin shall be able to manage users, vendors, and riders.

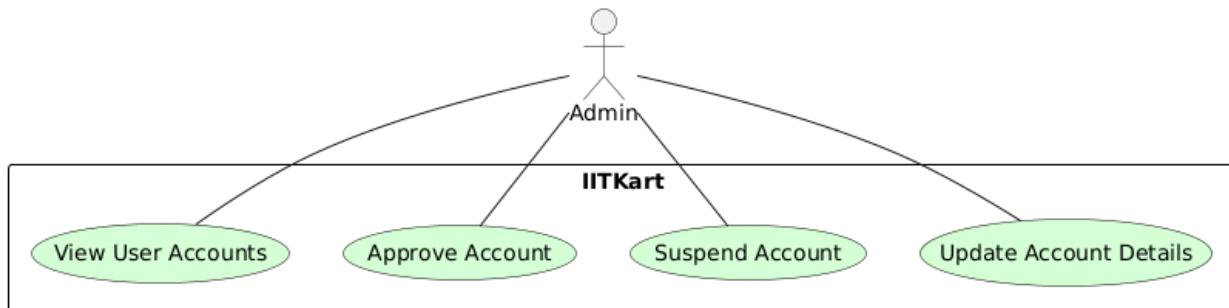
**Priority** - High.

**Preconditions** - Admin must be logged in to the system.

**Post conditions** - User, vendor, or rider status is updated.

**Actors** - Admin

**Exceptions** - Invalid user selection or insufficient privileges.



### 3.3.8. U8 - Admin Views Financial Records

**Author**-Hardik

**Purpose** - Allows the admin to view invoices, receipts, and settlement records.

**Requirements Traceability** - Admin shall be able to view invoices, receipts, and settlement records.

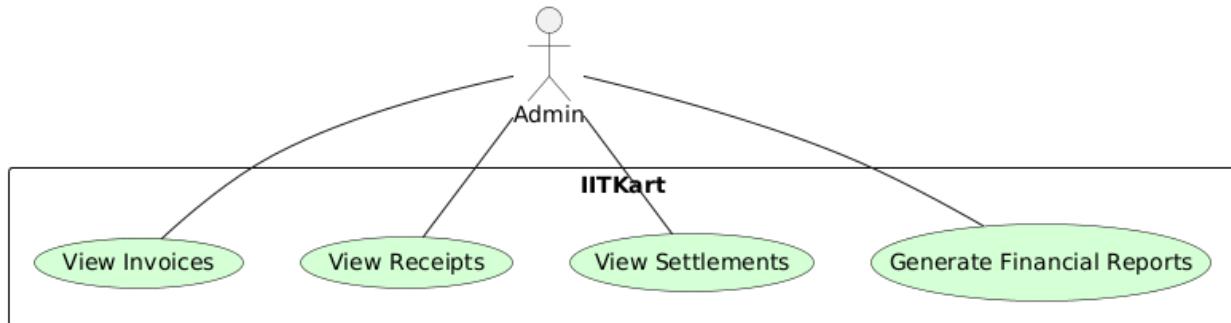
**Priority** - Medium.

**Preconditions** - Admin must be logged in to the system.

**Post conditions** - Financial records are displayed to the admin.

**Actors** - Admin

**Exceptions** - Incomplete or delayed transaction records.



### 3.3.9. U9 -Vendor can Execute Product Inventory Controls

**Author –** Palak Bandhu

**Purpose –** To allow vendors (e.g., Amul Parlour) to maintain an accurate digital menu by adding new items or updating existing ones.

**Requirements Traceability –**

- The system shall display a product list with images and prices (e.g., Veg Sandwich at ₹50).
- The system shall allow adding, editing, or deleting inventory items.

**Priority –** High. Crucial for ensuring customers only order available items.

**Preconditions –**

- Vendors are logged into their specific account (e.g., [amul@iitk.ac.in](mailto:amul@iitk.ac.in)).
- The system has successfully filtered items by the correct vendorID.

**Post conditions –**

- The database is updated with new product details.
- Changes are immediately visible to customers on the "IITKart" home page.

