Community Aid: UniHub

An Engineering Project in Community Service

Final Report

Submitted by

Team Members List

1. 20BAI10039 Ritwik Agarwal
2. 20BAI10095 Divyansh Yadav
3. 20BAI10301 Yuvraj Chopra
4. 20BCE10496 Amisha Goyal
5. 20BCE10526 Kirti
6. 20BCE10844 Varun Mishra
7. 20BCY10126 Trilok Dhakad
8. 20MIP10045 Telugunti Akhil

in partial fulfillment of the requirements for the degree of

Bachelor of Engineering and Technology

VIT®
BHOPAL
www.vitbhopal.ac.in

VIT Bhopal University
Bhopal
Madhya Pradesh

May 2023

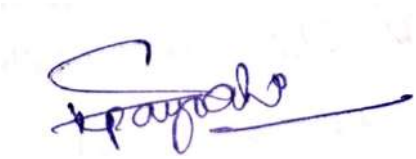
VIT BHOPAL
www.vitbhopal.ac.in

Bonafide Certificate

Certified that this project report titled **"Community-aid e-commerce: Virtual Store"** is the bonafide work of "20BAI10039 Ritwik Agarwal, 20BAI10095 Divyansh Yadav, 20BAI10301 Yuvraj Chopra, 20BCE10496 Amisha Goyal, 20BCE10526 Kirti, 20BCE10844 Varun Mishra, 20BCY10126 Trilok Dhakad, 20MIP10045 Telugunti Akhil" who carried out the project work under my supervision.

This project report (Phase II) is submitted for the Project Viva-Voce examination held on 14 May 2023.

Dr. Subhash Chandra Patel (Supervisor)



Dr. Susant Kumar Panigrahi (Reviewer 1)

Dr. Hemlata Gangwar (Reviewer 2)

Table Of Contents

1. INTRODUCTION 3
   1.1. Motivation 3
   1.2. Problem Statement 4
   1.3. Objective 4
2. Existing Work / Literature Review 6
   2.1. For Marketplace as selling platform for vendors and buying platforms for students 6
   2.2. For Reselling platforms: 7
   2.3. For Cab Pooling: 8
   2.4. For Club events: 8
3. Topic of work 9
   3.1. System Design / Architecture 9
   3.3. Architecture Diagram of website: 13
   E-commerce website architecture: 13
   Cab pooling website architecture: 14
   3.4. Flow Diagrams: 15
   3.5. Code snippets: 17
4. Results and discussions 24
5. Individual Contribution by members 32
6. CONCLUSION 40

Reference: 41

1. INTRODUCTION

The main purpose of this project is to help solve the problems of the community and make their lives easier. So, we thought of this idea as an initiative to help the student and vendor community, with which we can relate the most.

Our aim is to create a comprehensive platform that provides solutions for all the needs of college students, including buying daily necessities, transportation, club and events registration.

Most students relocate to different cities for better education. This involves carrying a lot of luggage. It becomes very difficult to carry basic necessities like mattresses, buckets, blankets etc. Buying these things first hand at a new place is another issue which involves lots of factors like distance of university from markets, vendor monopoly, price fluctuation, quality needs and exact specifications of products such as size of a mattress.

After discussing with batchmates and juniors, we felt that there is a gap which can be filled through this service. Most of the senior students bought these items from the vendors and now want to resell them to incoming juniors. Students are not aware of the vendors who sell these items from the vendors and vendors can't deliver these items inside college premises.

This platform, named UniHub, will act as a bridge between them. It is an online marketplace made using Django, Python and SQLite, for the student community in college. E-commerce site that connects students and local Sehore shop owners to VIT hostel. It is for various vendors, small shopkeepers, cab owners/drivers, and other sellers/students to list their services and products, thereby creating a transparent marketplace that meets the needs of students.

It's just like Amazon, Bigbasket redesigned for the student and vendor community. It will have products from local vendors at transparent prices and multiple vendors will be insured so that students willing to buy new or used stuff will not have to face vendor monopoly and abrupt high prices.

1.1. Motivation

We conducted a survey in which we asked freshers to share their experience of their first self-reliant journey from home to college and what problems they faced when they first reached the campus. Many of them complain about lack of access to local markets and nearby shopping areas. They are only dependent on E-commerce giants like Amazon and Flipkart.

We also took a survey from vendors. They also wish to expand their reach among students, as it is a good opportunity for them to expand their business and get into healthy competition.

For Event registration portal, while on our experiential learning, in VIT Chennai we came across their events website, where a user can register for any college event through one platform itself, no need to check the mails, fill google forms etc. So, we thought of bringing that idea to our college also.

This motivated us to work towards a trustworthy solution for all the parties involved and use our skills to help the student community with their day to day needs and vendor community by giving them more options.

1.2. Problem Statement

1. **Unable to carry everything from home, and expensive here:** One of the challenges faced by students is the need to travel frequently, often with a lot of luggage and study materials. It can be difficult to carry basic necessities such as mattresses, buckets, and blankets, and buying these items in every new place can be time-consuming and expensive, monopoly of vendors and limited options, online delivery takes time.

   So, a platform is needed to provide a solution by allowing vendors to list and sell these items to the college community, providing students with more options and saving them from monopoly pricing.

2. **No place to resell at transparent price:** At the time of leaving the college, it is difficult to carry all the stuff with own self, and if someone wants to sell it to juniors then there is no price transparency.

   So, a platform should be there, where students(seniors) can resell also and for juniors all the goods are listed with justified prices.

3. **Limited and expensive options for travel:** Our college is little far away from Bhopal city, students often have to travel from campus to Bhopal or Indore city by cab. There are limited cab owners that have connections with students, there are more taxi drivers and cab owners. Ola/uber are not easily available for intercity travel, even if they are, it is very expensive for one person to bear. Also, it is not so safe for girls to travel alone, forming or finding groups is a difficult and hectic task to do.

   We need a place where contacts and other details of all the local cab/taxi drivers/owners available are there for booking, at the same time, with cab pooling option, where people can connect, share their rides and fare with same travel date, time and place.

4. **Hazard of unique emails and google forms for every single college event:** For every event, club coordinators have to send mails, create google forms, payment links etc. Students also miss out event mails.

   It will be a lot easier for club coordinators to manage data if there is one portal where they can just list their events and event details, everything will be managed from one portal itself. And for other students, they can just check out among events, and can register for events of their interest. solution by providing one platform for all events registration.

1.3. Objective

As problems discussed earlier, we have come up with an idea to solve problems faced by our own student community. We have developed a platform which will allow students to purchase everyday items and edibles, connect with cab owners for ride-sharing, and register for upcoming events.

Overall, the goal of UniHub is to make life easier for students in their community by providing them with a convenient way to purchase necessities, connect with other students for transportation, and stay informed about upcoming events.

UniHub is created with keepings following objectives in consideration:

1. **Selling platform for nearby vendors:** Providing the local vendors a platform to put up their products and also to ensure transparent pricing between different vendors.

2. **Buying platform for students:** Primary objective is to provide a platform where juniors and seniors can connect and conduct transactions with transparency.
3. **Reselling platform for students:** Primary objective is to provide a platform where juniors and seniors can connect and conduct transactions with transparency.
4. **Ease of transactions:** Avoiding the usual nitty gritties of reselling transactions between seniors and juniors.
5. **Accessibility:** Providing three ways access to seniors, vendors and juniors for all kinds of transactions.
6. **Budget friendly, safe and convenient travel facility for students:** The platform will connect students who are looking to share a ride to Bhopal or Indore, which can be expensive for those traveling alone. By sharing the cost of the ride, students can save money and make new connections with their peers. It is more safe and environmentally friendly as well.
7. **All events listing and registration at one place:** All the club events will be listed and at one place only, event registration will also be done at the same website only.

2. Existing Work / Literature Review

2.1. For Marketplace as selling platform for vendors and buying platforms for students

A marketplace is a location where people regularly gather for the purchase and sale of goods. Since e-commerce websites allow us to view things online and make purchases, these solutions are more commodious and comfortable for us.

Online, there are several platforms and websites similar to our proposal, however they are mass produced with practically any kind of product; like Amazon, Flipkart, Big Basket, Blinkit etc.

2.1.1. Amazon

Amazon allows third-party sellers to offer their products on its platform, providing customers with a wider selection of goods to choose from. It has:

1. Online shopping: Amazon offers a wide range of products, including books, electronics, clothing, and home goods, that customers can purchase online through its website or mobile app.
2. Marketplace: Amazon allows third-party sellers to offer their products on its platform, providing customers with a wider selection of goods to choose from.
3. Amazon Fresh: Amazon's grocery delivery service, which allows customers to order fresh produce, meat, and other grocery items online and have them delivered to their doorstep.

Cons:

1. This does not deliver all items (fresh items) to our college and has a high delivery charge on many items.

2. It takes time.

2.2. Grofers

Grofers is an Indian online grocery delivery service that allows customers to purchase groceries and other household essentials through its website or mobile application.

Cons: Not available for VIT Bhopal campus.

2.3 BigBasket

Bigbasket is an online grocery delivery platform that allows customers to order groceries and household essentials from the comfort of their homes. Bigbasket offers a convenient and reliable way for customers to order groceries and household essentials online, with a wide range of products, fast delivery options, and attractive offers and discounts. Bigbasket offers a wide range of products including fruits, vegetables, dairy, breads, snacks, beverages, personal care, and household essentials.

1. Express delivery: Bigbasket offers express delivery in select cities, allowing customers to receive their orders within 90 minutes.
2. Scheduled delivery: Customers can also choose to schedule their deliveries for a convenient date and time

Cons: Not available for VIT Bhopal campus.

2.2. For Reselling platforms:

2.2.1. Resell it

One of our seniors made a project for selling or reselling and buying mattresses with WhatsApp bot. Resell it had its own pricing model also.

Cons:

1. This project was created only for sales of mattresses.

2. It was limited to boys' hostels only.

2.2.2. OLX:

OLX (Online Exchange) is an online consumer-to-consumer(C2C) advertising platform. The platform, which can be accessed on desktop, mobile-App, and mobile-web, offers a free, fast, and hyper-local way for both individuals and businesses to buy and sell goods such as pre-owned cars and motorbikes, second-hand mobile phones, and pre-loved personal and household goods, and services online. It has hundreds of millions of users every month across five continents, and in 2012, OLX received the status of the largest online ad resource in India. OLX pioneered the concept of C2C classifieds in India, and already has 80% market share of the C2C online trade in the country.

Users of OLX.ua can respond to an already-posted advertisement by calling or chatting with its creator on the website, or can pre-register via phone, email, or social network accounts in order to post their advertisement with offer description, images, and contact information included in the announcement submission form. On OLX.ua, adverts are often arranged according to the date they were posted. It also provides delivery service which involves blocking money on the buyer's card and writing it off only after receiving the goods.

Selling and buying through OLX also makes the planet greener with OLX India transactions proven to have saved 3.3 million tons of greenhouse gas emissions in 2015-16.[1]

Cons: One of the major and most important concerns while using OLX is risk of fraud. Several cases of selling stolen items, delivering damaged products have been reported.

2.2.3. FB marketplace:

Facebook Marketplace is the social network's classified ads section on Facebook that focuses on assisting people and companies sell goods locally, where people can list items for sale, businesses can show inventory for retail items, property rentals and vehicles and event tickets also user can browse listings, buy and sell items, search for items for sale in their area or find products available for delivery. Facebook leverages the Marketplace through its social network. Facebook users often already have an account and FB marketplace is used on the Facebook app itself. Users can quickly get set up to buy or sell from their existing accounts. Buyers and sellers must arrange for payment and delivery between themselves. Someone can set up a shop with your business Page and sell as a business on Marketplace. However, Facebook plays no role in the actual transaction, and buyers and sellers must arrange for payment and delivery between themselves. For sellers, a vast audience of 2.2 billion users and the ease for users to browse while already on Facebook are big draws [2].

Cons:

1. Businesses and commercial sellers are not allowed to create listings on the Marketplace.

2. Users can only report sellers for items sold not as advertised on Facebook Marketplace. Someone doesn't have to give their email addresses or phone numbers for communication. Some unscrupulous individuals may set up accounts specifically to defraud other users. The buyer must beware of frauds.

2.3. For Cab Pooling:

2.3.1. Ola and Uber

Ola and Uber are ride-hailing services that offer a range of features and functionalities such as mobile app for cab booking, real-time tracking, fare estimates, cashless payment options, safety features, and additional services.

1. Ride Booking: Customers can book a ride using the mobile app by entering their pickup location and destination. They can also choose the type of car they want, such as a sedan, hatchback, or SUV.
2. Driver Information: Once the ride is booked, the customer receives information about the driver, including their name, phone number, and vehicle details.
3. Real-time Tracking: The mobile app provides real-time tracking of the driver's location, allowing the customer to track the driver's progress towards the pickup location.
4. Fare Estimate: The app provides a fare estimate for the ride, which is calculated based on the distance and time taken to complete the journey. This allows customers to have an idea of how much the ride will cost before they book it.
5. Cashless Payment: Customers can pay for the ride using the app, which supports cashless payment methods such as credit/debit cards, digital wallets, and net banking. Cash payment options are also available.
6. Safety Features: Both Ola and Uber have safety features such as GPS tracking, driver rating, and emergency contact options, which ensure a safe and secure ride experience for customers.

