A PROJECT ON

**COURIER SERVICE MANAGEMENT SYSTEM**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

Hinjawadi

**SUBMITTED BY:**

Jujjavarapu Naveen,

Vyas Juili Pradeep,  
 Shruti Awate,

Khushi chaudhari,

Ravi Prasad

**UNDER THE GUIDENCE OF:**

Mrs. Pooja Jaiswal Faculty Member

Sunbeam Institute of Information Technology, Pune

**ACKNOWLEDGEMENT**

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Naveen Jujjavarpu,

Juili Vyas,

Shruti Awate,

Khushi Chaudhari,

Ravi Prasad

PG-DAC

SIIT Pune

**A PROJECT ON**

**“COURIER SERVICE MANAGEMENT SYSTEM”**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE COURSE OF

DIPLOMA IN ADVANCED COMPUTING FROM CDAC



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

Hinjawadi

**SUBMITTED BY:**

Naveen Jujjavarapu,

Juili Vyas,  
 Shruti Awate,

Khushi chaudhari,

Ravi Prasad

**UNDER THE GUIDENCE OF:**

Mrs. Pooja Jaiswal Faculty Member

Sunbeam Institute of Information Technology, PUNE.



**CERTIFICATE**

This is to certify that the project work under the title ‘COURIER SERVICE MANAGEMENT SYSTEM’ is done by Naveen Jujjavarapu, Juili Vyas, Shruti Awate, Khushi Chaudhari, Ravi Prasad in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Project Guide**

Date: 11-02-2025

**Mr. Yogesh Kolhe Course Co-Coordinator**

Table of Contents:

1. INTRODUCTION 2
2. [REQUIREMENT 3](#_TOC_250005)
   1. [Functional Requirements 3](#_TOC_250004)
   2. Customer Module 4
   3. Vendor Module 6
   4. Delivery Boy Module 9
   5. Admin Module 11
3. [Non-Functional Requirements 12](#_TOC_250003)
   * 1. Hardware and Software Interfaces 12
4. [DESIGN 13](#_TOC_250002)
   1. [Database design 13](#_TOC_250001)
5. CODING STANDARD IMPLEMENTED 19
6. [TEST REPORT 22](#_TOC_250000)
7. APENDIX A
8. Entity Relationship Diagram 27
9. Data Flow Diagram 28
10. Class Diagram 29
11. APENDIX B

UI Screenshots 30

1. REFERENCE

# INTRODUCTION TO PROJECT

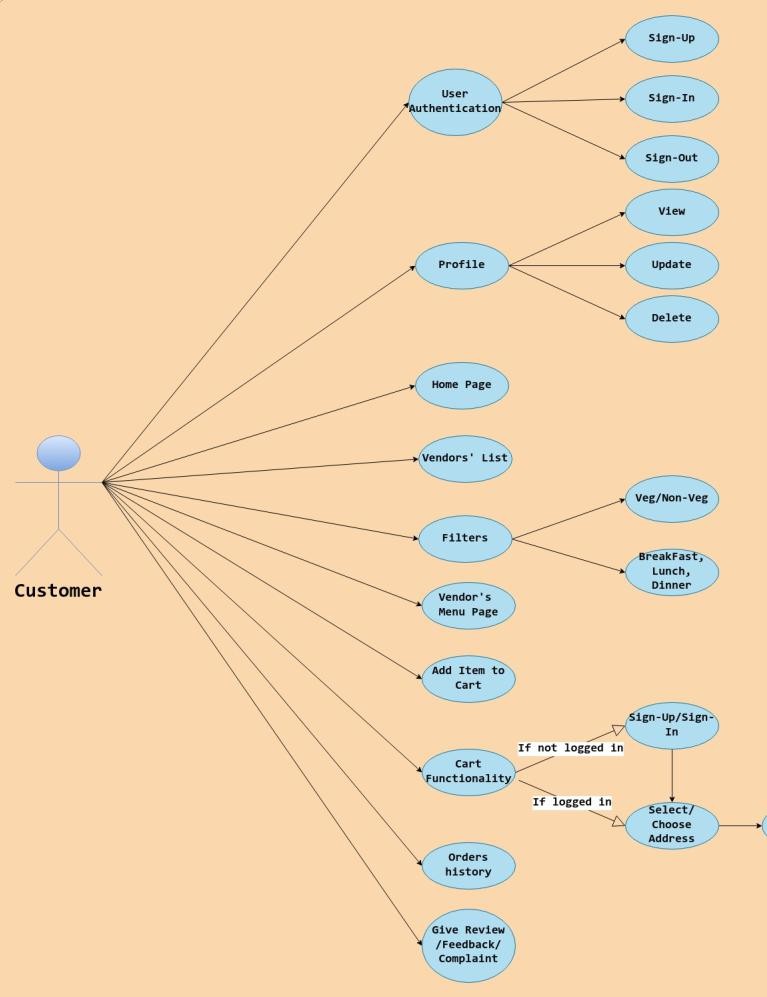
The Courier Management System is a comprehensive web application designed to streamline the process of managing courier services. The system provides a user-friendly interface for Users, Delivery Agents, Warehouse Managers, and Admins to interact with the platform, manage shipments, track deliveries, and oversee warehouse operations.

With the increasing demand for fast and efficient logistics services, businesses and individuals require a robust system to manage their shipments seamlessly. This system facilitates automation in courier operations, reducing manual work and improving overall efficiency. Users can easily schedule deliveries, track their packages in real-time.

Delivery agents benefit from real-time assignment of delivery tasks, allowing them to optimize their routes and update shipment statuses dynamically. Warehouse managers can efficiently handle inventory and shipments while ensuring that deliveries are timely and accurate. The system is equipped with comprehensive reporting tools to assist administrators in monitoring overall performance and making data-driven decisions.

# REQUIREMENTS

# FUNCTIONAL REQUIREMENTS



## Customer Flow

* + 1. **Home Page**
* **Objective**: Display a list of vendors.

## Features:

* + View a list of available vendors with search and filter options.

## Vendor Selection

* **Objective**: Select a vendor to view their menu.

## Features:

* + Click on a vendor to see their menu items.

## Menu Items

* **Objective**: Allow customers to view and manage menu items.

## Features:

* + View a list of menu items for the selected vendor.
  + Add items to the cart.
  + Update quantities and remove items from the cart.

## Cart Interaction

* **Objective**: Manage cart and checkout process.

## Features:

* + **If Logged In**:
    - Navigate to the cart page.

## If Not Logged In:

* + - Display a login prompt.
    - If not registered, prompt registration before login.

## Cart Page

* **Objective**: Finalize the purchase process.

## Features:

* + View cart items.
  + Choose a delivery address.
  + Place the order.
  + If no delivery address is selected, prompt the user to choose one.
  + Redirect to the home page after placing the order.

## Profile Page

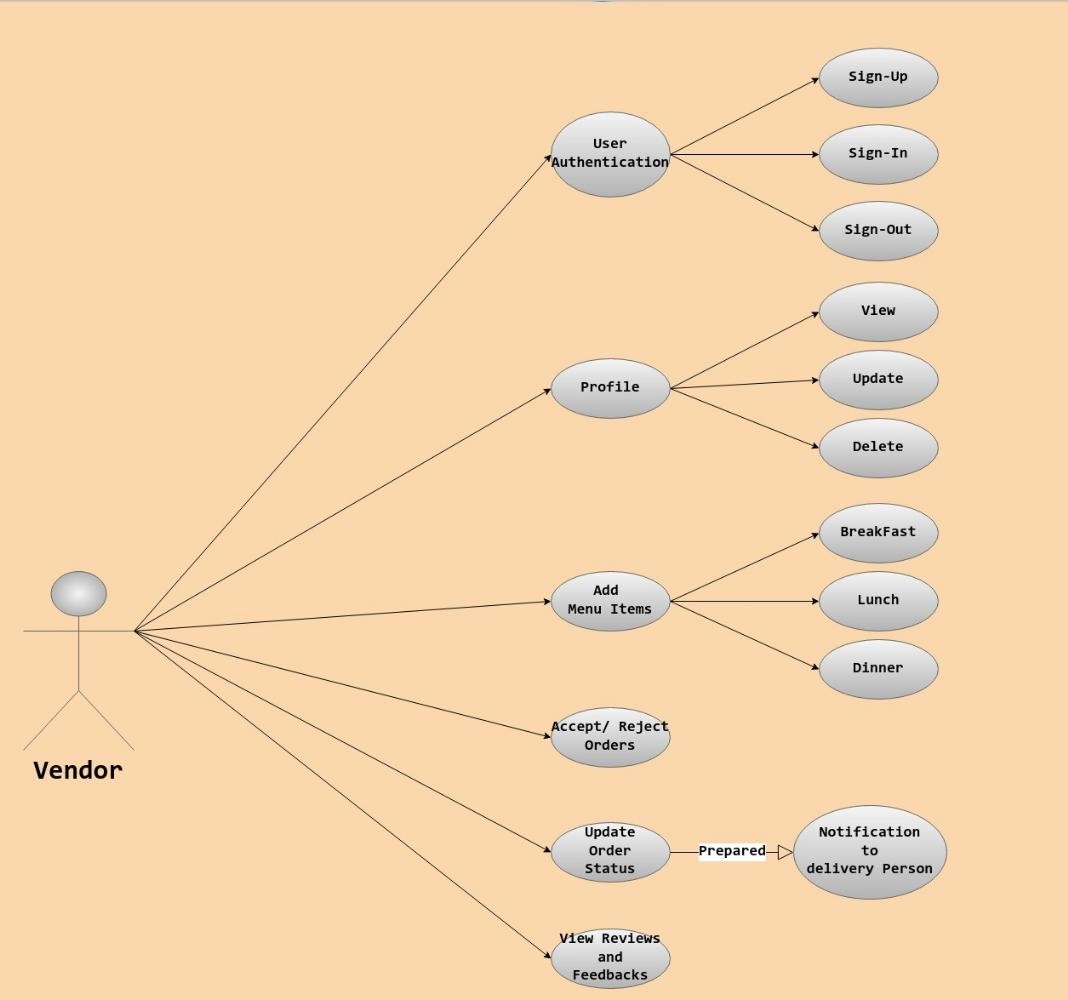
* **Objective**: Manage user profile and view order history.