**Actors –**

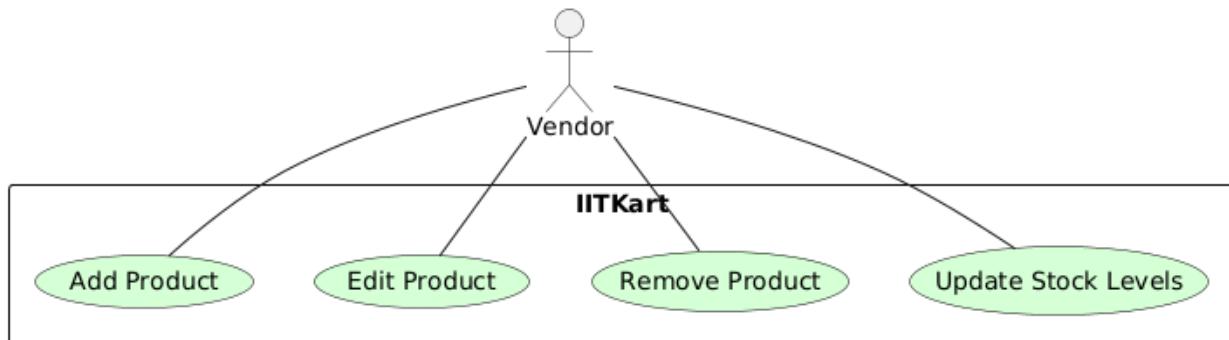
- Vendor: Initiates updates and manages stock.
- System: Validates data and updates the live inventory.

**Exceptions –**

- Image Upload Error: Product cannot be saved without a valid thumbnail.
- Invalid Input: Non-numeric values entered in the "Price" field.

**Includes – User Authentication.**

**Notes/Issues** – Currently, the fix for vendor user IDs ensures that Amul only sees Amul products and not those of other campus vendors.



### 3.3.10. U10 - Track Active and Past Orders

**Author** – Palak Bandhu

**Purpose** – To provide a real-time view of incoming orders and a log of completed transactions.

**Requirements Traceability** –

- Display "Active Orders" with specific locations (e.g., Hall 7, Room 410).
- Show "Order History" for record-keeping.

**Priority** – High.

**Pre Conditions** – A customer has placed an order; vendor is in "Online" mode.

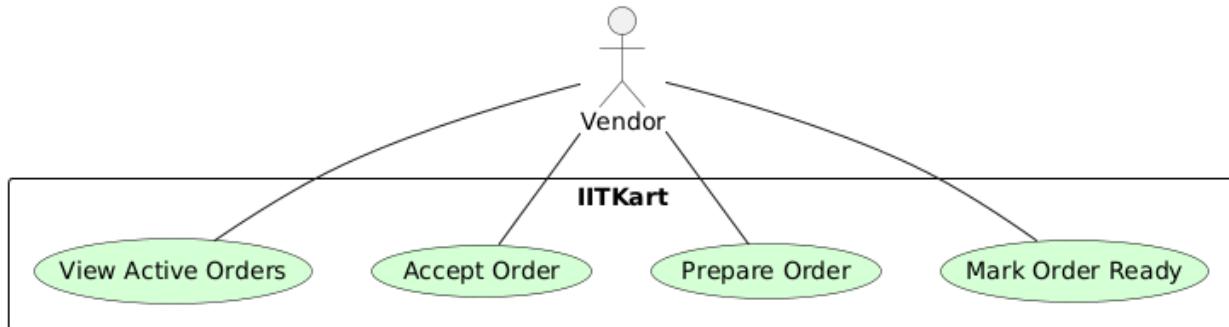
**Post conditions** – Order status changes to "Accepted"; system triggers rider search.

**Actors** – Vendor, System.

**Exceptions** – Network latency; missing delivery room details.

**Includes** – User Authentication.

**Notes/Issues** – UI must clearly show "Order #" (e.g., #ORD004) for hand-off.



### 3.3.11. U11 - Vendor Reviews Business Performance & Feedback

**Author** – Palak Bandhu

**Purpose** – To monitor customer satisfaction and read specific feedback regarding service quality.

**Requirements Traceability** –

- The system shall display the "Average Rating" (e.g., 4.0 stars).
- System shall list individual reviews with associated "Order ID" (e.g., #ORD001).