2.4. For Club events:

2.4.1. VIT Chennai events website: All the VIT Chennai events are listed at this website by respective coordinators, and other students register and pay for events on the same website.

Thus, this makes life of all club coordinators and students very easy. As any students who want to in a particular event doesn’t need to search a google form in his mail box. He just has to visit a particular site.

3. Topic of work

Since the advent of e-commerce, companies have been able to use the Internet to cut expenses related to purchasing, manage supplier relationships, streamline logistics and inventories, generate strategic advantage, and carry out business re-engineering successfully. E-commerce gives businesses the chance to enhance service offerings and supply chain communications, creating opportunities for competitive differentiation.

As there are plenty of online marketplace platforms that are on mass/global level, but there is no such platform for student and college localities, also there is a big risk of frauds on such big platforms. On your platform, we will list local vendors or shop owners only after meeting them personally and verifying information about them.

3.1. System Design / Architecture

E-commerce design is the method of shaping the coding. We have created our website using HTML, CSS, Bootstrap, Python and SQLite for designs, interfaces and information to a system to satisfy the needs.
System Module Description:
• Coding
• Seller
• Customer
• Management

Project Prerequisites: We will use the following technologies:

3.1.1 Front-end technologies:

1. HTML (HyperText Markup Language): HTML is the standard markup language for Web pages that displays content on web page.
2. CSS (Cascading Style Sheets): CSS is the language we use to style an HTML document and create beautiful styles. CSS describes how HTML elements should be displayed.

- Bootstrap: Bootstrap is a free and open-source CSS framework directed at responsive web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

3.1.2 Back-end technologies:

1. Python: High-level programming language.

- Python Pillow Library (PIL): Adds image processing capabilities to Python interpreter. This library provides fast access to data stored in a few basic pixel formats, extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

2. Django: Django is a free and open-source high-level Python web framework that encourages rapid development and clean, pragmatic design. It is fast, secure and scalable. It comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

- Django REST (Representational State Transfer: a standardized way for building and communicating with web services) Framework (DRF): A package built on the top of Django to create web APIs (Application Programming Interface, that defines the interaction

between different software components and is responsible for fetching or updating the records based on the database queries.). It provides the most extensive features of Django, Object Relational Mapper (ORM), which allows the interaction of databases in a Pythonic way.

Django REST framework made serialization super easy: Hence the Python object can't be sent over the network, so we need to translate Django models into the other formats like JSON, XML, and vice-versa.

DRF allows us to represent their functionality in the Django application in the form of REST APIs.

Database:

Language- SQLite (Structured Query Language): SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQL lets you access and manipulate databases.

Module- DB Browser: DB Browser for SQLite (DB4S) is a high quality, visual, open-source tool to create, design, and edit database files compatible with SQLite.

DB4S is for users and developers who want to create, search, and edit databases. DB4S uses a familiar spreadsheet-like interface, and complicated SQL commands do not have to be learned.

Controls and wizards are available for users to:

- Create and compact database files
- Create, define, modify and delete tables
- Create, define, and delete indexes
- Browse, edit, add, and delete records
- Search records
- Import and export records as text
- Import and export tables from/to CSV files
- Import and export databases from/to SQL dump files
- Issue SQL queries and inspect the results
- Examine a log of all SQL commands issued by the application
- Plot simple graphs based on table or query data.

3.2. Working Principle

UniHub is designed following MVT architecture.
MVT (Model View Template) is a collection of three important components Model View and Template.

1. **Model**

The Model is a data access layer which handles the data and database. The model provides data from the database. The models are usually located in a file called models.py.

In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.

The most common way to extract data from a database is SQL.

Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.

2. **View**

A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result. The views are usually located in a file called views.py.

The View is used to execute the business logic and interact with a model to carry data and render a template.

3. **Template**

The Template is a presentation layer which handles the User Interface (UI) part completely. A template is a file where you describe how the result should be represented.

Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results.

Django uses standard HTML to describe the layout, but uses Django tags to add logic:

<h1>My Homepage</h1>

My name is {{ firstname }}.

The templates of an application are located in a folder named templates.

- **URLs**

Django also provides a way to navigate around the different pages in a website. When a user requests a URL, Django decides which *view* it will send it to. This is done in a file called urls.py.

MVT Architecture:

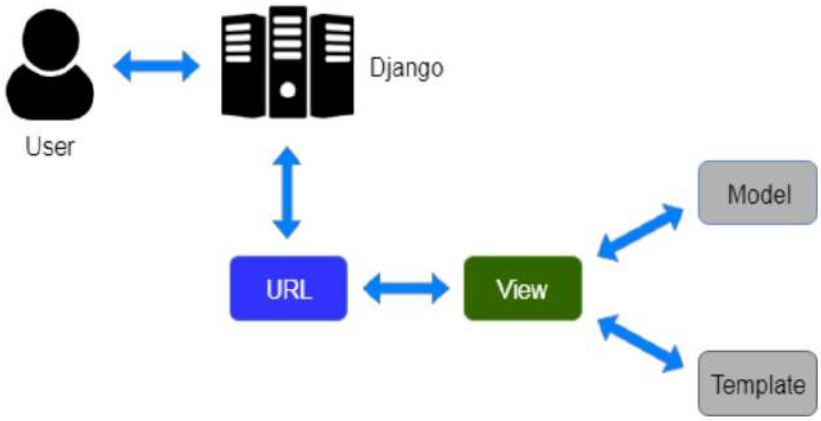


Fig 3.1: MVT Architecture

MVT working:

After installing Django and creating first Django web application, and the browser requests the URL, this is basically what happens:

1. Django receives the URL, checks the urls.py file, and calls the view that matches the URL.
2. The view, located in views.py, checks for relevant models.
3. The models are imported from the models.py file.
4. The view then sends the data to a specified template in the template folder.
5. The template contains HTML and Django tags, and with the data it returns finished HTML content back to the browser.

3.3. Architecture Diagram of website:

E-commerce website architecture:

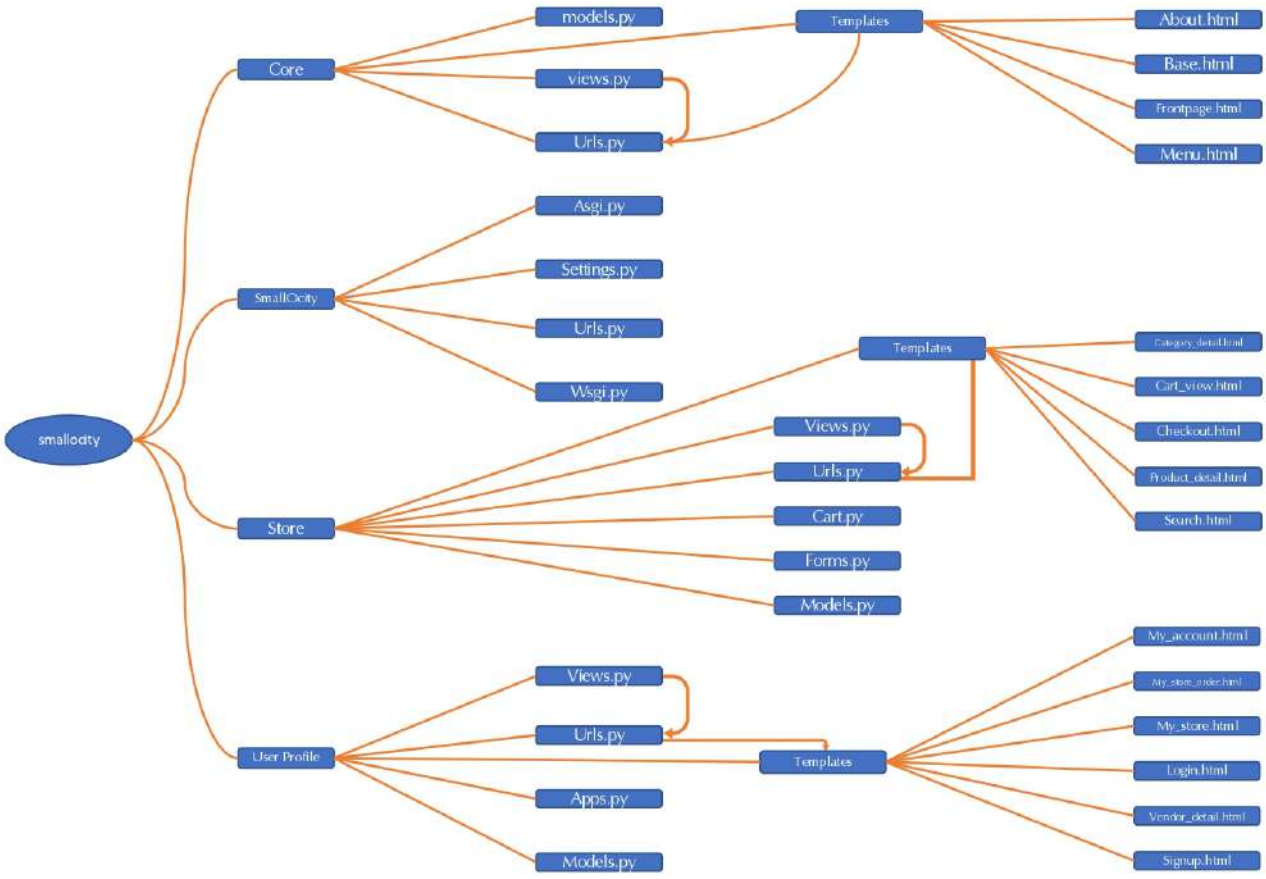


Fig 3.2. E-commerce website architecture

Cab pooling website architecture:

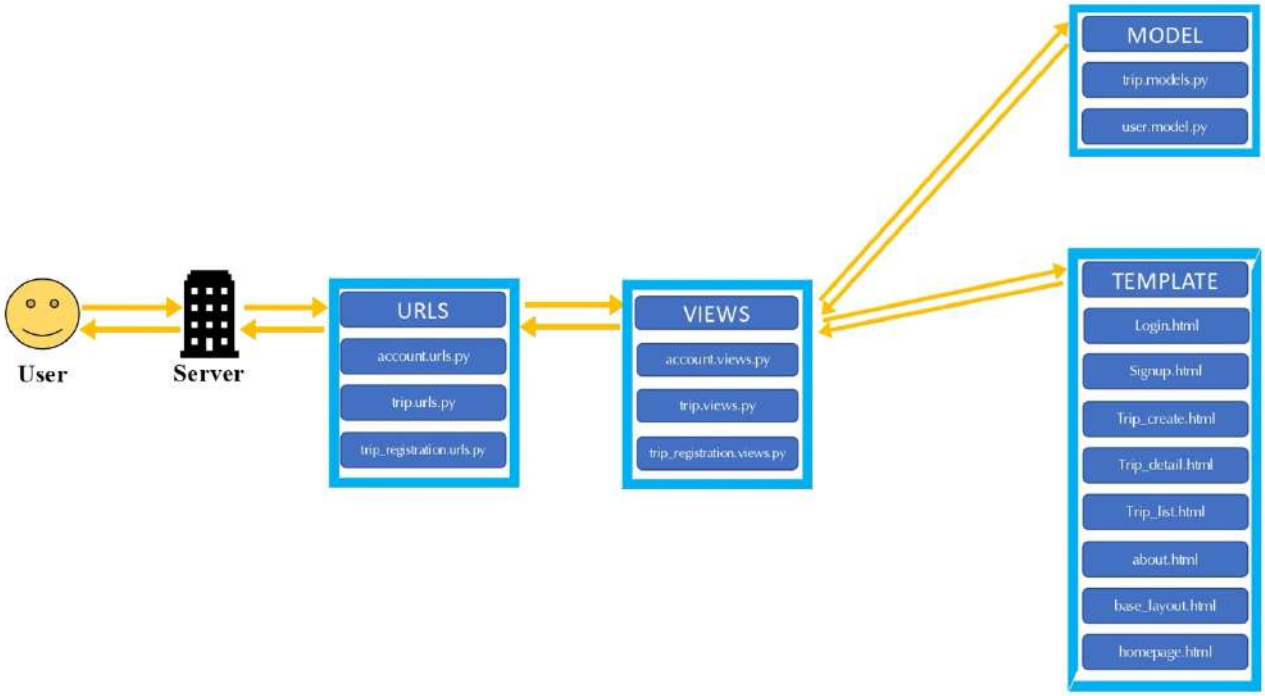


Fig 3.3 Cab pooling website architecture

Event management portal architecture:

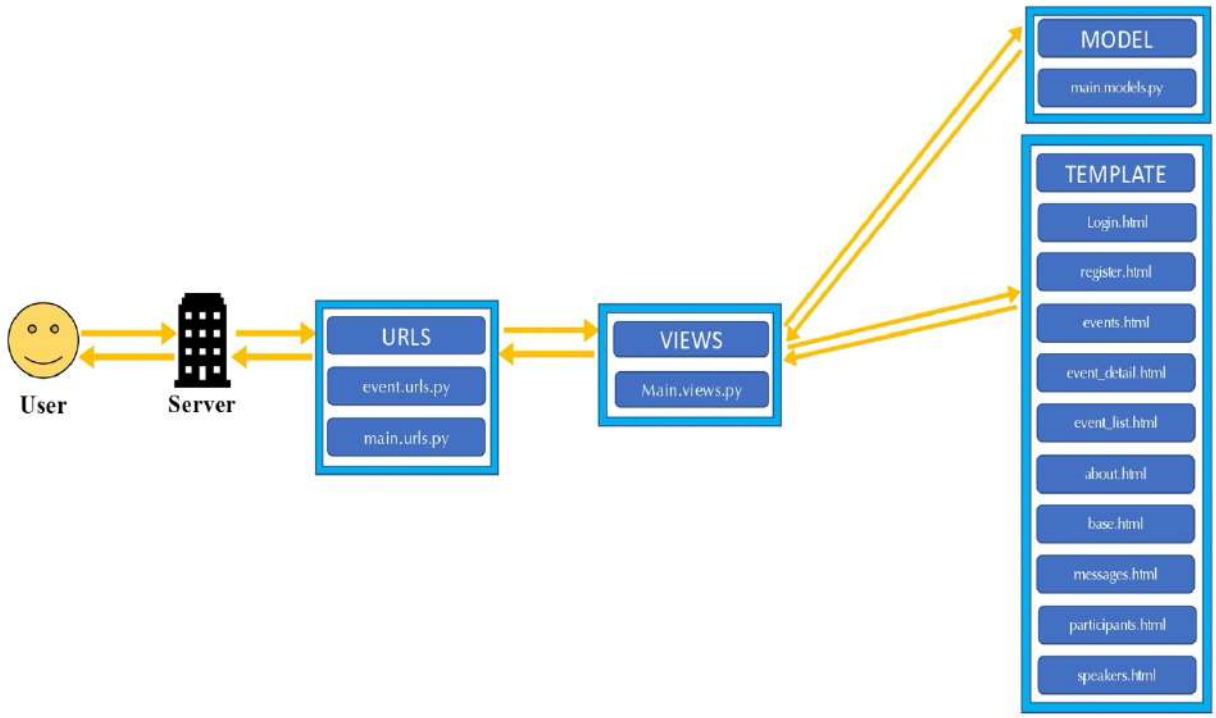


Fig 3.4 Event management portal architecture

3.4. Flow Diagrams:

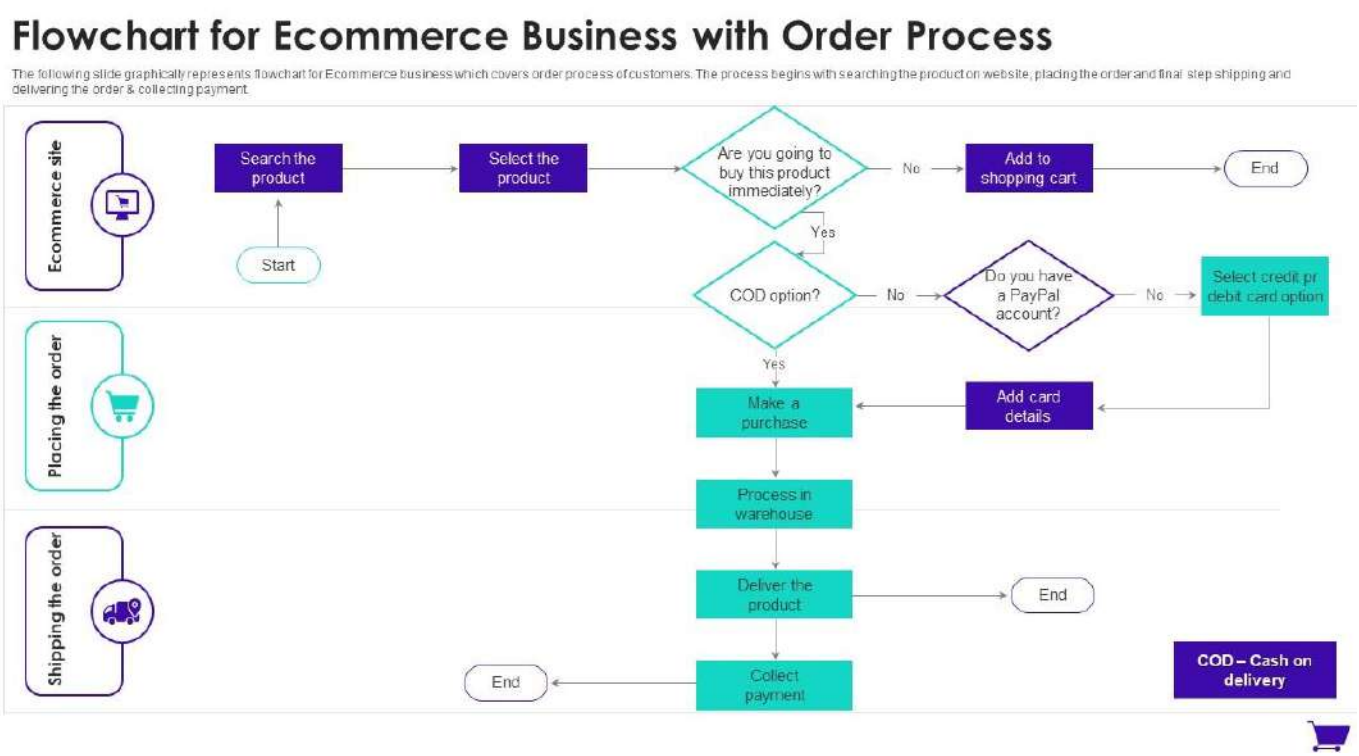


Fig 3.5 Flow diagram of order processing on e-commerce website

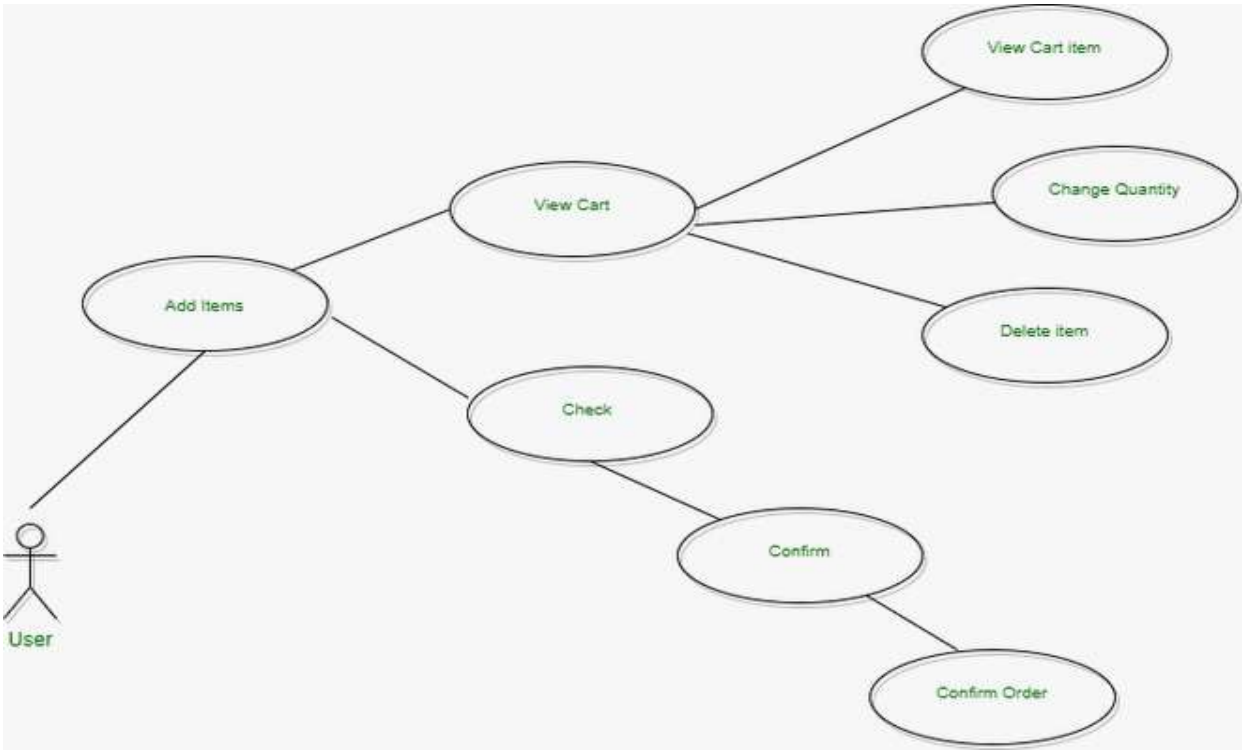


Fig 3.6 Flow of process for buyer

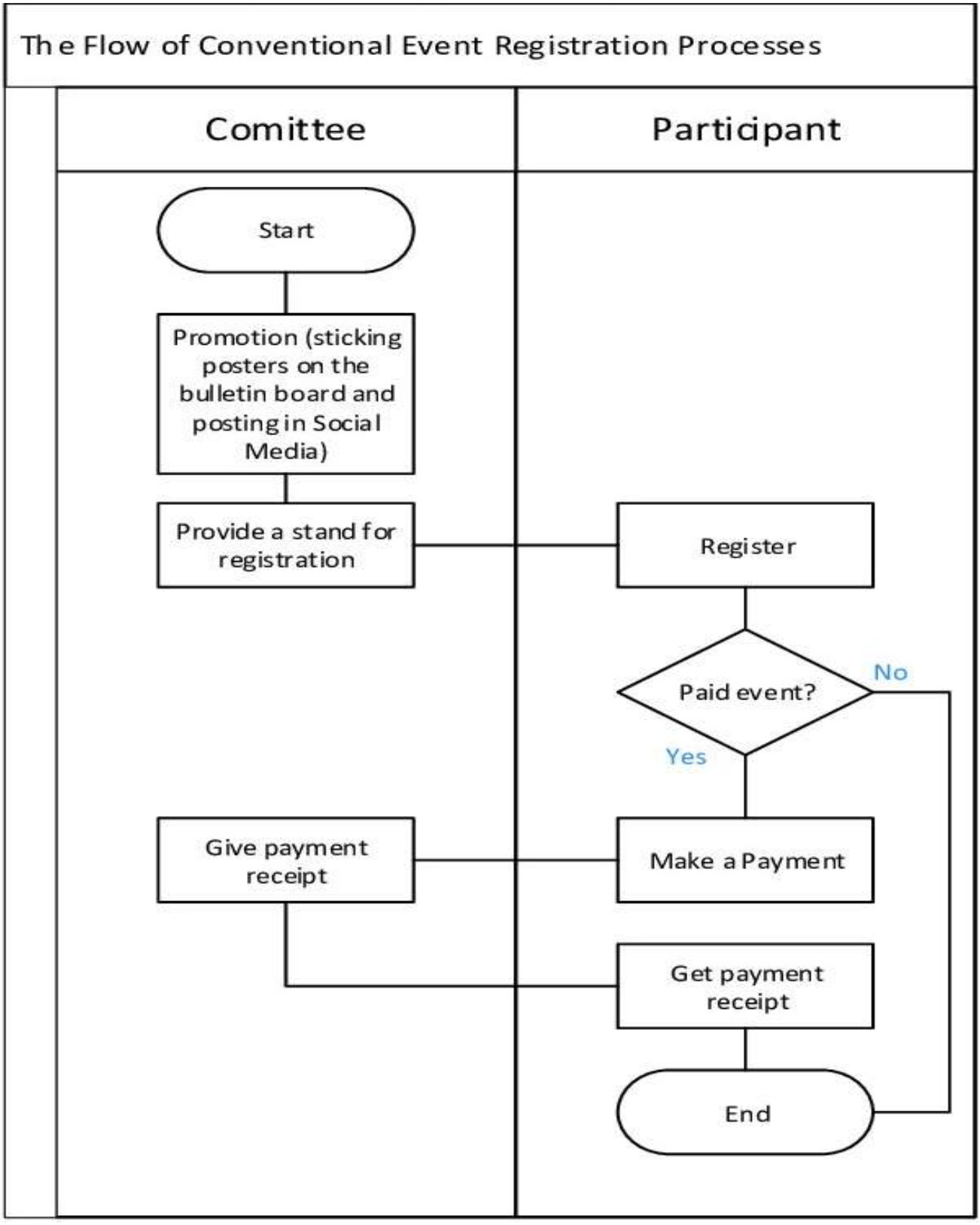


Fig 3.7 Flow diagram of Event registration process

3.5. Code snippets:

Base html for UniHub E-commerce portal:

```
{% extends 'main/base.html' %}

{% load static %}


{% block title %}

   Welcome to Club-Event

{% endblock %}


{% block content %}

   <section class="container">

      <div class="row mt-5 align-items-center">

         <div class="col-md-6">

            <h1 class="text-capitalize">

               Welcome to Club-Event!

            </h1>

            <h3>We are excited to have you join us for an engaging and inspiring experience.</h3>

            <p>

               Here at Club-Event, we know that it can be difficult to find the perfect event for you.

               That's why we've created a platform that allows you to find and register for events that

               are tailored to your interests.

            </p>

         </div>

         <!-- <div class="col">

            <img src="{% static 'images/landing_page_image.jpg' %}" alt="landing page image" class="img-
                fluid">

         </div> -->

      </div>

      <div class="row mt-5 align-items-center search">

         <div class="col-md-6">
```