## Features:

* + **If Not Logged In**:
    - Display a sign-in prompt.

## If Logged In:

* + - View and update profile details.
    - View order history and addresses.
    - Submit reviews for orders.
    - Logout.



## Vendor Flow

* + 1. **Home Page**
* **Objective**: Allow vendors to manage their accounts.

## Features:

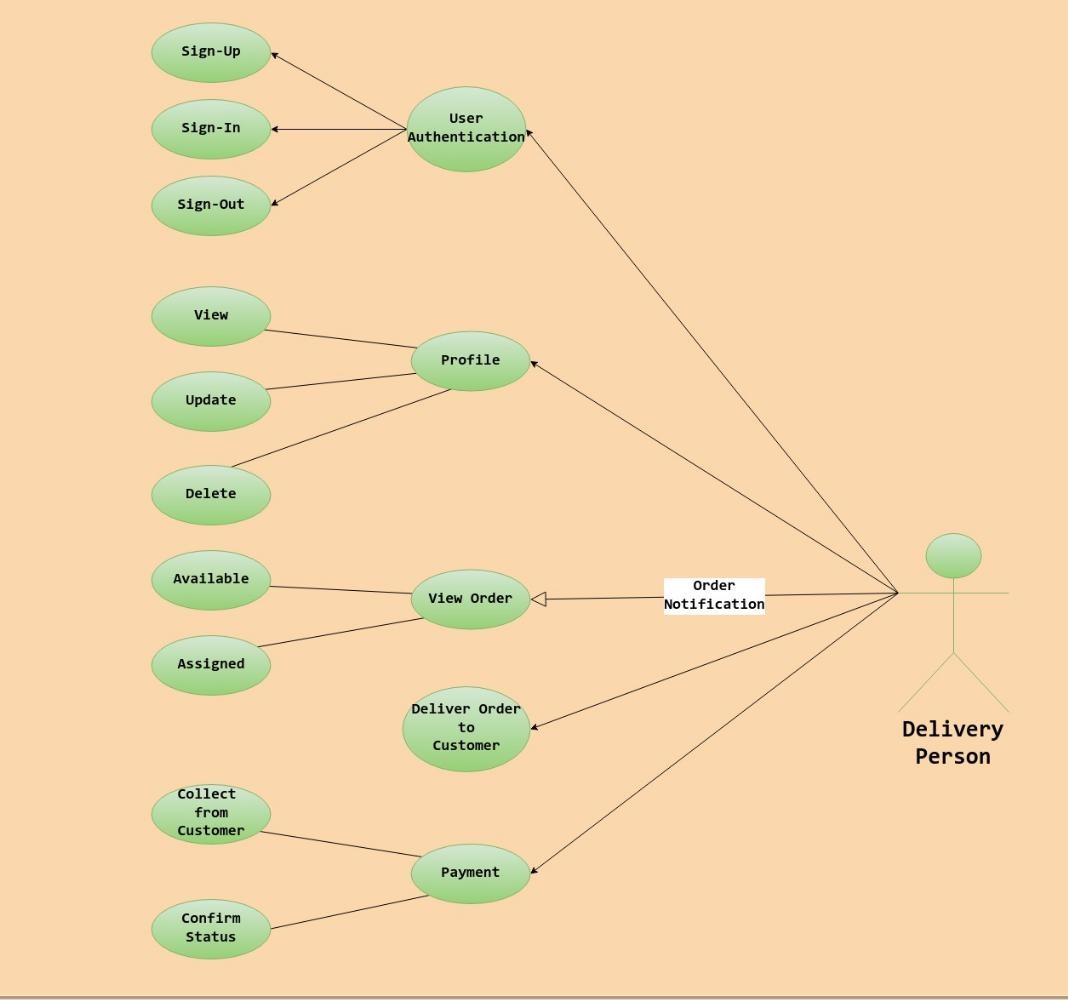
* + Sign up and sign in to access vendor functionalities.

## Vendor Home Page

* **Objective**: Manage menu items and view orders.

## Features:

* + Add menu items in breakfast, lunch, and dinner categories.
  + View placed and delivered orders.
  + Update or delete menu items.
  + Logout.

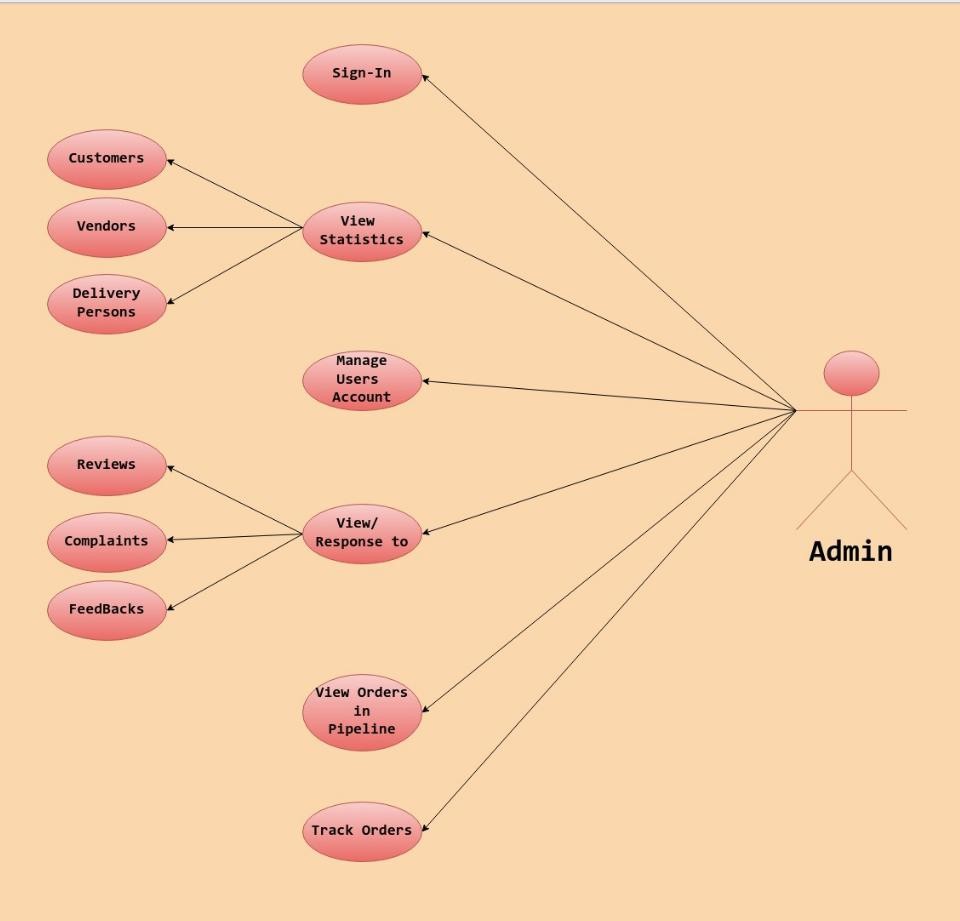


## Delivery Boy Flow

* + 1. **Home Page**
* **Objective**: Manage delivery boy account.
* **Features**:
* **Sign up and sign in functionalities.**
  + 1. **Delivery Boy Home Page**
* **Objective**: Manage assigned and delivered orders.

## Features:

* + View and assign placed orders.
  + Mark Placed Order as Delivered after delivering the order.
  + View delivered orders.
  + Logout.



## Admin Flow

* + 1. **Home Page**
* **Objective**: Admin access to system functionalities.

## Features:

* + Sign in to access admin functionalities.

## Admin Home Page

* **Objective**: Oversee and manage system operations.