**Priority** – Medium.

**Pre Conditions** – Vendor is logged in; at least one review has been submitted.

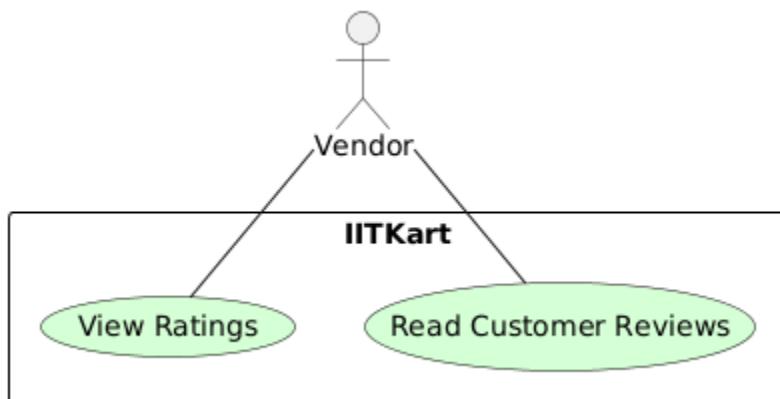
**Post conditions** – Vendor views aggregated performance data.

**Actors** – Vendor, System.

**Exceptions** – No reviews available (display 0.0); data sync delay.

**Includes** – User Authentication.

**Notes/Issues** – The "Average Rating" card on the dashboard must sync with the data in the Reviews tab.



### 3.3.12. U12 - Vendor Customizes Shop Settings

**Author** – Palak Bandhu

**Purpose** – To update the shop's identity and operational hours.

**Requirements Traceability** –

- Allow editing of Shop Name, Email, and Phone.
- Set "Availability Hours" for the campus service.

**Priority** – Medium.

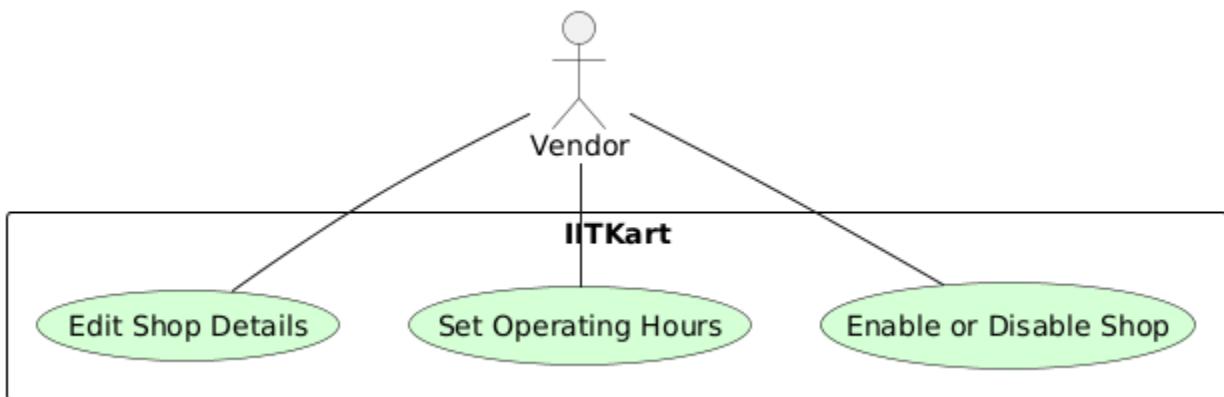
**Pre Conditions** – Vendor is on the "Settings" tab.

**Post conditions** – Profile updated in database and customer app.

**Actors** – Vendor, System.

**Exceptions** – Invalid email format.

**Includes** – User Authentication.



### 3.3.13. U13 - Analyze Operational Performance Metrics

**Author** – Palak Bandhu

**Purpose** – To provide vendors with a real-time, data-driven overview of financial earnings, order volume, and customer satisfaction metrics to facilitate informed operational decision-making.

**Requirements Traceability** –

- Vendor accesses the **Vendor Dashboard** upon successful login.
- The system invokes the analytics engine to pull data from the `MOCK_VENDORS` array in the `AppContext`.
- System renders the following specific KPIs prominently at the top of the dashboard:
 

<b>Total Earnings:</b> (e.g., ₹45,600)	<b>Total</b>
<b>Total Orders:</b> (e.g., 156)	<b>Active</b>
<b>Average Rating:</b> (e.g., 4.5 stars)	
<b>Products:</b> (e.g., 5 products)	
- Vendor evaluates performance metrics to adjust store availability or inventory.