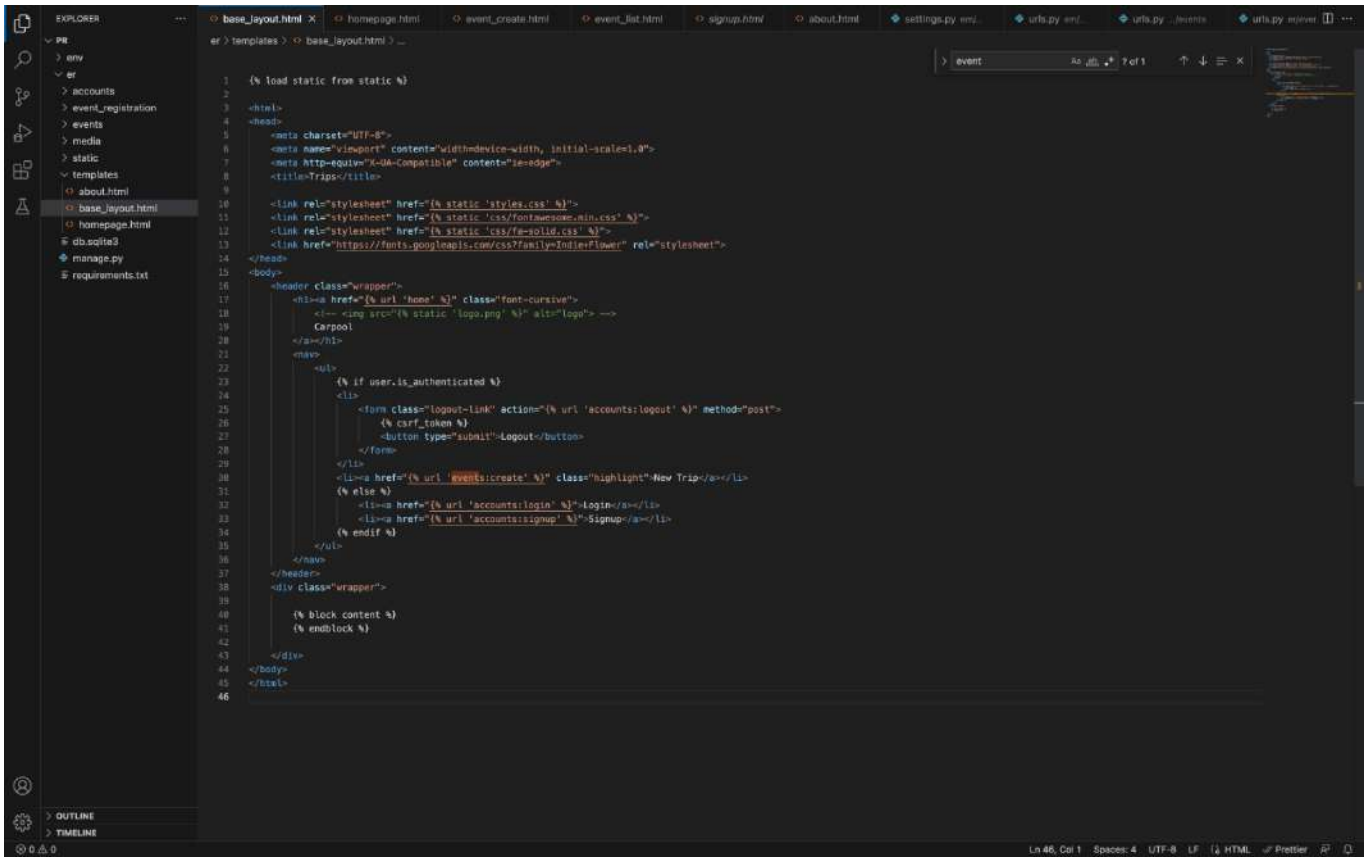
```
        <img src="{% static 'images/search_image.svg' %}" alt="search image" class="img-fluid">
    </div>
    <div class="col-md-6">
        <h1 class="text-capitalize">
            Find Events
        </h1>
        <h3>Search for events by category, location, or title.</h3>
        <p>
            Our search engine allows you to find events that are relevant to you. You can search
            by category, location, or event title.
        </p>
        <a class="btn btn-change" href="{% url 'events' %}">Explore Events</a>
    </div>
</div>
<div class="row mt-5 align-items-center">
    <div class="col-md-6 mb-sm-3">
        <h1 class="text-capitalize">
            Register for Events
        </h1>
        <h3>Register for events that you are interested in.</h3>
        <p>
            Once you find an event that you are interested in, you can register for it. You will
            be able to see the event details and register for the event.
        </p>
        <a class="btn btn-change" href="{% url 'register' %}">Get Started</a>
    </div>
    <!-- <div class="col">
        <img src="{% static 'images/register_image.avif' %}" alt="register image" class="img-fluid rounded-
            3">
    </div> -->
```

```
</div>
<div class="row mt-5 align-items-center search">
    <div class="col-md-6">
        <img src="{% static 'images/attend_image.svg' %}" alt="attend image" class="img-fluid">
    </div>
    <div class="col">
        <h1 class="text-capitalize">
            Attend Events
        </h1>
        <h3>Attend events that you have registered for.</h3>
        <p>
            Once you have registered for an event, you will be able to see the event details and
            attend the event. You will be able to see the event details and attend the event.
        </p>
    </div>
</div>
<div class="row mt-5 align-content-center justify-content-center">
    <div class="col-md-6 my-auto">
        <p>
            Our website is designed to make the event registration process as straightforward and stress-free as
            possible.
            With our easy-to-use interface, you can quickly and easily search for events, view event details, and
            register for events of your choice.
        </p>
    </div>
    <div class="col mt-sm-5">
        <h2 class="text-capitalize">Why Choose Us?</h2>
        <ul class="list-unstyled">
            <li>
                <h3 class="text-capitalize text-purple">Easy to use interface</h3>
```

```
<p>
    Our platform is designed to be user-friendly and intuitive. You can quickly and easily search and register for events in few clicks.
</p>
</li>
<li>
    <h3 class="text-capitalize text-purple">Wide range of events</h3>
    <p>
    We have a wide range of events from small gatherings to large conventions. Whatever you are looking for,
    we are sure to have something for you.
    </p>
</li>
<li>
    <h3 class="text-capitalize text-purple">Free to use</h3>
    <p>
    Our platform is free to use. You can register for as many events as you want without any additional fees.
    </p>
</li>
<li>
    <h3 class="text-capitalize text-purple">Security</h3>
    <p>
    We take your privacy and security seriously. We use the latest security technologies to ensure that your information is safe.
    </p>
</li>
<li>
    <h3 class="text-capitalize text-purple">24/7 customer support</h3>
    <p>
    We are always here to help. If you have any questions or concerns, please feel free to contact us.
```

```
                </p>
              </li>
            </ul>
          </div>
        </div>
        <div class="row mt-5">
          <div class="col-md-8 mt-sm-3 align-items-center">
            <p class="h5">
              Sign up today to start exploring our selection of events and register for the ones that interest you!
            </p>
          </div>
          <div class="col-md-4 mt-sm-5 text-center">
            <a href="{% url 'register' %}" class="btn btn-change w-50">Get Started</a>
          </div>
        </div>
    </section>
{% endblock content%}
```

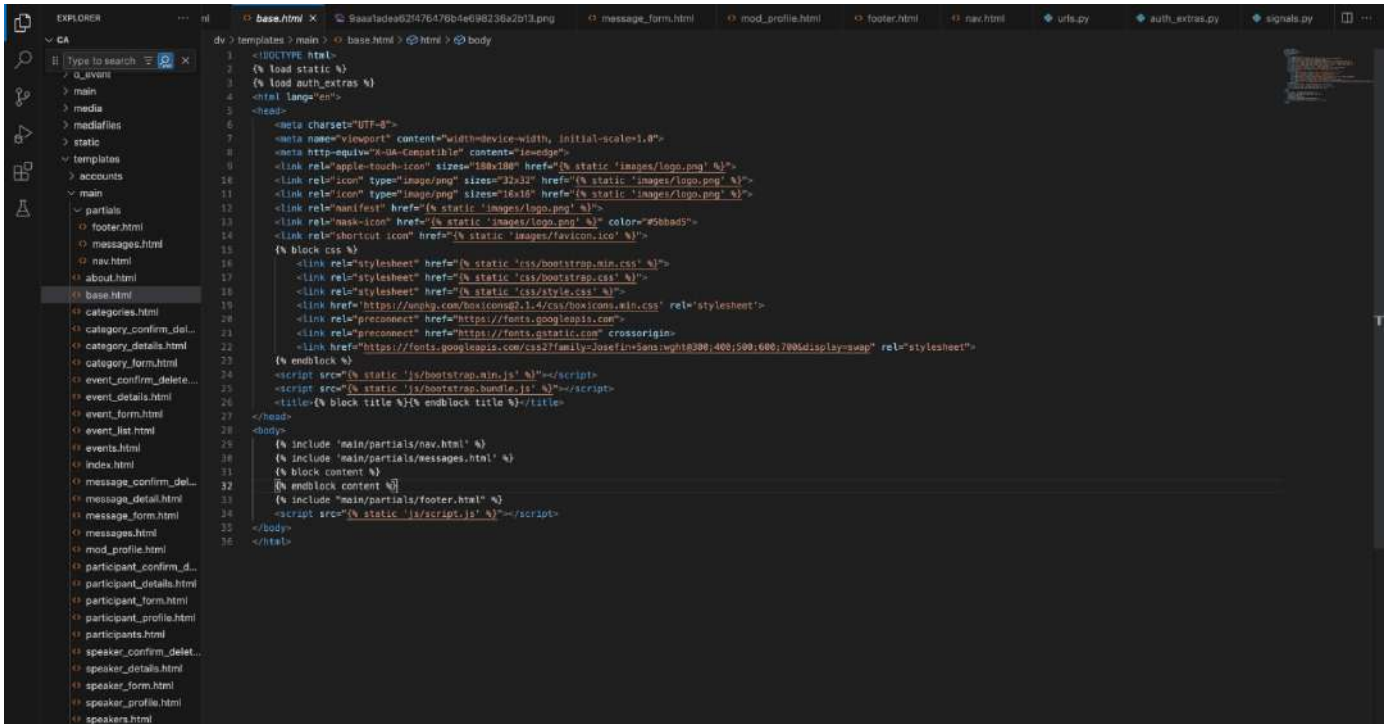
Base html for Car pool portal:



```
{% load static from static %}

<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Trips</title>

    <link rel="stylesheet" href="{% static 'styles.css' %}">
    <link rel="stylesheet" href="{% static 'css/fontawesome.min.css' %}">
    <link rel="stylesheet" href="{% static 'css/fa-solid.css' %}">
    <link href="https://fonts.googleapis.com/css?family=Indie+Flower" rel="stylesheet">
</head>
<body>
    <header class="wrapper">
        <h1><a href="{% url 'home' %}" class="font-cursive">
            <!-- <img src="{% static 'logo.png' %}" alt="logo"> -->
            Carpool
        </a></h1>
        <nav>
            <ul>
                {% if user.is_authenticated %}
                <li>
                    <form class="logout-link" action="{% url 'accounts:logout' %}" method="post">
                        {% csrf_token %}
                        <button type="submit">Logout</button>
                    </form>
                </li>
                <li><a href="{% url 'events:create' %}" class="highlight">New Trip</a></li>
                {% else %}
                    <li><a href="{% url 'accounts:login' %}">Login</a></li>
                    <li><a href="{% url 'accounts:signup' %}">Signup</a></li>
                {% endif %}
            </ul>
        </nav>
    </header>
    <div class="wrapper">

        {% block content %}
        {% endblock %}

    </div>
</body>
</html>
```

Fig 2.1 Code snippet for car pool portal

Base HTML for Event management portal:



```
<!DOCTYPE html>
{% load static %}
{% load auth_extras %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="apple-touch-icon" sizes="180x180" href="{% static 'images/logo.png' %}">
    <link rel="icon" type="image/png" sizes="32x32" href="{% static 'images/logo.png' %}">
    <link rel="icon" type="image/png" sizes="16x16" href="{% static 'images/logo.png' %}">
    <link rel="manifest" href="{% static 'images/logo.png' %}">
    <link rel="mask-icon" href="{% static 'images/logo.png' %}" color="#5bbad5">
    <link rel="shortcut icon" href="{% static 'images/favicon.ico' %}">
    {% block css %}
        <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
        <link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">
        <link rel="stylesheet" href="{% static 'css/style.css' %}">
        <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
        <link rel="preconnect" href="https://fonts.googleapis.com">
        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
        <link href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@300;400;500;600;700&display=swap" rel="stylesheet">
    {% endblock %}
    <script src="{% static 'js/bootstrap.min.js' %}"></script>
    <script src="{% static 'js/bootstrap.bundle.js' %}"></script>
    <title>{% block title %}{% endblock title %}</title>
</head>
<body>
    {% include 'main/partials/nav.html' %}
    {% include 'main/partials/messages.html' %}
    {% block content %}
    {% endblock content %}
    {% include "main/partials/footer.html" %}
    <script src="{% static 'js/script.js' %}"></script>
</body>
</html>
```

Fig 2.2 Code snippet for Event management portal-I

Index and html of event management:

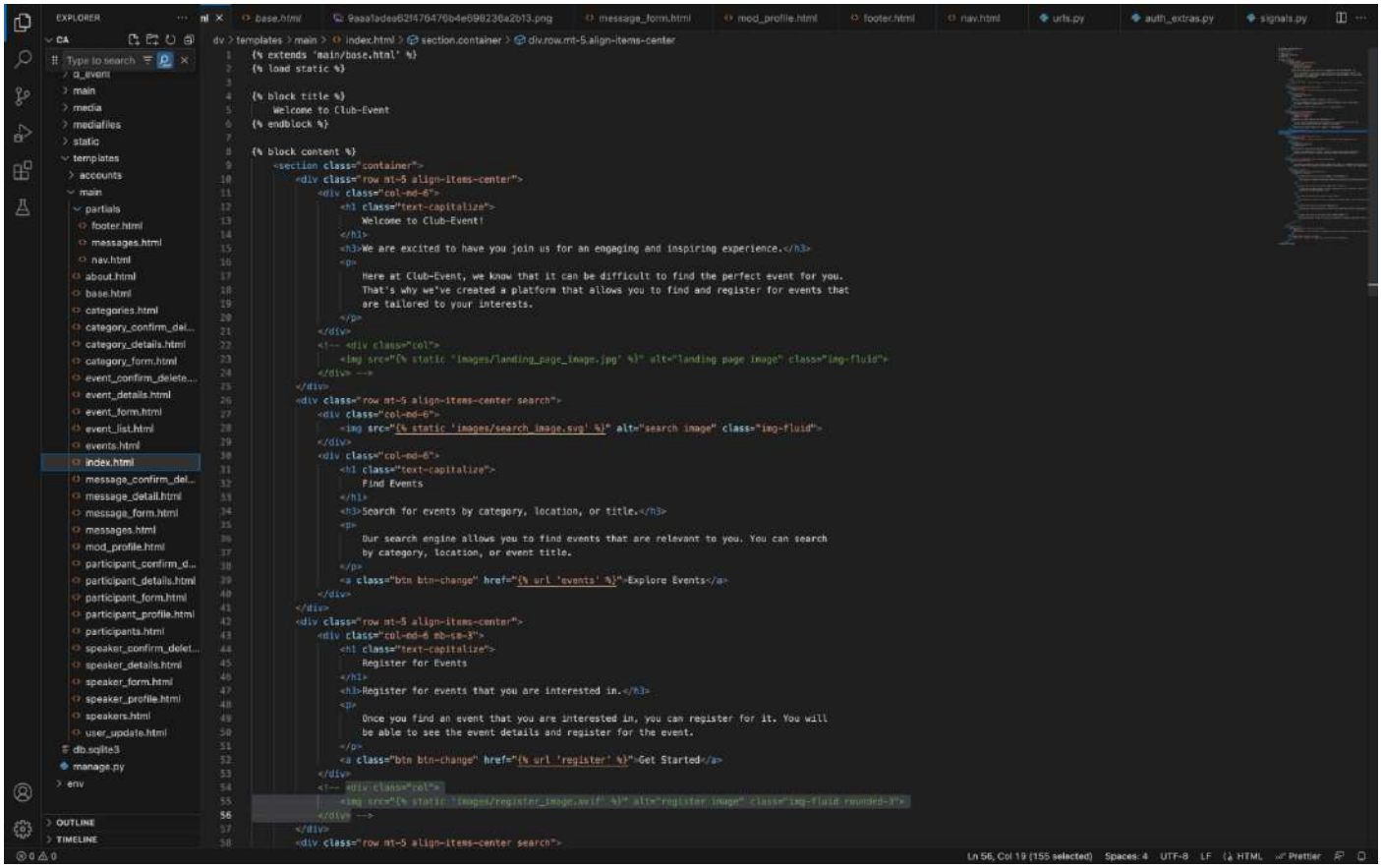


Fig 2.3 Code snippet for Event management portal-II

4. Results and discussions

An online marketplace with clear shopping interface, detailed product categorization, robust search and chatbots, where users can make a profile and sell or resell products and other users can buy that.

A cab-hailing portal, where users can get contacts of available cab owners, cab drivers, cabs, etc. And another option of cab pooling where users can share a ride with others by putting all their travel details such as date, time, cab type, available capacity fare and also their contact details.

An event registration website where event managers can list events with all details and other users can simply register for events on that portal itself.

Final Results:

Our Vendor partner: Raj Shopping All

Fig 3.1 UniHub vendor partner

UniHub E-commerce portal: Functionalities and Features:

For seller: User registration, login and logout as seller(vendor, shopkeeper or senior);

Product categorization and listing with price and quantity etc;

Order management;

For customers: User registration, login and logout as customer;

Product viewing;

Adding to cart or order;

Managing the cart;

Screenshots:

Order management by seller:

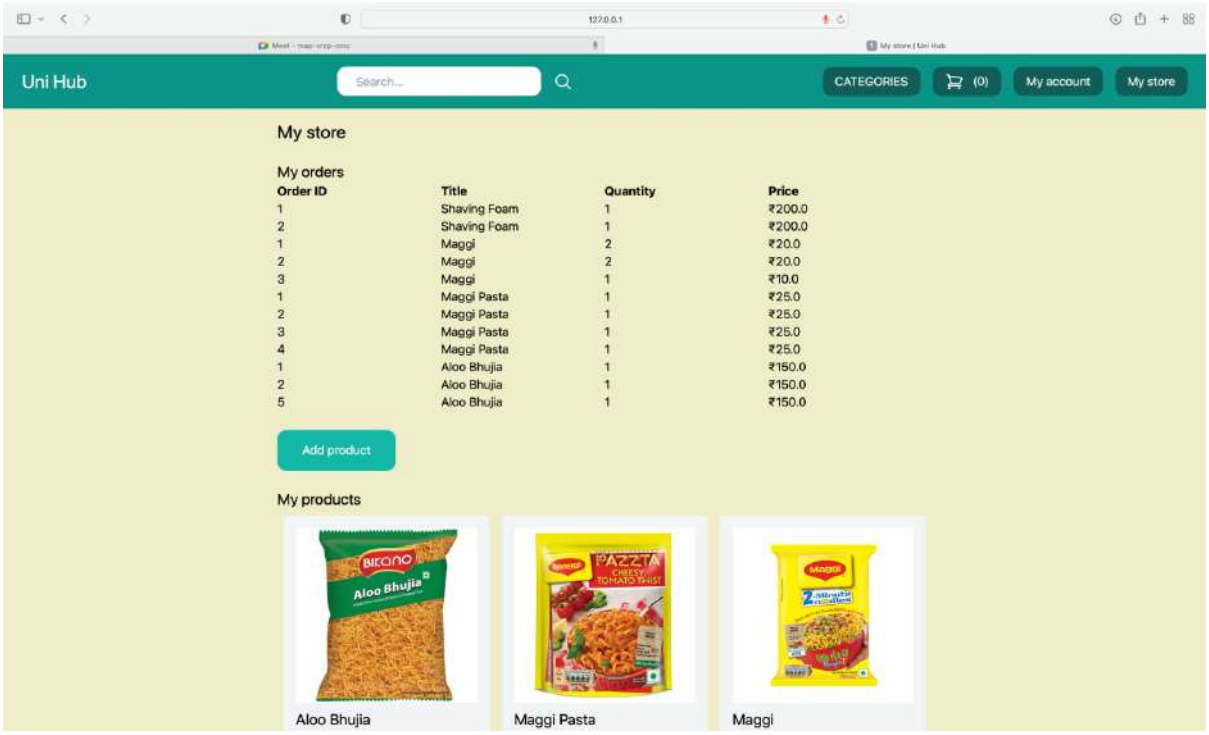


Fig 3.2 Order management by seller

Products listing on website:

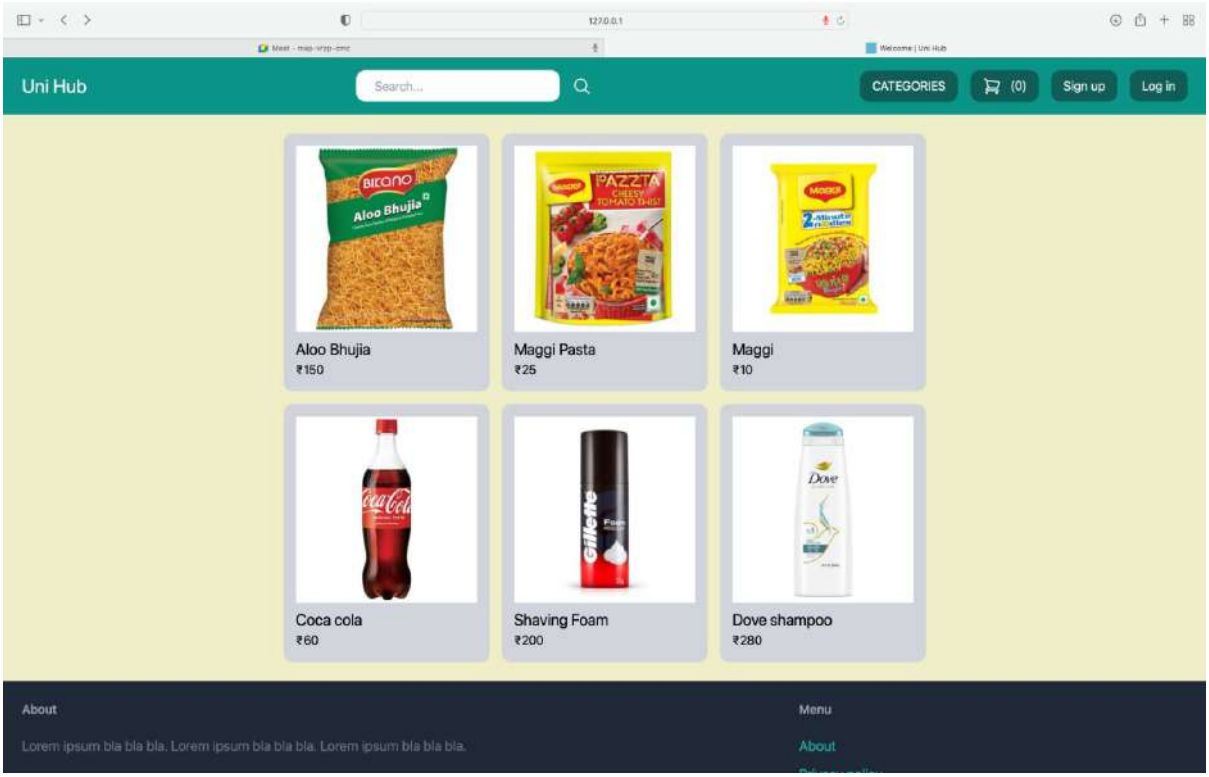


Fig 3.3 Products listing on website

Product Categorization:

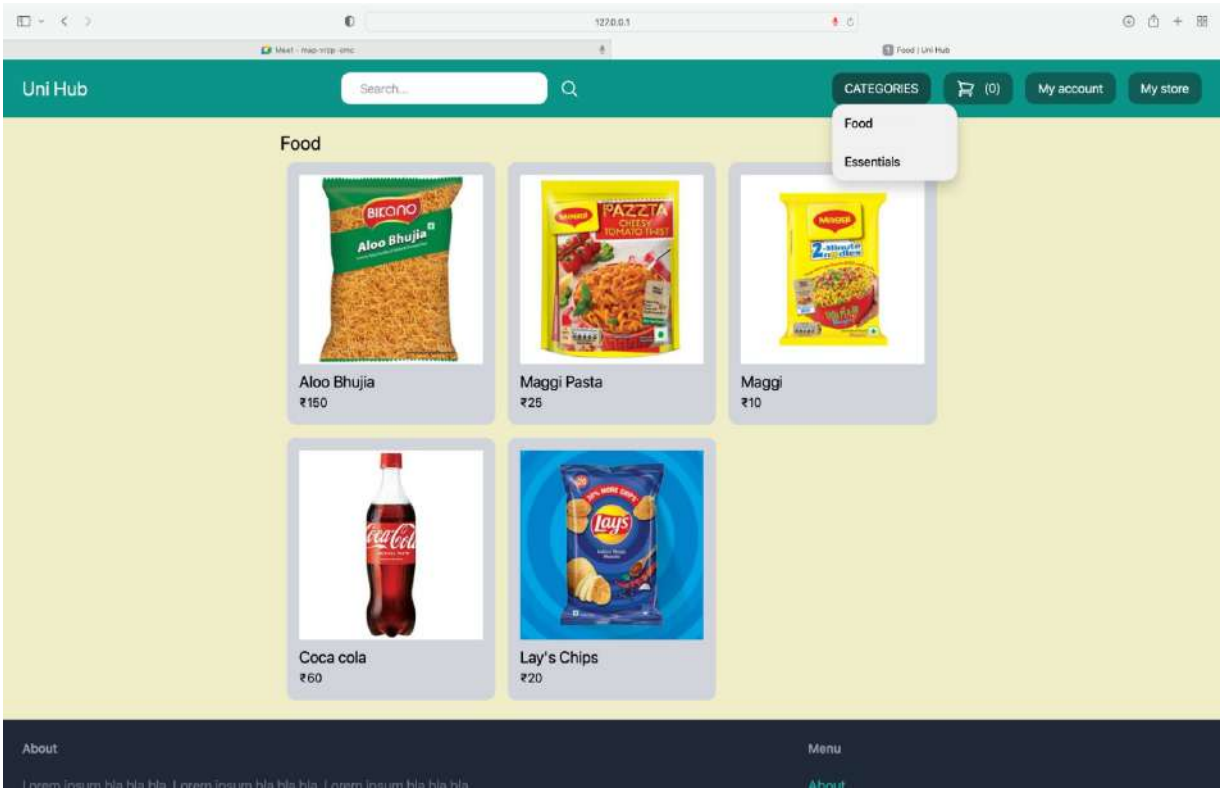


Fig 3.4 Product Categorization

Cart view and management:

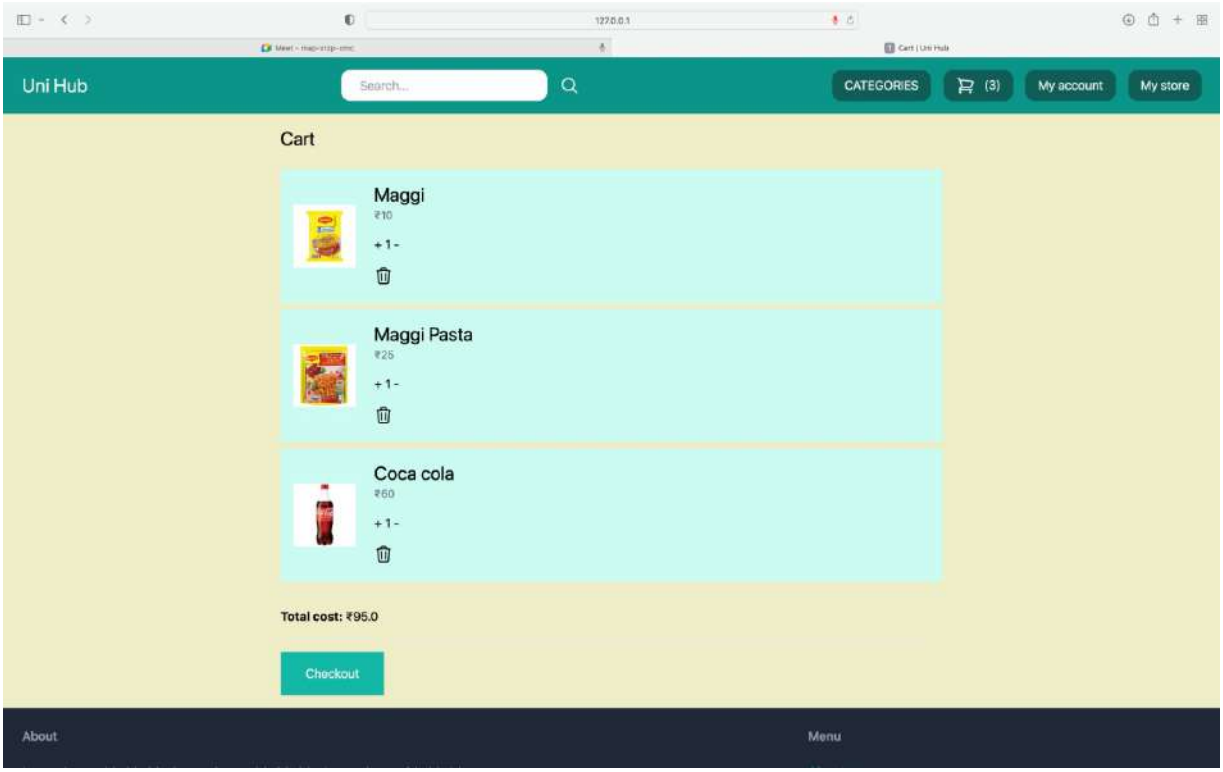


Fig 3.5 Cart view and management

Car Pooling portal: Functionalities and Features:

User Registration, login and logout;

New trip creation;

View trip;

User addition to trip.

Screenshots:

Creating new trip:

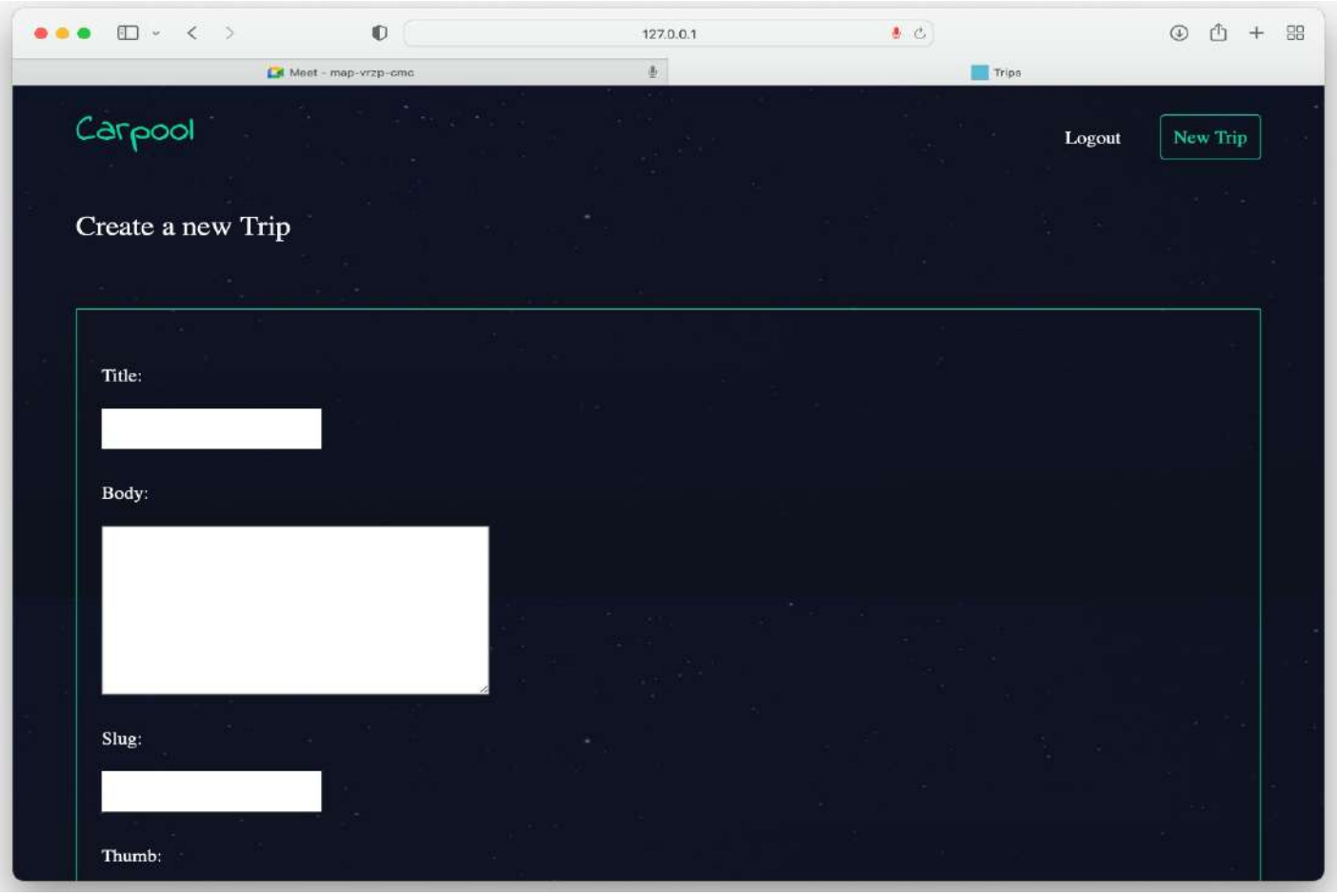


Fig 3.6 New trip creation process

View trip details:

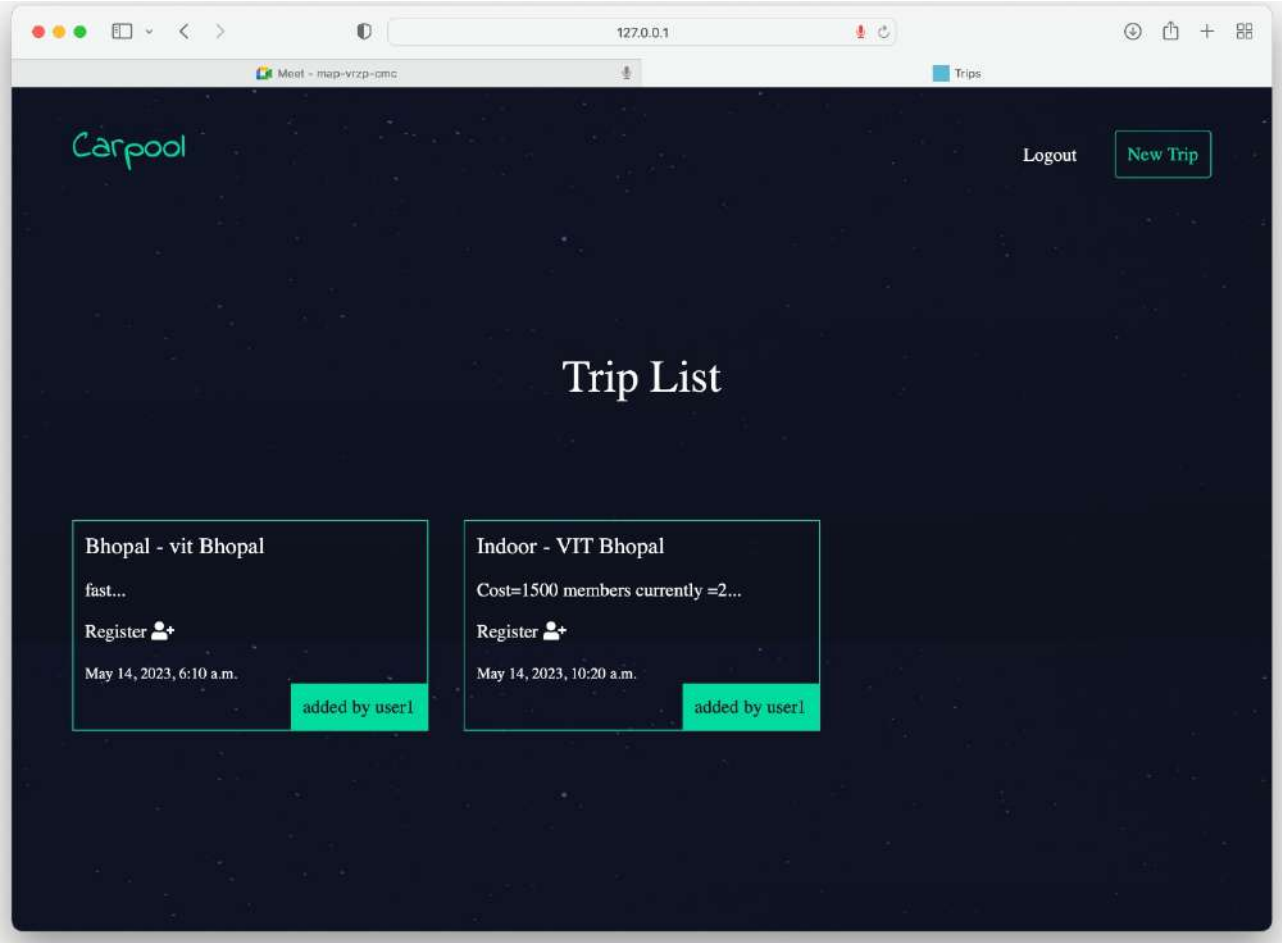


Fig 3.7 Trip listing and trip details

Event Management portal: Functionalities and Features:

User registration, login and logout;

Registering new events and event details by event managers;

List of events and event details to view and register for other users.

Screenshots:

Home page:

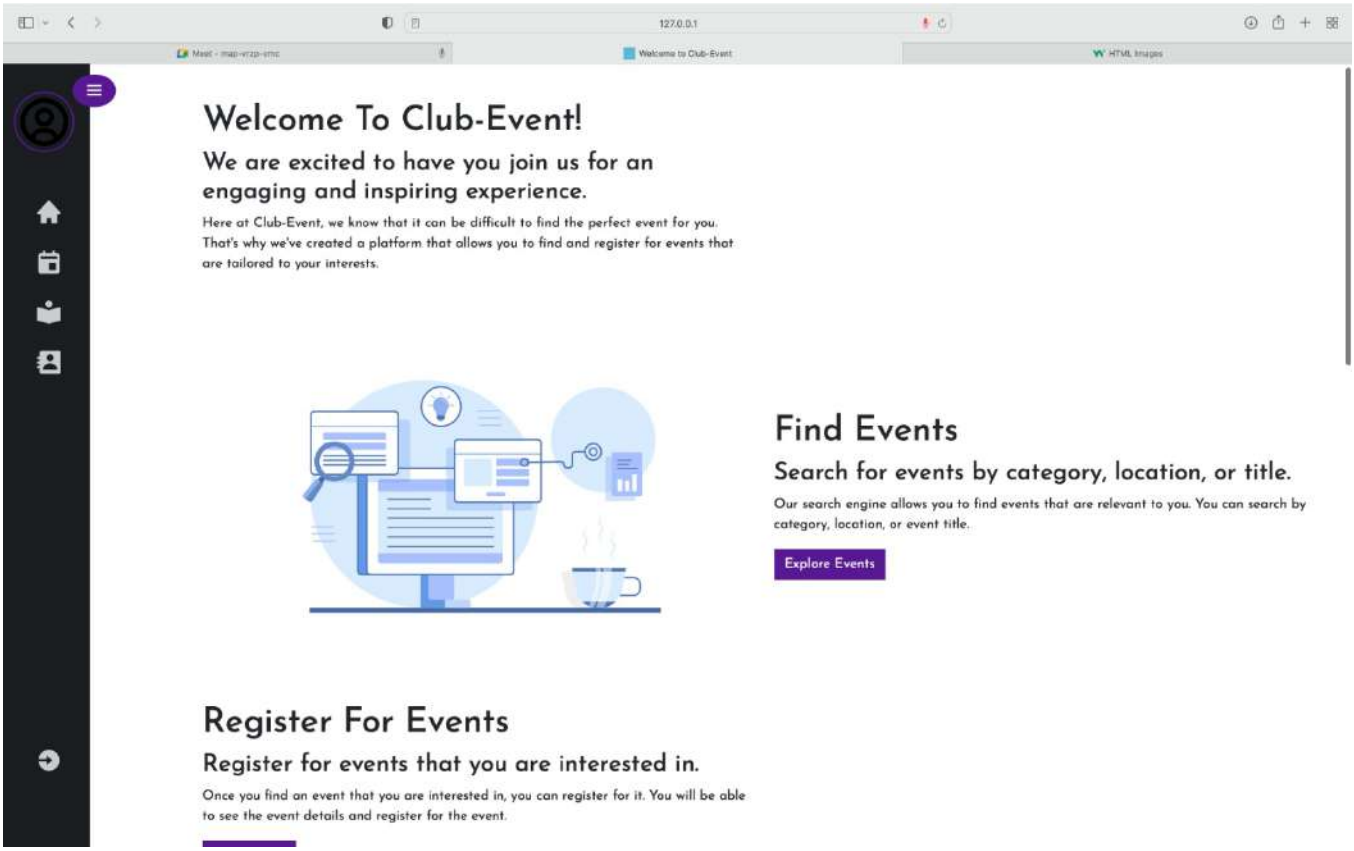


Fig 3.8 Home page of event registration site

Event listing:

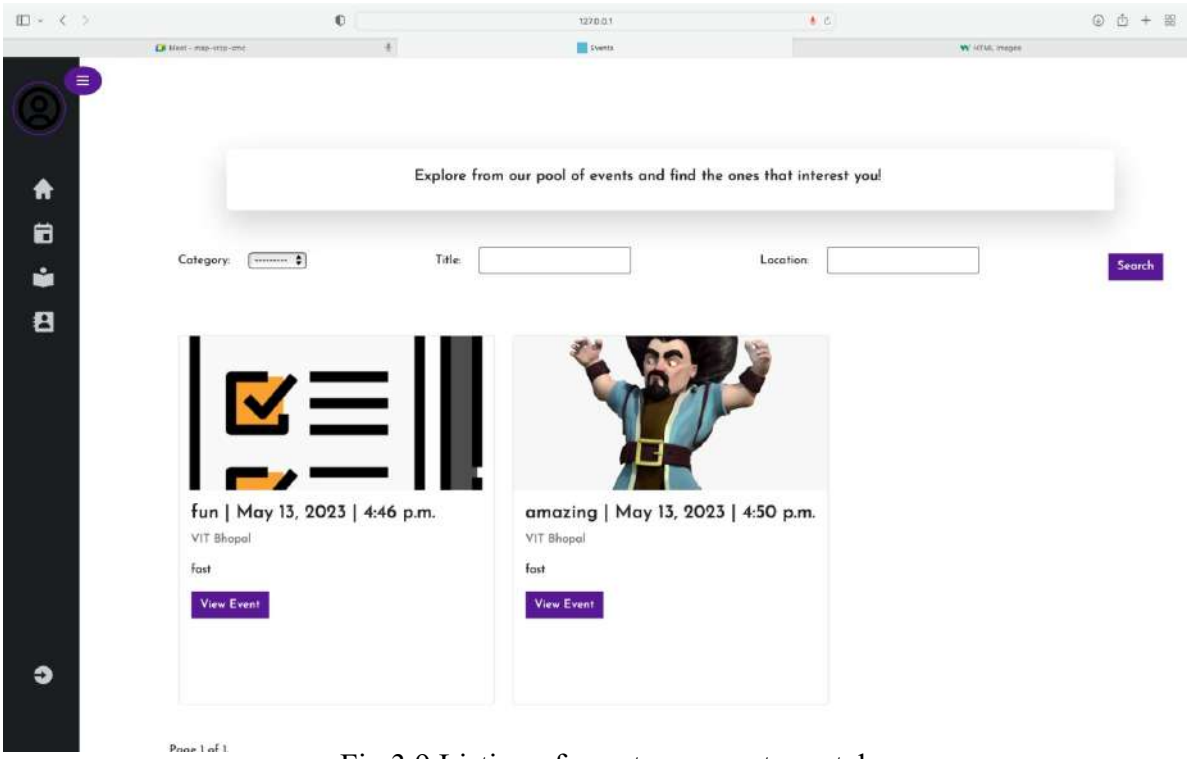


Fig 3.9 Listing of events on events portal

About us and contact information:

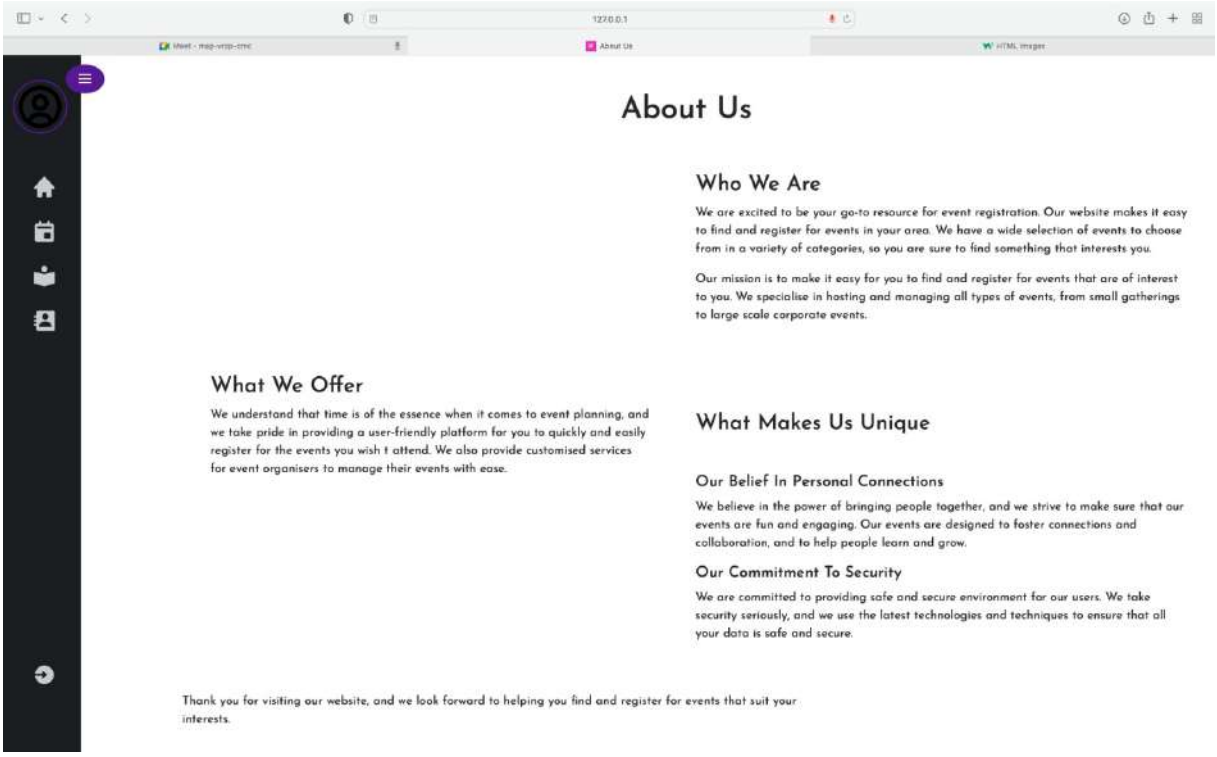


Fig 3.10 Our information page

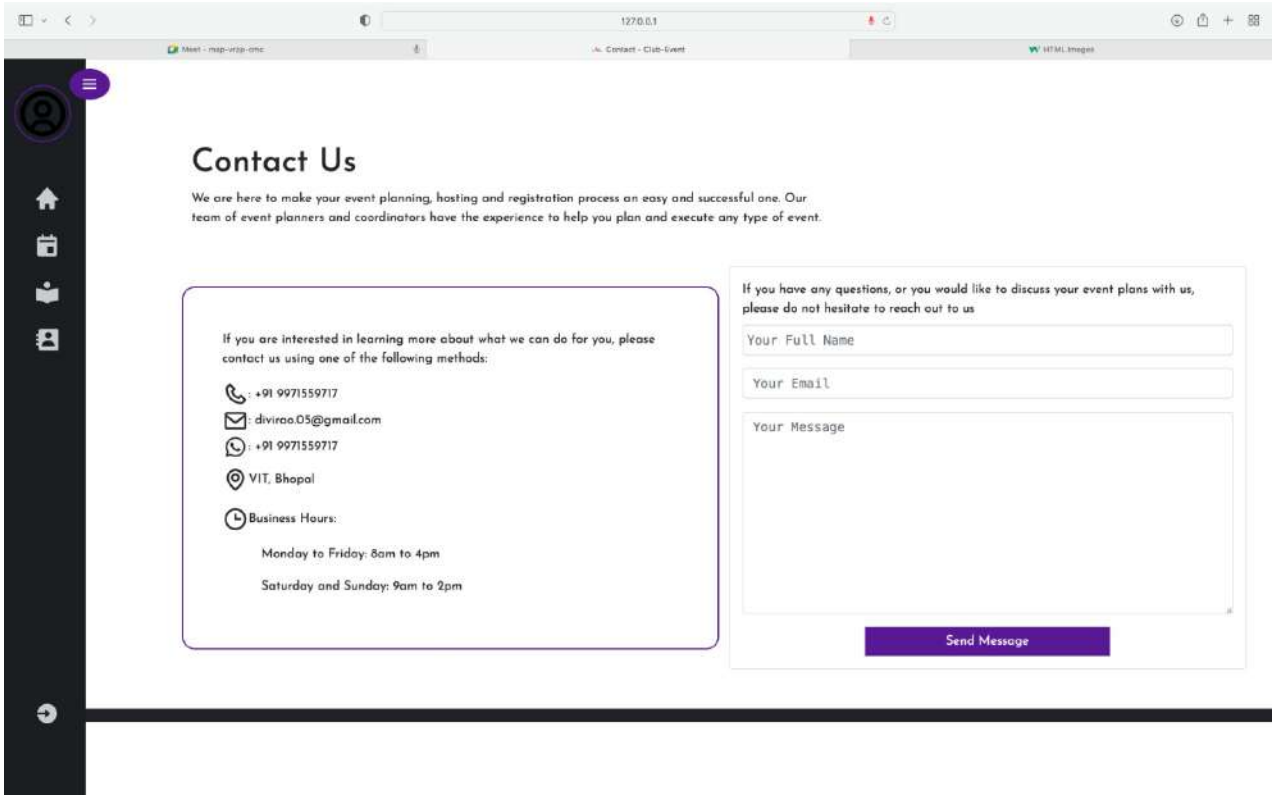


Fig 3.11 Our contact information page

5. Individual Contribution by members

1. Divyansh Yadav (20BAI10095)

Team Lead, responsible for frontend and backend development of event registration/management section, frontend and backend implementation/management of the whole project.

As the team lead, I was responsible for managing and compiling the website code that would provide a seamless experience to the users. By using Django, HTML and CSS with python being the main programming language for the project, I created a comprehensive backend and frontend for event management part of our UNI-HUB website.

1. Frontend and Backend for Event Management webpages:
   Used HTML and CSS to make attractive webpages where user and view different events and take part in the events by registering. They can view their registered events and admin/event organizer can view all user registered to his event. Created interactive event creation page for admin. There is a separate webpage for speakers where they can view their upcoming events.
2. Frontend and Backend Management of whole project:
   For this I linked backends and frontend of every functionality so that correct information gets processed for particular directory and user can use every feature though a single website and have a seamless experience.

The User Interface is kept as easy and unique as possible. The admin being the event organizers can create a new event with respective details such as event name, date, time, location, description, photos, etc. The admin can make the event description as attractive as possible. The normal users being the students can login and register for any of the club events. The admin will be able to check the participant's details. Currently the system works seamlessly for free and on spot payment events as we have not integrated any payment gateway. We can enable this feature to make the payment process of paid events much simpler.

Apart from this I integrated all different functionalities for the backend and frontend of every element in the project. Like integrating every model/ database (SQLight) of the different directories and elements. This made the whole user experience very seamless, as now they have one single website for all their university needs. They can order essentials, book cabs, carpool with friends and register for club events from a single place.

2. Ritwik Agarwal (20BAI10039)

Data collection through form collection, chatting and convincing vendors. Helped in debugging

Introduction: This section introduces the purpose of the survey, the scope of the survey, and the methodology used to collect the data.

Executive Summary: This section provides a brief overview of the key findings and insights from the survey.

Survey Results: This section presents the survey data and analyzes the results. This may include tables, charts, and graphs to illustrate the data and findings.

Analysis and Interpretation: This section interprets the data and provides insights and conclusions based on the findings.

Recommendations: This section suggests recommendations based on the survey results and analysis. These may be recommendations for further research, policy changes, or other actions.

Conclusion: This section summarizes the survey findings and highlights the key takeaways.