## Features:

* + View counts of customers, vendors, and delivery boys.
  + View lists of customers, vendors, and delivery boys.
  + View orders and customer reviews/feedback.

## Non-Functional Requirements

* 1. **Performance: The system should handle at least 1000 concurrent users.**
  2. **Scalability: The system should be scalable to accommodate future growth.**
  3. **Security: Ensure secure user authentication and data encryption.**
  4. **Availability: The system should have 99.9% uptime.**
  5. **User Interface: The interface should be intuitive and user-friendly.**

* 1. **Other Requirements**

**3.6.1 Hardware Interfaces**

**Requirements: Intel Core i5 or higher (or AMD equivalent),**

1. **GB RAM, 512 GB SSD or larger.**



## 3.6.2 Software Interfaces

* **Operating Systems**: MS Windows 13, Ubuntu 22.04.
* **Database**: MySQL.
* **Server**: Embedded Tomcat.
* **Browsers**: Compatible with modern web browsers.

## System Design

**3.6.4 Architecture**

* **Front-End**: Developed using React.js, providing an

intuitive UI for users.

* **Back-End**: Built using Spring Boot and Spring Security, handling all

processing and logic.

* **Database**: Uses MySQL for structured data storage and

retrieval.

* **Server**: Embedded Tomcat for hosting the application.

# DESIGN

## Database Design

The following table structures depict the database design.

**Table 1: Users**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes; 0=No)** |
| **3** | **id** | **BIGINT** | **-** | **0** |
| **0** | **first\_name** | **VARCHAR** | **50** | **0** |
| **0** | **last\_name** | **BIT(1)** | **50** | **0** |
| **0** | **password** | **VARCHAR** | **-** | **0** |
| **0** | **email** | **VARCHAR** | **40** | **0** |
| **0** | **role** | **ENUM** | **-** | **0** |
| **1** | **address\_id** | **BIGINT** |  | **1** |
| **0** | **contact\_number** | **VARCHAR** | **10** | **0** |

**Table 2: Addresses**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| 0 | city | VARCHAR | 255 | 1 |
| 0 | flat\_no | VARCHAR | 255 | 1 |
| 0 | landmark | VARCHAR | 255 | 1 |
| 0 | pincode | int | - | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 0 | state | VARCHAR | 255 | 1 |
| 0 | street\_name | VARCHAR | 255 | 1 |

**Table 3: Orders**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| 0 | contact\_number | VARCHAR | 10 | 1 |
| 0 | Delivery\_date | DATE | - | 1 |
| 0 | order\_time | DATE | - | 1 |
| 0 | price | DOUBLE | - | 0 |
| 0 | Receiver\_name | VARCHAR | 255 | 1 |
| 0 | status | ENUM | - | 1 |
| 0 | tracking\_id | VARCHAR | 255 | 1 |
| 0 | weight | DOUBLE | - | 0 |
| 1 | from\_warehouse\_id | BIGINT | - | 0 |
| 1 | delivery\_agent  \_id | BIGINT | - | 1 |
| 1 | To\_warehouse\_id | BIGINT | - | 1 |
| 1 | sender\_id | BIGINT | - | 0 |

**Table 4: Routes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| 0 | Arrival\_date | DATE | - | 1 |
| 0 | Dispatch\_date | DATE | - | 1 |
| 0 | status | ENUM | - | 1 |
| 1 | from\_wid | BIGINT | - | 0 |
| 1 | order\_id | BIGINT | 10 | 0 |
| 0 | to\_wid | BIGINT | - | 0 |

**Table 5: warehouse**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 3 | id | BIGINT | - | 0 |
| UNI | location\_id | BIGINT | - | 1 |
| UNI | manager\_id | BIGINT | - | 1 |

**Table 6: warehouse\_delivery\_agents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/Constraint** | **Column Name** | **Data Type** | **Lengt h** | **Allow Null (1=Yes; 0=No)** |
| 1 | warehouse\_id | BIGINT | - | 0 |
| 3 | delivery\_agents\_id | BIGINT | - | 0 |

**E-R Diagram,Dataflow diagram and Class Diagram:**

Go to Appendix A

# CODING STANDARDS IMPLEMENTED

## Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | User, Order, UserController | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type  prefixes like ‘C’ for class. |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Camel | SignUp, SignIn,  addReviews | Methods should use **verbs** or verb  phrases. |
| Parameter | Camel | firstName, lastName, email, password | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in  most scenarios. |
| Interface | Pascal with "I" prefix | UserRepository, OrderRepository | Do not use the ‘\_’ sign |
| Annotation | Pascal | SpringBootAppli  cation | Use @ at start of annotation |
| DTOs | Camel | ApiResponseDTO, SignUpReqDTO, OrderDetailsRes DTO | Use to transfer data between the processes |
| Exception Class | Pascal with "Exception"  suffix | ResourceNotFoun dException |  |

## Comments

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines.

End-line comments are

comments that follow code on a single line.

* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

# TEST REPORT

**Another group called Linux did the testing and the report of the testing is given hereunder.**

# GENERAL TESTING:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | SignUp Page | Signup  successfully message | OK | Nothing |
| 2 | SignIn Page | Pop-up will come | Ok | Please enter username and password again . |
| 3 | HomePage | Vendor List fetched from db | Ok | Failed to fetched vendor list |
| 4 | Menu List page | Gives all menu item for selected  vendor | Ok | Nothing |
| 5 | Cart Page | Give a list of  added item in the cart | Ok | Please log in first |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | Customer Order history | Order history list render successfully | Ok | Failed to fetched orders |
| 7 | Add Menu Item | Menu added successfully | Ok | Nothing |
| 8 | See placed orders by vendor | Placed order list view | Ok | No orders is  placed |
| 9 | Generate Token after first  login | Token generated successfully | Ok | Failed to generate token |
| 10 | View all  users list by admin | Seeing the list of all users | Ok | Nothing |
| 11 | Placing Order by customer | Order placed successfully and redirect to home page | Ok | Nothing |
| 12 | Logout | It will logout from user profile. | Ok | Nothing |
|  | **STATIC TESTING** |  |  |  |
| **SR-NO** | **Deviation** | **Program** |  |  |
| 1 | Commenting not followed | All Web Application |  |  |

1. **PROJECT MANAGEMENT RELATED STATISTICS**

## DATE WORK PERFORMED SLC PHASE Additional Notes

April 11,

2024

April 17,

2024

Project Allotment and User Requirements Gathering

Initial SRS Document Validation and Team

Feasibility Study

Requirement Analysis

Our team met the client Mr. Nitin Kudale (CEO, SIIT Pune) to know his requirements.

The initial SRS was presented to the client to

Structure Decided

(Elicitation) understand his requirements

better.

April 30, Designing the use- Requirement Database Design completed.

## DATE WORK PERFORMED SLC PHASE Additional Notes

2024 cases, Class Diagram, Collaboration Diagram, E-R Diagram, and User Interfaces

Business Logic May 5, 2024 Component Design

Started

Analysis & Design Phase

Design Phase

---------------------------

-

June 16, 2024 Coding Phase Started Coding Phase 70% of Class Library

implemented.

June 17, 2024

Implementation of Web Application and Window Application Started

Coding Phase

Class Library Development going on.

June 18, 2024 Off Off Off

June 19, 2024

June 21, 2024

Implementation of Web Application and Window Application Continued

Implementation of Web Application and Window Application Continued

Coding Phase and Unit Testing

Coding Phase and Unit Testing

|  |  |  |
| --- | --- | --- |
| August 2024 | 10, | After Ensuring Proper Functioning the Required Validations were Implemented |
| August | 12, | The Project was Tested by the respective Team |
| 2024 |  | Leaders and the Projec Manager |
|  |  | The Project was Submitted to Other |

Coding Phase and Unit Testing

Testing Phase (Module

Class Library Modified as per the need.