**Priority** – High – Essential for strategic oversight and quality assurance within the campus delivery ecosystem.

**Pre Conditions** –

- Vendor is successfully authenticated into the administrative portal.

- The system has processed at least one transaction and associated customer feedback

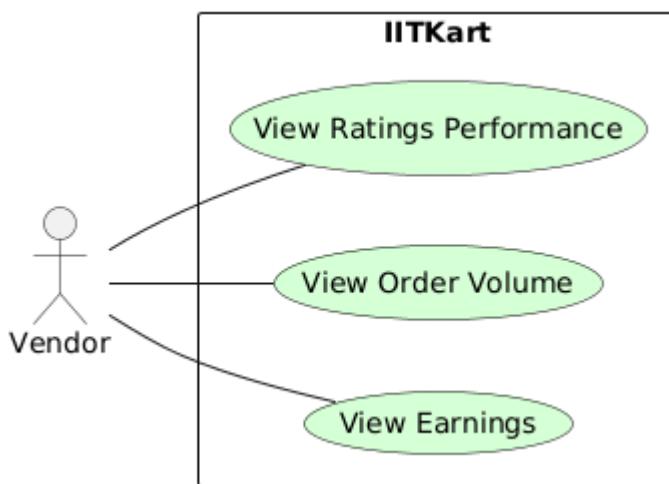
**Post Conditions –**

- Vendor identifies net revenue, total order volume, and service quality trends.
- Analytical data is presented clearly via high-level KPI cards.

**Actors – Vendor (Primary), System (Secondary).**

**Exceptions –**

- E1: Null Data State:** If no orders exist, the system displays default zero values or "No Data Available" markers.
- E2: Data Latency:** Real-time earnings may show a slight delay between order completion and dashboard reflection.
- Includes – Data Isolation:** The system must strictly ensure each vendor sees only their specific metrics upon login



### 3.3.14. U14 - Rider Registration

**Author** - Mahi Mittal

**Purpose** - Allows a new delivery partner (rider) to register on the IITKart platform by providing mandatory personal, contact, and verification details. Successful registration enables the rider to access the delivery partner interface and perform delivery-related actions.

**Requirements Traceability -**

- System shall allow delivery partners to register on the platform
- System shall collect and validate rider details during registration

**Priority** - High (Essential for onboarding delivery partners)

**Preconditions -**

- Rider is not already registered on the system
- Rider has valid identification and contact information

**Postconditions -**

- Rider account is created in the system
- Rider is redirected to the delivery partner interface
- Account status is set to *Active* or *Pending Approval*

**Actors** - Riders, System, Admin (for approval, if required)

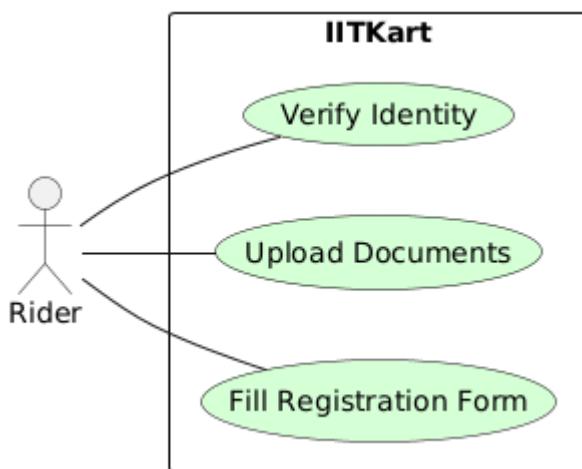
**Exceptions -**

- Rider is below the minimum age requirement (18 years)
- Mandatory fields are missing or invalid
- Duplicate registration attempt

**Includes (other use case IDs) -**

- U7 – Admin Manages Users, Vendors, and Riders

**Notes / Issues** - Registration details include name, gender, age, contact number, residence, vehicle type, and payment information. Additional verification may be required by the admin.



### 3.3.15. U15 - Rider views active orders

**Author** - Mahi Mittal

**Purpose** - This use case allows a rider to view a list of all currently active and ongoing delivery orders that have been accepted by the rider.

### Requirements Traceability -

- Rider shall be able to view active delivery orders

**Priority** - High (Necessary for managing ongoing deliveries)

### Preconditions -

- Rider is logged into the system
- Rider has accepted at least one delivery order

### Postconditions -

- Active orders are displayed with current status

**Actors** - Riders, System

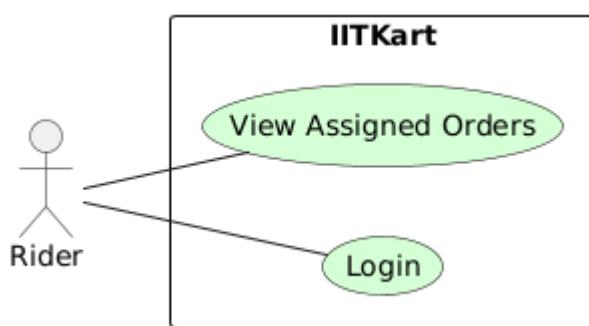
### Exceptions -

- No active orders available

### Includes (other use case IDs) -

- U11 – Rider Views Order Details and Accepts/Rejects Delivery

**Notes / Issues** - Each active order should be displayed with a unique order ID and status to avoid confusion during delivery.



### 3.3.16. U16 - Rider views order details and accepts/rejects delivery

**Author** - Mahi Mittal

**Purpose** - This use case enables a rider to view the complete details of a delivery order and decide whether to accept or reject the delivery assignment.

### Requirements Traceability -

- Rider shall be able to view order details
- Rider shall be able to accept or reject delivery requests

**Priority** - High (Core functionality for delivery execution)

**Preconditions** -

- Rider is logged into the system
- A customer has placed an order and completed payment

**Postconditions** -

- *If accepted: Order status is updated to Accepted by Rider*
- *If rejected: Order is reassigned or marked as No Rider Available*

**Actors** - Riders, System

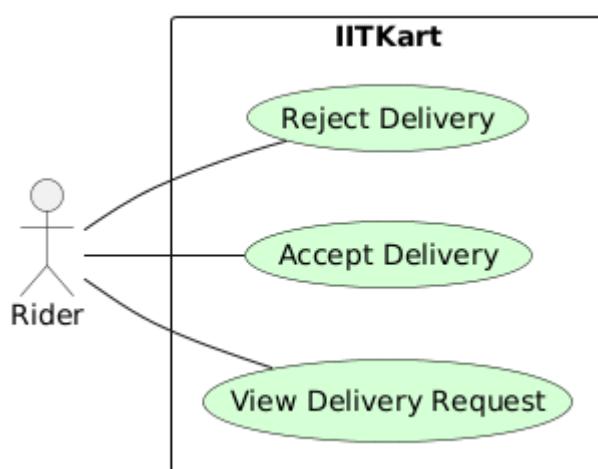
**Exceptions** -

- Order is already accepted by another rider
- Rider does not respond within the allowed time window

**Includes (other use case IDs)** -

- U2 – Customer Places an Order
- U3 – Customer Completes Payment

**Notes / Issues** - Order details may include pickup location, delivery location, expected delivery time, and rider earnings for the order.



### 3.3.17. U17 - Rider performance metrics and personal details

**Author** - Mahi Mittal

**Purpose** - This use case allows the rider to manage personal account details and view performance metrics such as completed deliveries, ratings, reviews, and earnings.

**Requirements Traceability** -

- Rider shall be able to manage account details
- Rider shall be able to view performance metrics

**Priority** - Medium (Important for transparency and long-term engagement)

**Preconditions** -

- Rider is logged into the system

**Postconditions** -

- Updated account details are saved in the system
- Performance metrics are displayed to the rider