---------------------------

-

Module Integration was done by the Project Manager

---------------------------

t Testing) -

|  |  |  |
| --- | --- | --- |
| 2024 |  | Project Leader of Oth  Project Group For Testing |
| August 2024 | 14, | The Errors Found were Removed |
| August 2024 | 16, | Final Submission of Project |

August 13,

Testing Phase The Project of Other Team

er (Acceptance was Taken up by the Team Testing) for Testing

Debugging The Project was complete

for submission

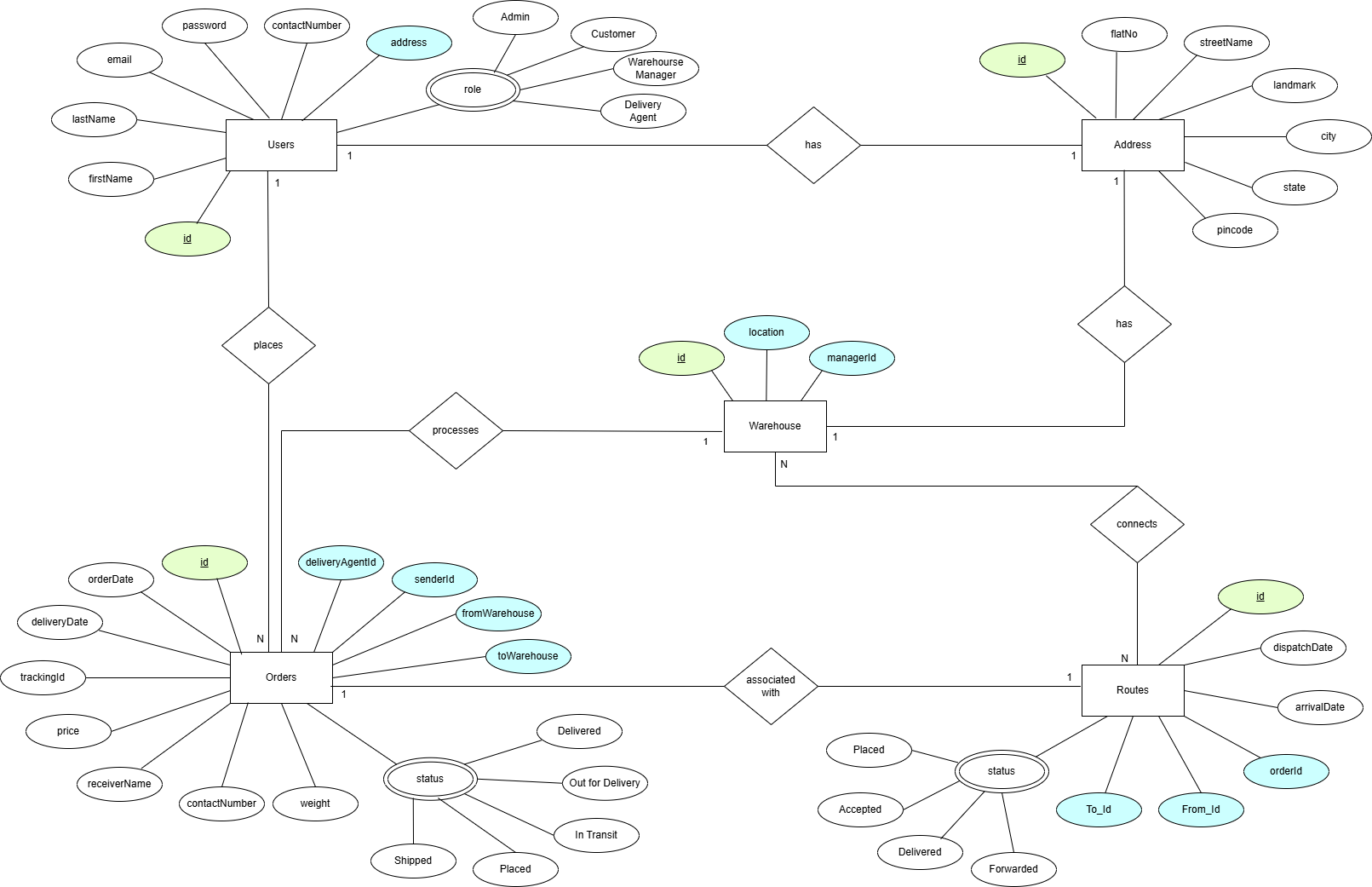
-------------

--

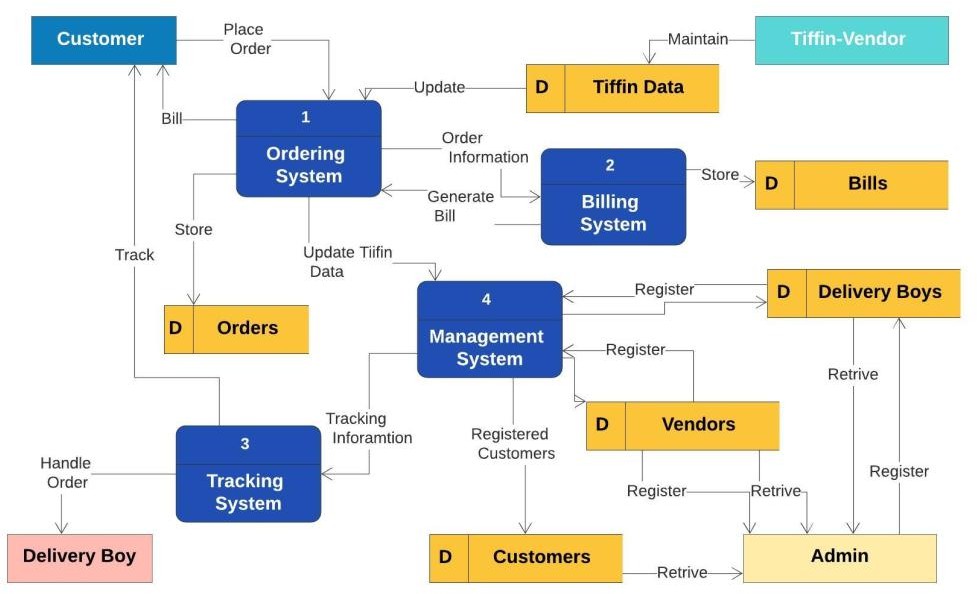
-

**Appendix A**

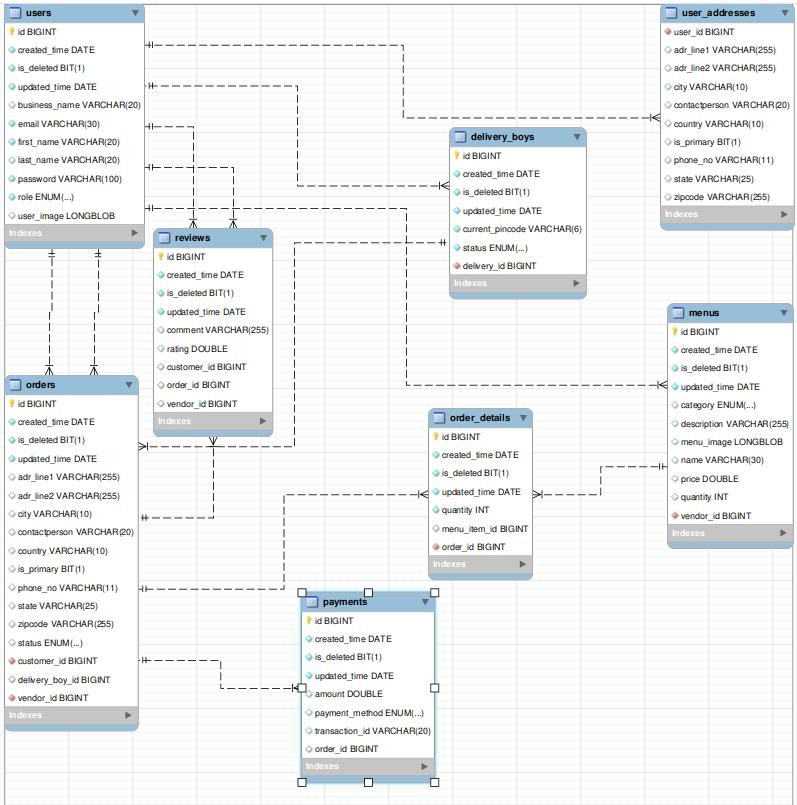
**Entity Relationship Diagram**



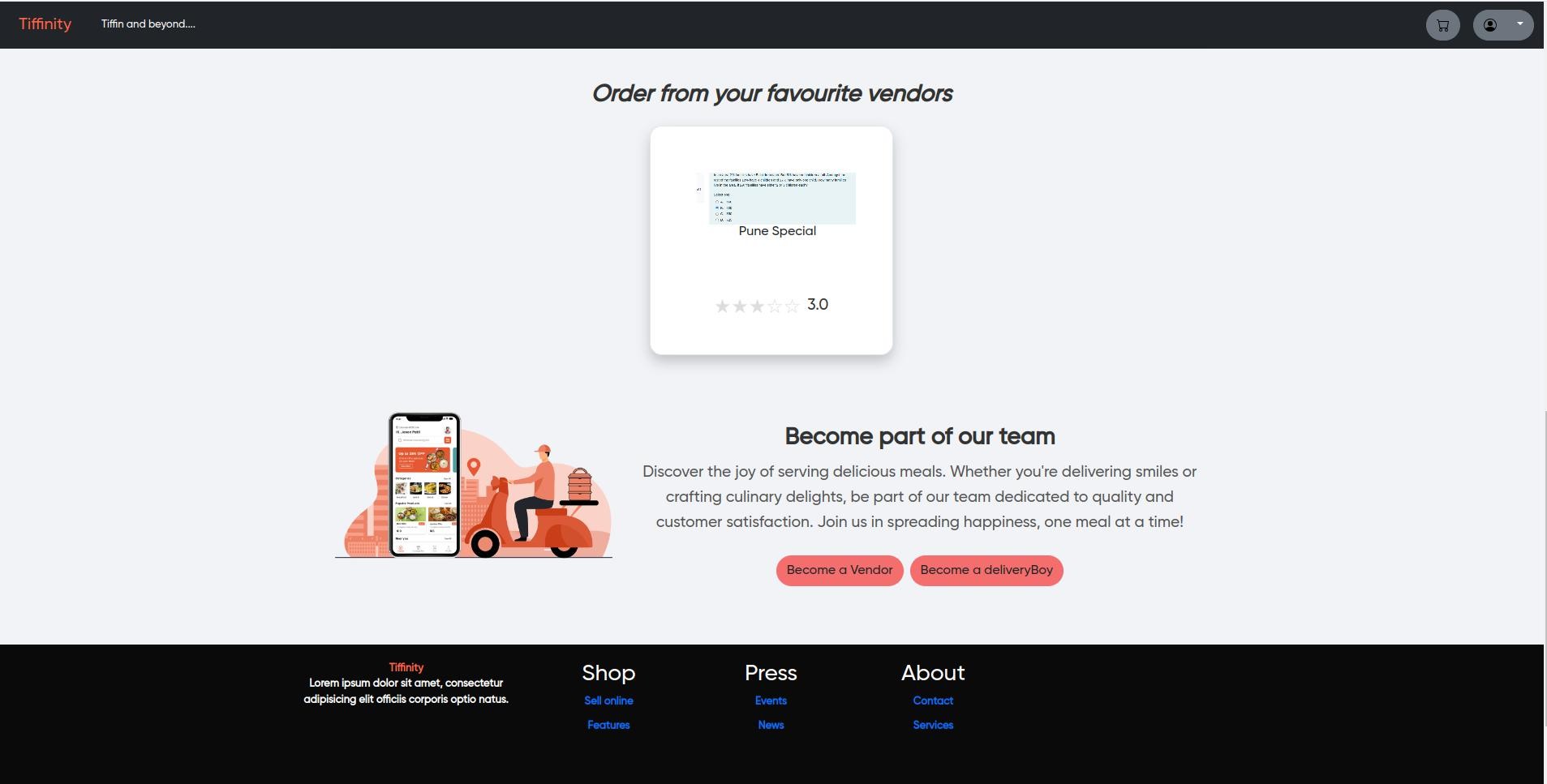
**Data Flow Diagram:**



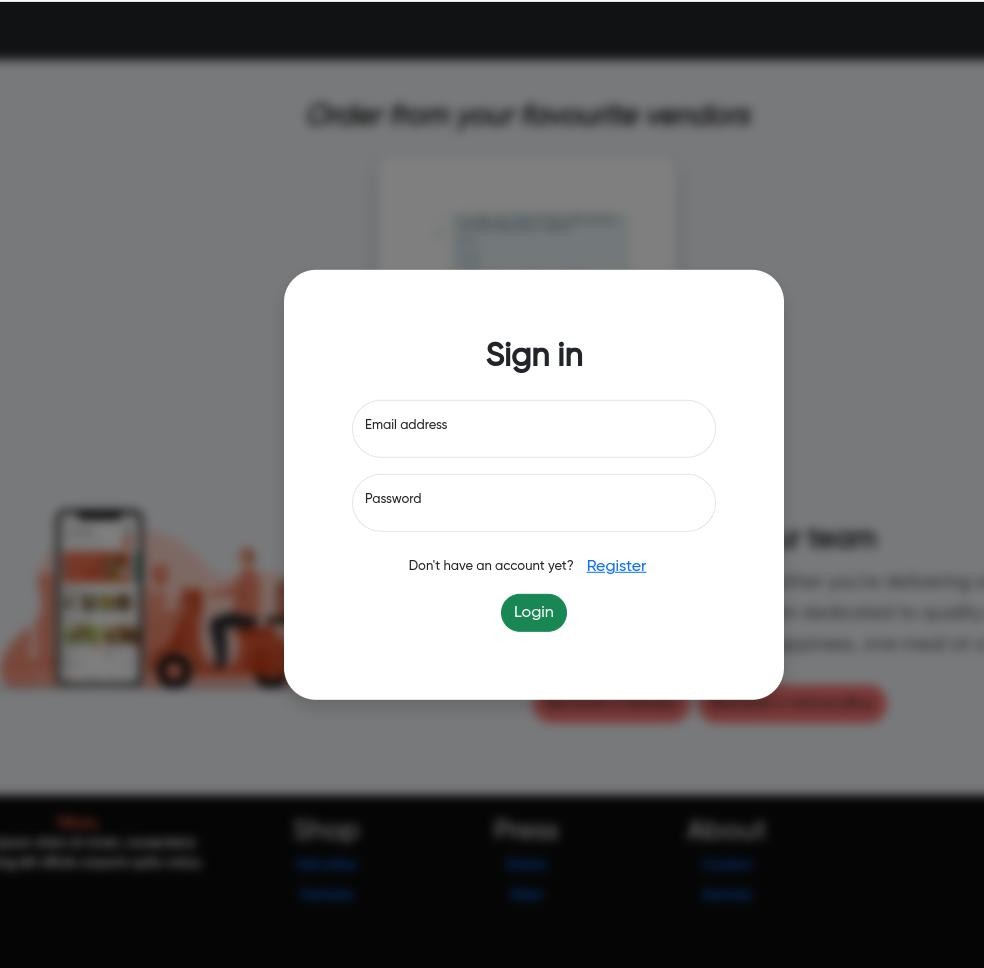
**Class Diagram**



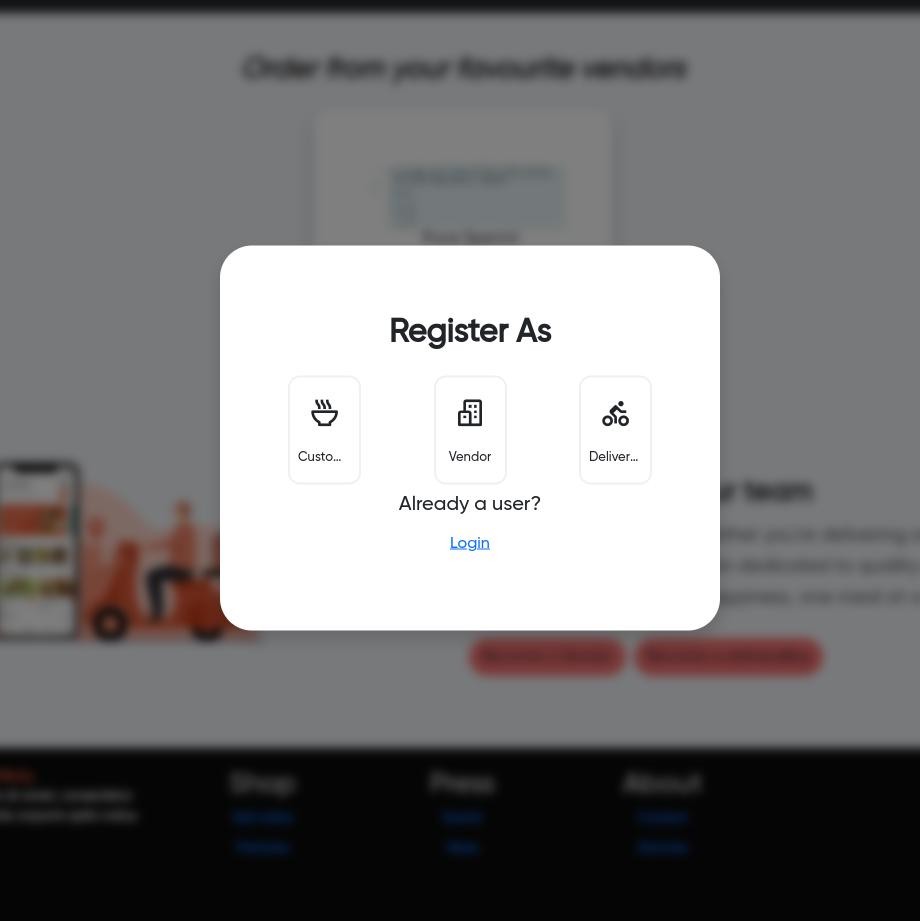
**Appendix B Homepage:**



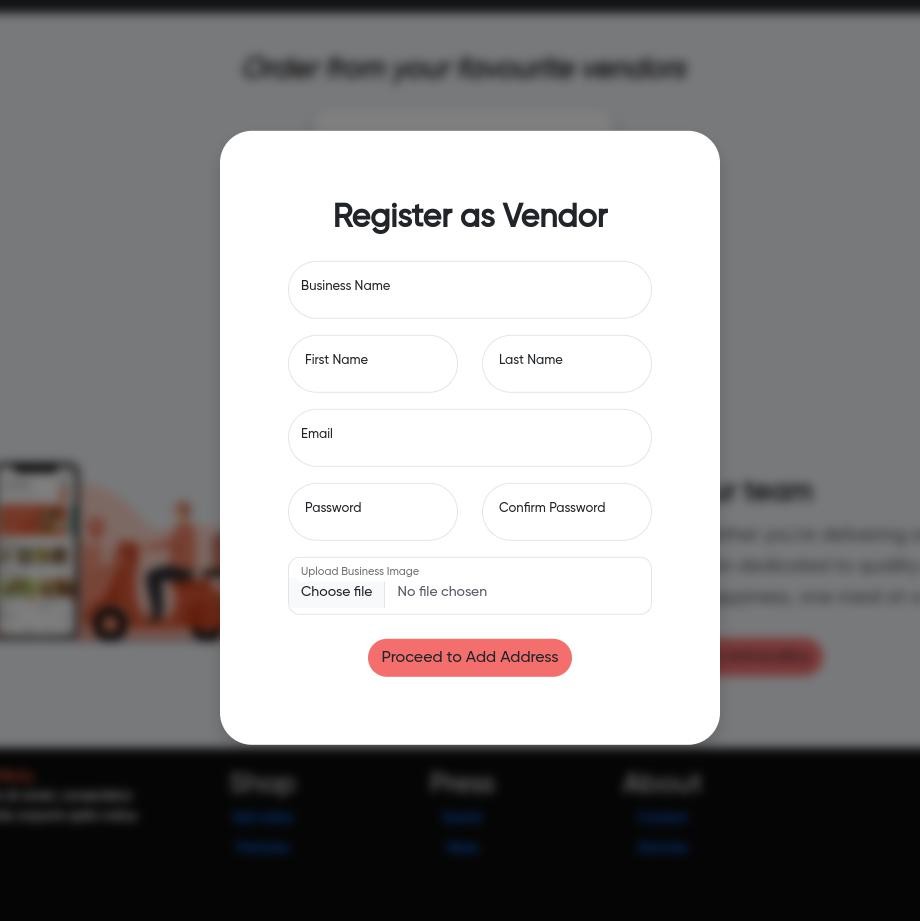
**LoginPopup:**



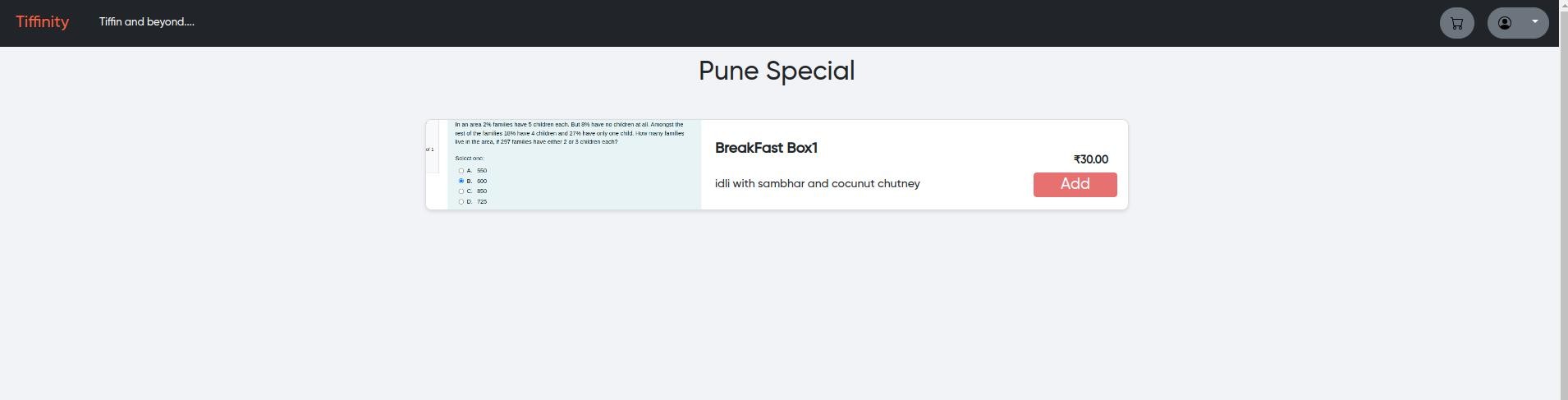