**Actors** - Riders, System

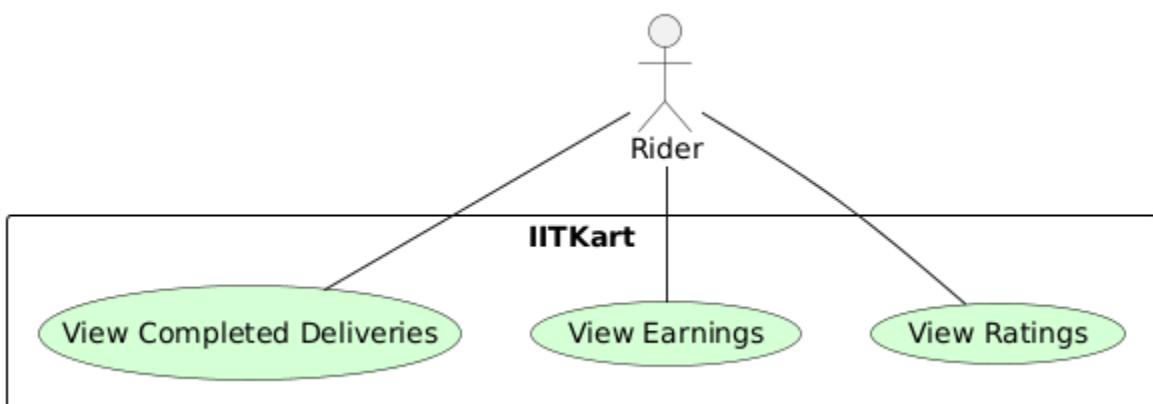
**Exceptions** -

- Invalid information entered during profile update
- No completed deliveries available for metric computation

**Includes (other use case IDs)** -

- U5 – Viewing Account / Opening Settings
- U3 – Customer Completes Payment

**Notes / Issues** - Performance metrics may influence rider rankings or priority in delivery assignment in future system versions.



### 3.3.18. U18 - Rider Reports Delivery Issue / Requests Support

**Author** - Mahi Mittal

**Purpose** - Allows a rider to report issues encountered during the delivery process and request assistance from the system or administrator. The objective is to handle exceptional situations that prevent smooth or successful order completion and ensure proper resolution.

**Requirements Traceability** -

- System shall allow riders to report delivery-related issues
- System shall record and track reported issues for administrative review
- System shall notify the admin when rider assistance is required

**Priority** - Medium (Not part of the normal delivery flow, but essential for robustness and real-world reliability of the system.)

**Preconditions** -

- Rider must be logged into the system
- Rider must have an active delivery order assigned

**Postconditions** -

- The reported issue is logged in the system with associated order ID
- Admin is notified of the issue
- Order status may be temporarily updated to *On Hold* or *Issue Reported*

**Actors** - Riders, System, Admin (Secondary – for resolution and intervention)

**Exceptions** -

- Rider submits incomplete or unclear issue details
- Network failure while submitting the issue report
- Order status changes before issue submission (e.g., order cancelled)

**Includes (other use case IDs)** -

- U12 – Rider Views Active Orders
- U7 – Admin Manages Users, Vendors, and Riders

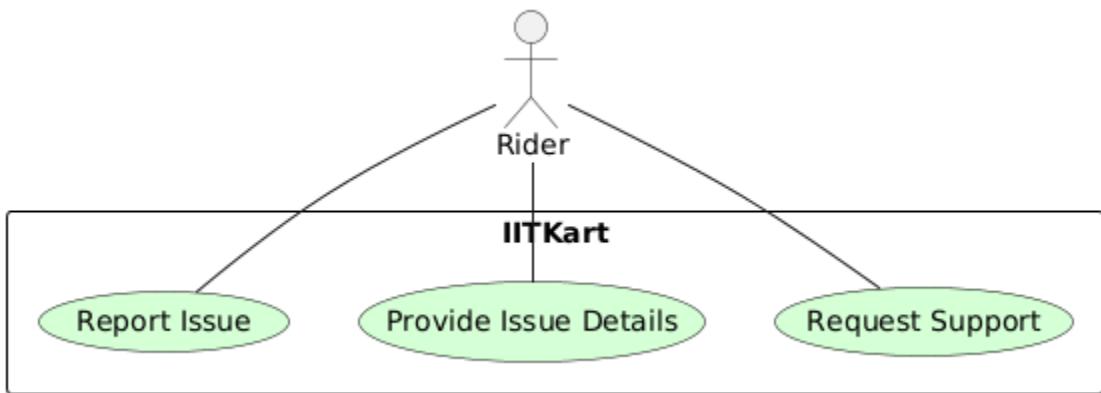
**Notes / Issues** -

Examples of delivery issues include:

- Customer is unreachable or not present at delivery location
- Incorrect or incomplete delivery address
- Vendor delay in handing over the order

- Item missing or damaged
- Emergency situations preventing delivery completion

This use case ensures that riders are not forced to mark orders as delivered or rejected inappropriately when genuine issues arise. Future versions may include predefined issue categories or in-app chat support.



## 4. Other Non-functional Requirements

### 4.1. Performance Requirements

- The system shall support simultaneous access by multiple users (customers, vendors, riders, and admin) within the campus network without system crashes or data inconsistency.
- For normal campus usage conditions, page load time should not exceed 3 seconds for common operations such as login, browsing items, viewing cart, and placing orders.
- The system shall process order placement requests within a reasonable time such that the user receives confirmation within 2–3 seconds after submitting the order.
- The system shall be capable of handling peak usage hours, such as meal times or evenings, when a large number of users may place orders concurrently.
- In case of unavailability of riders, the system shall notify the user promptly without causing the application to hang or fail.

### 4.2. Safety and Security Requirements

- The system shall require user authentication (login) for all registered users including customers, vendors, riders, and administrators.
- User passwords shall be securely stored using encryption or hashing techniques, and plaintext passwords shall not be stored.
- Payment-related information (if online payment is enabled) shall be handled securely and no sensitive payment data shall be stored on the system servers.
- The system shall prevent unauthorized access to user data and ensure privacy of personal information such as phone numbers and order history.
- The system shall handle unexpected failures gracefully without loss or corruption of order data.
- The system shall ensure role-based access control, where:
  - ❖ Customers can only access customer functionalities.
  - ❖ Vendors can manage only their own listed items.
  - ❖ Riders can view and accept assigned orders.
  - ❖ Admin has elevated privileges to manage users and complaints.

## 4.3. Software Quality Attributes

### 4.3.1. Usability

- The system shall provide a simple and intuitive user interface suitable for campus users with minimal learning effort.
- Users shall be able to perform key actions such as searching items, placing orders, and tracking deliveries with minimal number of steps.
- Error messages and system notifications shall be clear and user-friendly.

### 4.3.2. Reliability

- The system shall function correctly for extended periods during daily campus operations without frequent crashes.
- Orders once placed shall not be lost or duplicated, even in the case of temporary system failures.

### 4.3.3. Maintainability

- The system shall be designed in a modular manner, allowing individual components (user management, order handling, vendor management) to be modified independently.
- Future updates such as adding new features or modifying workflows shall be possible without major changes to the entire system.

### 4.3.4. Scalability

- Although the current scope is limited to a single campus, the system shall be designed such that it can be extended to multiple campuses in the future with minimal architectural changes.

### 4.3.5. Availability

- The system shall be available for use during most hours of the day, especially during peak ordering times.
- Planned maintenance or downtime shall be minimized and preferably scheduled during low-usage hours.

## Appendix A – Data Dictionary

### Data Dictionary

- User and Authentication Entities

Variable Name	Data Type	Description	Constraints / States	Operations
user_id	String (UUID)	Unique identifier for every user.	Primary Key, Not Null	Read
email	String	Any valid email.	Unique, valid email format (RFC 5322), must be verified	Read, Write
password_hash	String	Hashed password string.	Encrypted (SHA-256 / Argon2)	Write, Update
role	Enum	User permission level.	CONSUMER, VENDOR, RIDER, ADMIN	Read
phone_number	String	Contact number for delivery coordination.	10 digits, Unique	Read, Update
is_verified	Boolean	Status of email/phone verification.	True (Verified), False (Pending)	Read, Update
created_at	DateTime	Timestamp of account creation.	Auto-generated	Read

- Order and Transaction Entities

Variable Name	Data Type	Description	Constraints / States	Operations
order_id	String (UUID)	Unique identifier for a transaction.	Primary Key	Read

consumer_id	String (UUID)	ID of the user placing the order.	Foreign Key (User)	Read
vendor_id	String (UUID)	ID of the shop fulfilling the order.	Foreign Key (User/Vendor)	Read
rider_id	String (UUID)	ID of the delivery partner.	Foreign Key (User), Nullable	Read, Update
order_status	Enum	Current lifecycle state of the order.	PENDING, ACCEPTED, PREPARING, READY, OUT_FOR_DELIVERY, DELIVERED, CANCELLED	Read, Update
total_amount	Decimal	Final billing amount.	Must be > 0.00	Read
delivery_fee	Decimal	Charge for delivery service.	Default ₹0 for Pickup	Read
payment_status	Enum	Status of the financial transaction.	PENDING, SUCCESS, FAILED, REFUNDED	Read, Update
otp_delivery	String (4 chars)	Code to verify handover to Consumer.	Generated at OUT_FOR_DELIVERY	Read, Verify