**SignUpPopup:**



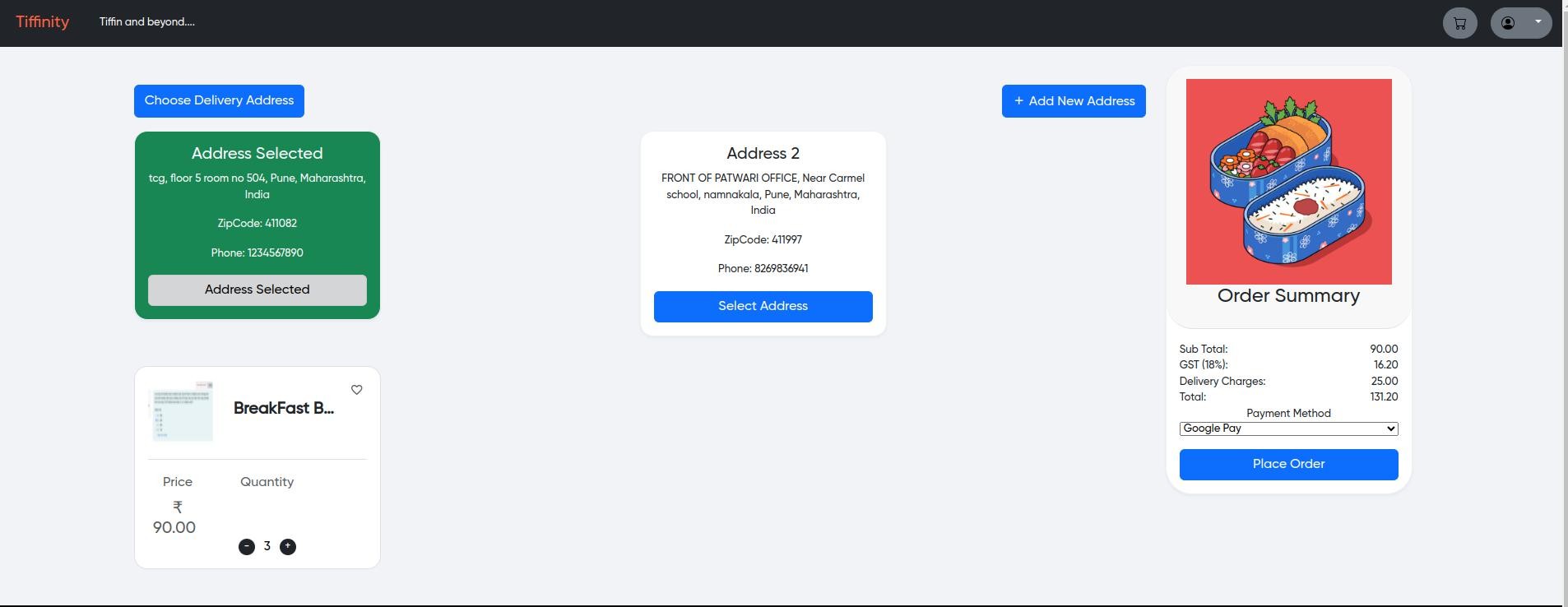
**Register Pop:**



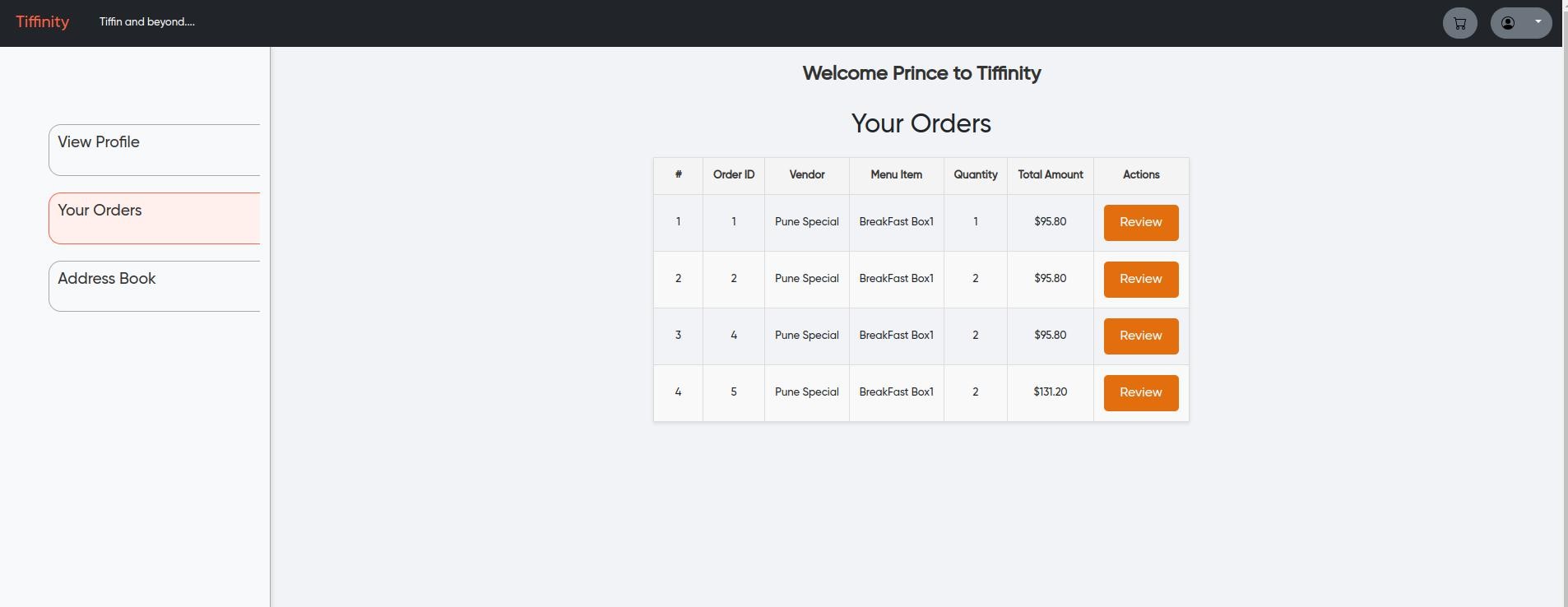
**Menu Page**



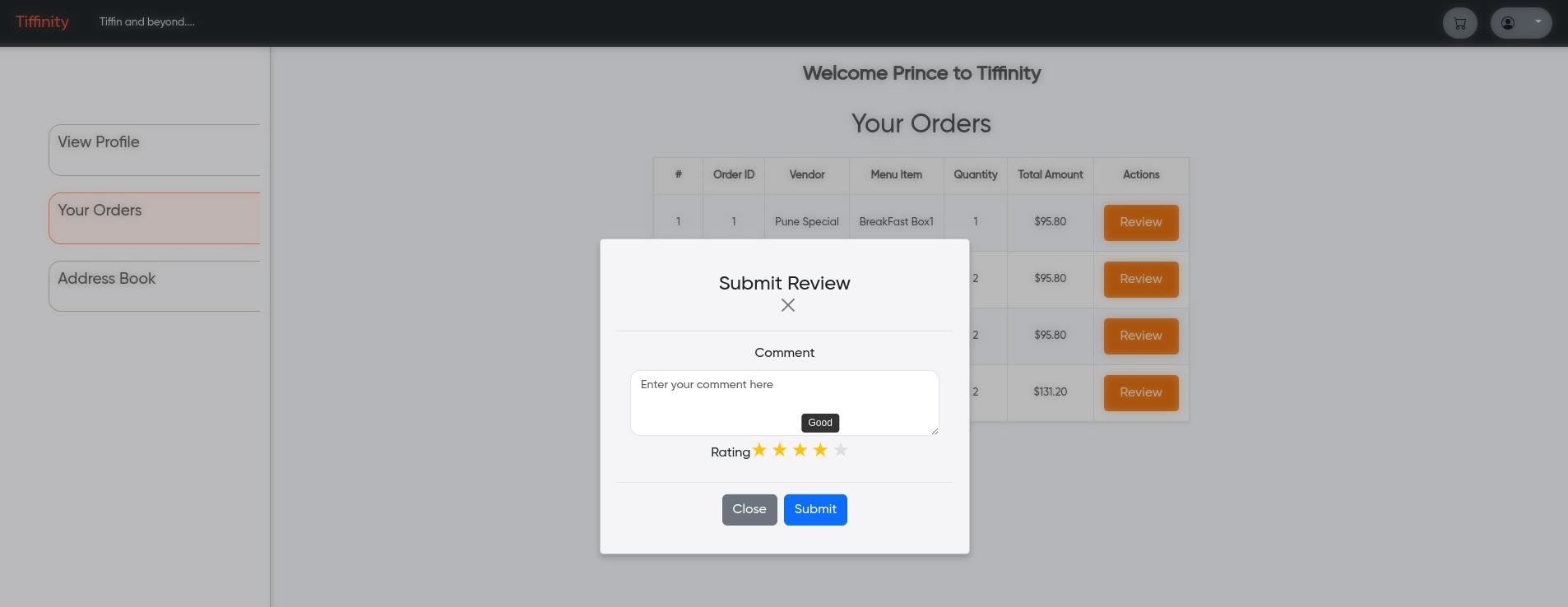
**Cart Page**



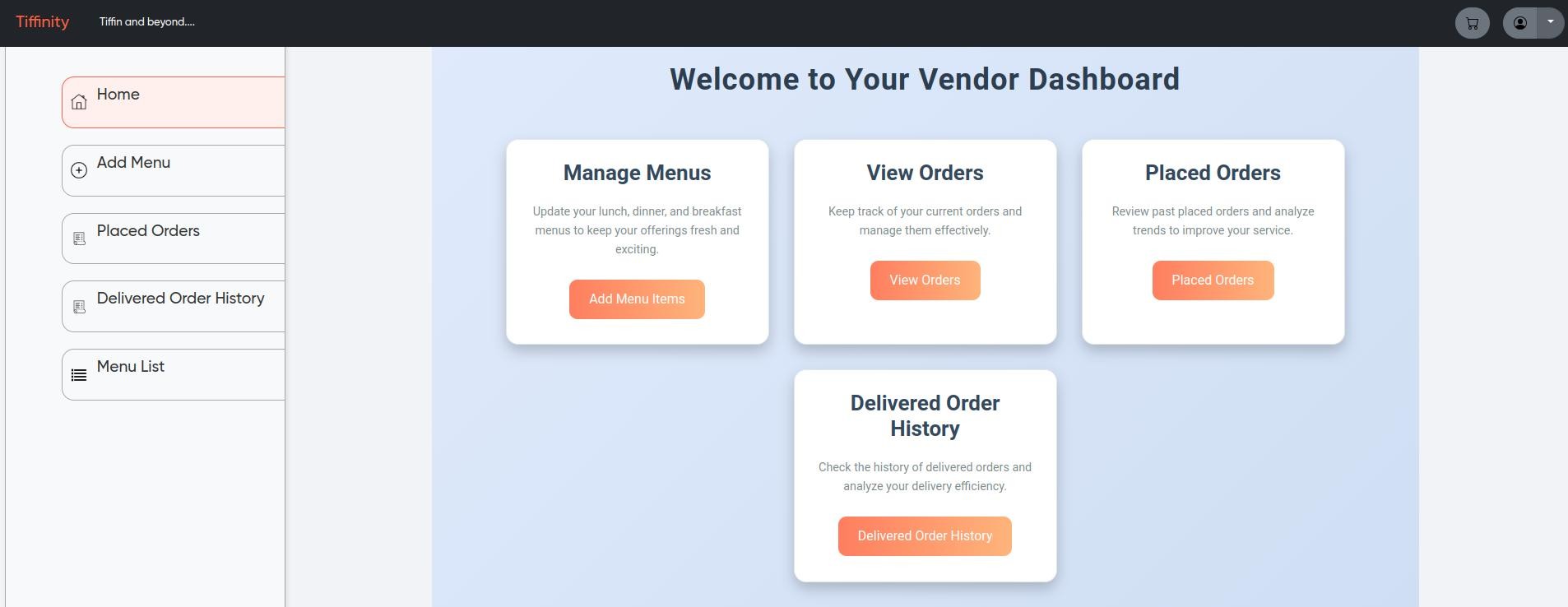
**Customer Profile Page**

****

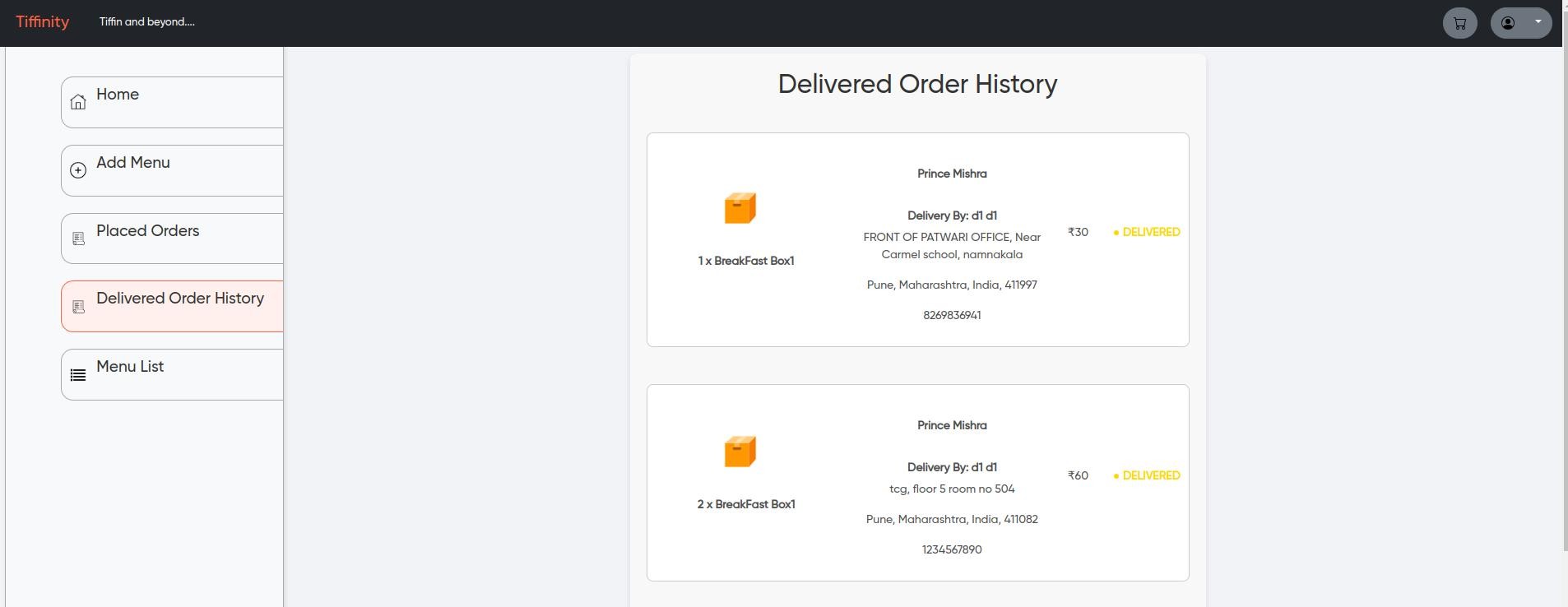
**Review Modal**

****

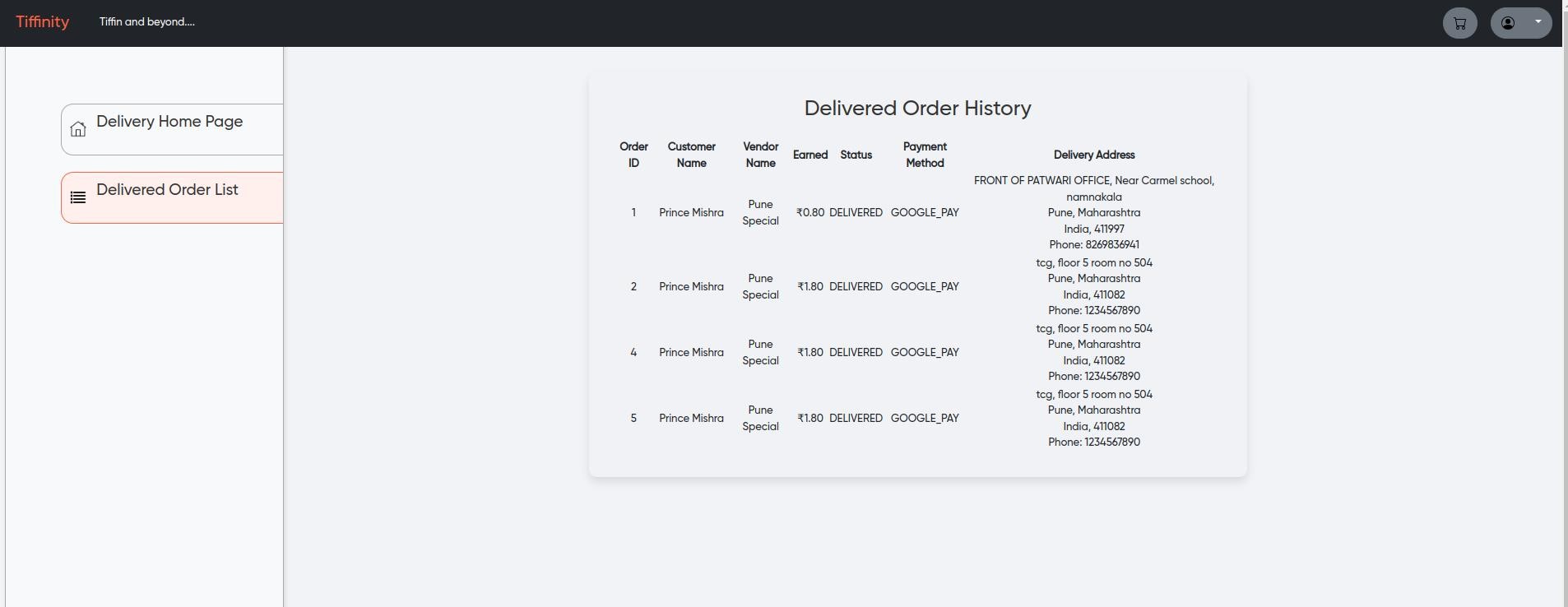
**Vendor Home Page**



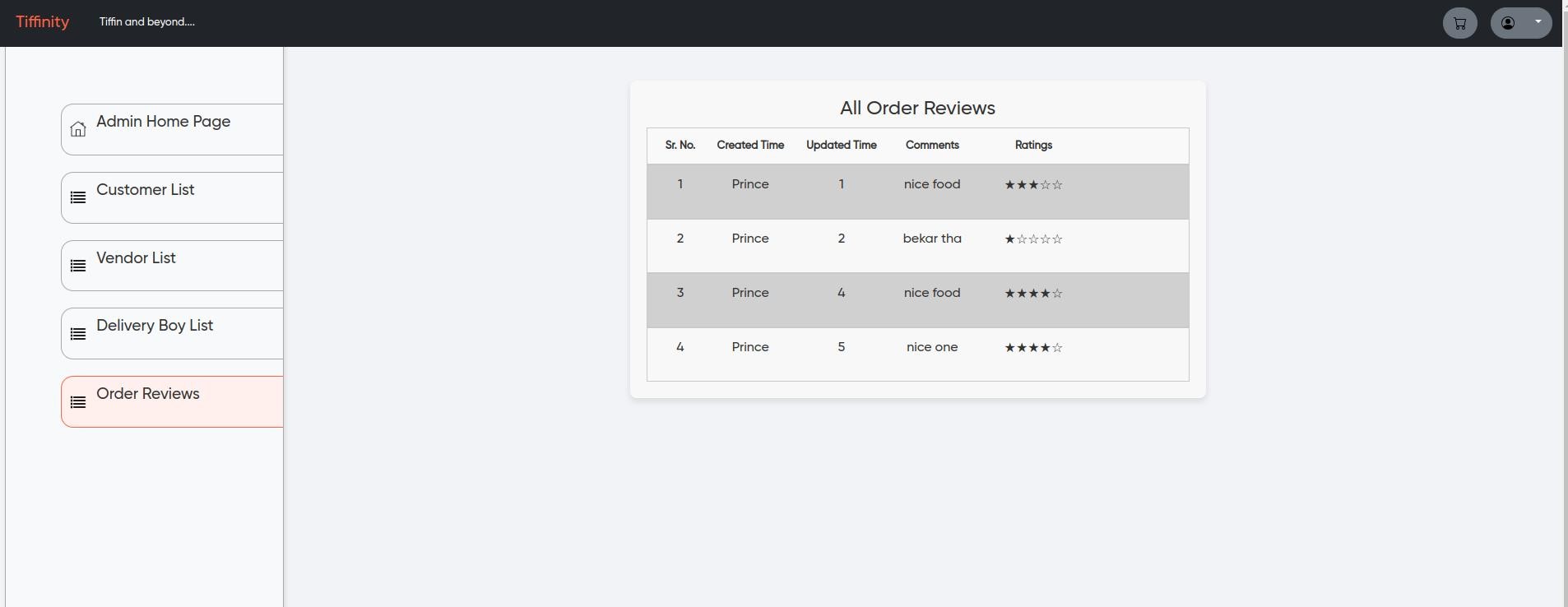
**Vendor Delivered Order History**



**Delivery Boy Delivered Ordered History**



**Admin**



# REFERENCES

## Spring Boot Documentation

URL: <https://spring.io/projects/spring-boot>

## React.js Documentation

URL: https://reactjs.org/docs/getting-started.html

## Redux Documentation

URL: https://redux.js.org

## Java Programming Language

URL: <https://www.oracle.com/java/>

## MySQL Workbench Documentation

URL: <https://dev.mysql.com/doc/workbench/en/>

## Spring Boot with React and Redux

URL: https://[www.baeldung.com/spring-boot-react-and-redux](http://www.baeldung.com/spring-boot-react-and-redux)

## Java Persistence API (JPA) Documentation

URL: https://[www.eclipse.org/eclipselink/documentation/2.7/](http://www.eclipse.org/eclipselink/documentation/2.7/)

## Swagger Documentation for Spring Boot

URL: <https://springdoc.org/>

## MDN Web Docs

URL: <https://developer.mozilla.org/>

## React Redux Integration Guide

URL: https://react-redux.js.org/