- Product and Inventory Entities

Variable Name	Data Type	Description	Constraints / States	Operations
product_id	String (UUID)	Unique ID for an item.	Primary Key	Read
name	String	Name of the dish or product.	Max 100 characters	Read, Write
category	Enum	Classification of the item.	FOOD, STATIONERY, GROCERY, DRINKS	Read, Update
price	Decimal	Unit price of the item.	Must be > 0	Read, Update
is_veg	Boolean	Dietary classification.	True (Veg), False (Non-Veg)	Read, Update
stock_status	Enum	Availability status.	IN_STOCK, OUT_OF_STOCK	Read, Update
image_url	String	Link to product image.	Valid URL format	Read, Update

- **Cart and Reviews (Auxiliary)**

Variable Name	Data Type	Description	Constraints / States	Operations
cart_id	String (UUID)	Temporary storage for unplaced orders.	One per Consumer	Read, Write
quantity	Integer	Number of units of a product.	Min 1, Max 20	Update
review_id	String (UUID)	Unique ID for feedback entry.	Primary Key	Read
rating	Integer	Score given by Consumer.	Range 1–5	Read, Write
comment	String	Text feedback (optional).	Max 500 chars, profanity filtered	Read, Write
review_target	Enum	Entity being reviewed.	VENDOR, RIDER	Read

- Rider and Logistics

Variable Name	Data Type	Description	Constraints / States	Operations
vehicle_number	String	License plate of rider vehicle.	Required for gate entry	Read, Update
current_lat	Float	Current GPS latitude.	-90 to 90	Update
current_long	Float	Current GPS longitude.	-180 to 180	Update
is_active	Boolean	Rider duty status.	True (Online), False (Offline)	Read, Update
earnings_today	Decimal	Total payout for the day.	Resets at 00:00	Read, Update

### Data Relationships and Constraints

- Relationships

- Customers place orders that are associated with specific vendors offering the requested products or services.
- Each order may be assigned to a delivery partner (rider) responsible for picking up the order from the vendor and delivering it to the customer.
- Vendors manage product listings and receive multiple orders from different customers.
- Delivery partners may handle multiple delivery assignments, subject to system-defined availability and workload constraints.
- Administrators oversee and manage all users, orders, and platform activities.

- Constraints

- Email addresses must be unique across all user roles (customers, vendors, delivery partners, and administrators).

- Passwords must comply with defined security policies (minimum length and required character types).
- An order cannot be marked as *Delivered* unless it has been accepted by a delivery partner.
- Customers are allowed to submit ratings and feedback only after successful order completion.
- A delivery partner cannot be assigned more concurrent deliveries than the system-defined limit.
- Vendors cannot receive new orders when marked as unavailable or closed.

## Appendix B - Group Log

- A WhatsApp group was created for communication between members.
- Offline meetings were hosted whenever necessary to take inputs from members on the project progress and discuss further course of actions.

Meeting Date	Meeting Type	Ongoing Discussions in the meet
07.01.2026	Offline Meet	Initial brainstorming session focused on identifying and evaluating multiple potential project ideas. The team discussed feasibility, scope, and relevance before shortlisting a set of preferred project options.
10.01.2026	Offline Meet	Detailed discussion with the instructor regarding the finalized project concept, objectives, and scope. The project overview and related details were finalized, documented, and submitted on the same day.
15.01.2026	Offline Meet	Identified functional and non-functional requirements for all user roles. Requirements were categorized and refined for clarity.
18.01.2026	Offline Meet	Completion of the requirement identification phase followed by the initiation of the Software Requirements Specification (SRS) document. Preliminary designs for user interfaces and system workflows were also discussed.
20.01.2026	Offline Meet	Finalized task allocation and UI theme selection.
21–23.01.2026	Offline Meet	Multiple working sessions dedicated to drafting, structuring, and refining the Software Requirements Specification document
24.01.2026	Offline Meet	Final meeting conducted prior to submission to review the complete SRS document. The team performed detailed proofreading, resolved formatting and consistency issues, and validated the document against project requirements.