Appendices: This section includes any additional information such as survey questionnaires, detailed data tables, and other supporting materials.

link-
https://docs.google.com/forms/d/e/1FAIpQLSdLsVGvqKlodkHrxVC1disPn38QCZ6bj4Rukz2M7yPrTTqi7g/viewform?usp=sf_link&usp=embed_facebook

A survey was conducted for our project through google forms which had around 10 questions concerning the user's need and opinion. We got over 1500+ responses over a period of 3 months. Most of the user's had a positive review towards our project which gave us a lot of motivation regarding the development of our project which we are continuing to do.

After getting a positive review from the users. It was time to contact the local vendors, to make them aware of our project. Its advantages and disadvantages and get to know whether they are onboard or not or whether they like the idea or not. We started with some of the vendors which are outside our college campus. Some of the methods used were: -

•Direct Mail or Flyers: Consider sending out direct mail or flyers to local vendors in the area. This is a cost-effective way to introduce your e-commerce site and offer an incentive for them to sign up as a vendor.

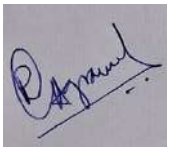
•Social Media: Utilize social media platforms to promote your e-commerce site and target local vendors. You can create posts, run ads, or join local groups and communities to get your message across.

•Email Marketing: Send targeted email campaigns to local vendors and offer incentives such as free sign-up or lower commission rates for early adopters.

•Personal Outreach: Consider reaching out to vendors directly through phone calls or in-person meetings. This can help build a personal relationship and increase the likelihood of them signing up as a vendor.

•Partnerships and Referrals: Consider partnering with local business associations or influencers to promote your e-commerce site. Referrals from trusted sources can help increase the credibility and visibility of your project among local vendors.

By using the above methods, mostly by contacting by personal outreach, we finally managed to get 12 local vendors onboard with us for the project which were comfortable with our prices and agreed to sell their products to the users through our website.



3. Yuvraj Chopra (20BAI10301)

Front end for ecommerce, vendors, customers, cab pooling and driver information webpage.

In the process of developing our group epics project, I was been assigned to design the front end of our project and to create a separate webpage for taxi services along with it I also contributed to debugging the code, all of my contributions are explained below.

a) Developing front end

The front end of a website is responsible for enhancing user and site interaction while developing a website creating a user-friendly frontend is always prioritized in order to achieve our goal, we may sometimes use graphics, headings, buttons, etc. Thus, the main objective of creating a front end is to make the user smooth experience.

Markup language (HTML), Cascading language (CSS), Scripting language (JavaScript)

Besides all these above-mentioned skills we should also be careful in developing a responsive and visually attractive front end for these there are a lot of front-end frameworks present like Bootstrap, react JS, and angular JS.

Thus, while creating frontend of our website, we have especially concentrated on the above-mentioned things

b) Designing a static web page for taxi services

It is a static web page for our epics group project which is having contacts of various taxi drivers stored in cards by clicking on the call button we can directly make a phone call to them thus which makes the process of booking a taxi much easier and quicker, for making this webpage I utilized my skills of HTML, CSS, and JavaScript along with my creativity which helped me a lot in creating the webpage.

c) Debugging

Along with contributing in frontend and creating a web page for taxi booking. I also helped my friend in debugging process as I am having a fundamental knowledge of Django framework and python along with my past working experience in Django flask and python which helped me in fixing some small bugs.

Conclusion: While completing these tasks, I actually gained a lot of experience in developing an industry-grade project, maintaining team coordination, and at last fixing small bugs

4. Amisha Goyal (20BCE10496):

Responsible for front end of ecommerce, vendors, customers and carpooling sections.

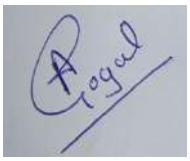
As the team member responsible for 'Home Page' designing, my task was to create a visually attractive and easy-to-use home page for the UniHub app. I was responsible for designing the layout, selecting appropriate colors, and choosing relevant icons or images that align with the brand identity and convey information to users.

As the team member responsible for 'Custom button' designing, my responsibility was to create custom buttons that match the visual design of the app. These buttons should be easy to use and visually appealing to users.

When implementing the 'date_controller' in the 'home-page' file, I was responsible for creating the necessary code to enable users which product they have chosen and added in cart. This would require integrating the app with a date and time picker library and implementing the necessary logic to save the reminder date and time to the database.

As the team member responsible for debugging and coordinating with the working of the 'add_task_bar' file, I identified and fixed any issues or bugs in the 'add_task_bar' file and ensured that it functions correctly. This involved 'add_task_bar' file and debugging any issues that arise.

In summary, as the team member working on these responsibilities, I was responsible for designing user interfaces, creating custom buttons, implementing date and time picker functionality, and debugging code. I worked collaboratively with my team members to ensure a seamless integration of features and resolve any issues that could have arisen during the development process.



5. Varun Mishra (20BCE10844)

Backend part of vendors and customers.

As a member of the project team, I was tasked with creating two distinct backends for the customer and vendor portals. It was a fascinating challenge that required a combination of several technologies to achieve the desired results.

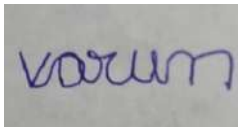
The first step I took was to set up the database using SQLite, which is a lightweight and user-friendly database management system. It allowed me to create tables that captured crucial details about vendors and customers, such as their names, addresses, contact information, and orders. The database was vital in providing a foundation for the entire project, as it stored all the data required to manage the customer and vendor portals.

For the vendor portal backend, I decided to use the Django framework, which is a powerful and flexible tool that enables developers to create views, models, and templates with ease. With Django, I was able to manage vendor information such as new vendors, updating their details, and viewing their orders through the Django admin interface. This made it easier for us to keep track of all the vendors in the system and monitor their activities.

Similarly, I utilized Django to develop the customer portal backend. With Django, I was able to create views that allowed customers to view products, add items to their cart, and place orders. I also integrated the Pillow Python imaging library to handle product image uploads, making it easier for customers to view the products they wanted to purchase.

To ensure the system's security and stability, I implemented several measures such as password encryption, input validation, and data sanitization. I also set up a testing environment to test the system's functionality and identify any bugs or issues.

Overall, I am incredibly proud of the contribution I made to the project. By creating separate backends for the customer and vendor portals, we ensured that the system was well-organized and easy to manage. The integration of SQLite, Django, Python, and Pillow made it possible to develop a powerful and adaptable system that met all the project's requirements.



5. Kirti (20BCE10526)

Backend part of carpooling.

As the primary backend developer for our group project, I had the responsibility of designing and creating the backbone of our carpooling portal for students. I wanted to make sure that the platform was not only reliable and efficient, but also easy to use and secure for our users.

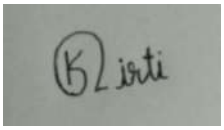
To achieve this, I used several technologies such as SQLite, Django, Python, and Pillow. These technologies allowed me to develop a solid foundation for our platform that could handle large amounts of data and user requests. I used SQLite to manage our data and Django as the framework for developing the backend. Python was my primary language of choice for backend development, as it provided me with a lot of tools and libraries to use. Lastly, I used Pillow to process and manage images for our platform.

One of the main features I created for our platform was the ride-sharing function. With this feature, students could easily find and share rides with fellow students who were going in the same direction. To develop this feature, I used a complex algorithm that took into account different factors such as the destination, the time of the ride, and the number of seats available in the car. This algorithm made it possible for students to find rides that were convenient for them while minimizing the number of empty seats in the cars.

Another feature I developed was the directory of cab owner/driver contracts. This allowed students to have a variety of options when looking for a ride. They could search for and contact different cab owners/drivers in their area. I made sure to keep this directory updated regularly to ensure that students had access to the latest information on available cabs in their area.

Throughout the development process, I paid special attention to the security of our platform. I ensured that user information was encrypted and protected from unauthorized access. I also conducted extensive testing to make sure that the platform was free of bugs and could handle a high volume of traffic.

Overall, our carpooling portal for students will have a positive impact on the daily lives of students in our community. The platform is efficient, secure, and user-friendly, and I'm grateful to have worked with such a talented and motivated team.



7. Trilok Dhakad (20BCY10126)

Security testing

Software Testing in Agile Development:

In our application development process, we adopted Agile software development techniques, where requirements, programming, and testing are carried out concurrently. To ensure the proper development and testing of our application, we implemented the following Testing Cycle:

Meeting Requirements and Use Case Verification:

We verified that our application met the requirements that guided its design and development, aligning with the desired outcomes of the stakeholders. We cross-verified all the use cases identified in the SRS (Software Requirements Specification) documents to ensure their correct implementation. Additionally, we asked friends and family to use the application, evaluating its usability without external assistance and ensuring that all necessary features for a carpool management system were fulfilled. Furthermore, we diligently tested and documented the entire code base, making it accessible for future developers.

Responding to Inputs:

Static Testing: Before committing or shipping each section of our code, we conducted verification activities such as reviews, walkthroughs, or inspections. Another team member reviewed the code, providing appropriate comments for any necessary changes. We utilized GitHub as our version control system, which facilitated efficient handling of these activities.

Dynamic Testing: We ensured that proper logging statements were printed on the terminal or console, making it easier to debug any runtime errors that occurred during the execution of the application.

Unit Testing: For each function (views.py) developed within our Django environment, we wrote corresponding unit tests using Python's unit test module. By running these tests, the test utility automatically built a test suite comprising all test cases (subclasses of unittest.TestCase) from files starting with "test". We aimed to achieve a code coverage of approximately 90% to minimize the risk of failures in our application.

Performing Functions within Acceptable Timeframes:

Functional Testing: We conducted functional testing to verify specific actions or functions of the code. These tests answered questions such as "can the user do this?" or "does this particular feature work?" Before integrating specific services, such as Google Maps API or payment calculation services, with our codebase, we ensured they functioned correctly independently.

Integration Testing: To expose any defects in the interfaces and interaction between integrated components (modules), we performed integration testing. Before and during the code push for different modules (APIs and services like payment), we thoroughly checked their interactions to avoid breaking any existing functionality and to seamlessly integrate them with our system.

Ensuring Usability:

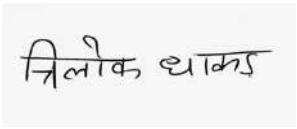
Software Performance Testing: We executed performance testing to determine how our system or sub-system performed in terms of responsiveness and stability under specific workloads. This testing also allowed us to investigate, measure, validate, or verify other quality attributes of the system, such as scalability, reliability, and resource usage. We conducted load testing to verify the application's ability to handle multiple user instances (both riders and drivers) and to ensure its performance remained efficient even with increased user load. Stress testing was utilized to validate the proper functioning of important modules under unexpected amounts of traffic, while stability testing ensured continuous functionality within an acceptable timeframe.

Installation and Environment Testing:

Installation Testing: We followed the software installation instructions mentioned in the project's readme file on multiple operating systems, ensuring a smooth installation process without encountering any issues.

Compatibility Testing: To prevent software failure due to incompatibility, we performed compatibility testing. We ran the application on multiple operating systems (Ubuntu, Windows, and macOS) and tested its compatibility with different web browsers (Chrome, Firefox, and Safari). This process verified that all modules functioned properly across various environments.

By following this comprehensive software testing approach, we aimed to deliver a high-quality, reliable, and usable application.



8. Telugunti Akhil (20MIP10045)

Responsible for front end of ecommerce, vendors, customers and carpooling.

As the team member responsible for 'Onboarding Screens' designing, my task was to create visually appealing and user-friendly onboarding screens for the UniHub app. I was responsible for making the content appropriate, fetching the relevant information and convey information to users in a clear and concise manner.

One of my primary responsibilities as a front-end developer was to design the website's user interface. I worked closely with the project's design team to ensure that the website had a consistent look and feel, and that the layout was optimized for ease of use. I also focused on creating a responsive design that would provide a seamless experience for users on a range of devices, from desktops to mobile phones.

Looking ahead, there are several potential improvements that could be made to the website. These might include adding new features, improving the user interface, or optimizing the

website for better search engine optimization (SEO) performance. By continuing to work closely with the team and incorporating user feedback. I worked closely with other members of the team, including designers, back-end developers, and project managers. I communicated regularly with them to ensure that the website met the project requirements and was delivered on time and within budget.

In conclusion, as a front-end developer on the UniHub project, I played a critical role in designing and developing the website's user interface. Through my efforts, I helped to create a visually appealing, user-friendly, and responsive website that provides customers with a convenient and affordable way to shop for groceries online. I look forward to continuing to work on this project and making further improvements to enhance the user experience.

6. **CONCLUSION**

As engineers, we are expected to solve various small-large problems and are taught the same. So, we decided to start with solving our daily life problems, to make a platform with college localities to provide students products and travel facilities in the most comfortable and hazard free way possible that too at a reasonable price. And with this, local vendors and shopkeepers will be able to increase their reach and profit as well.

This e-commerce website will feature the online trading facility of various products under a single web space. The proposed web application will allow multiple shopping vendors, small shopkeepers, cab owners/drivers, and students to list their services and products, creating a transparent marketplace for student needs that benefits both the buyers and sellers, offering fair prices and efficient transactions. The product management in the system will be done in the form of categories.

By providing a centralized platform, we aim to simplify the lives of college students and create an efficient, transparent marketplace for vendors and students. Our platform will offer a range of functionalities, including daily necessities purchasing, budget and environment friendly, safe and convenient transportation booking, club events notification, and registration, all available in one place.

We are able to create a responsive platform for e-commerce web application with product categorization and shopping cart options, our own cab pooling application with a link on the homepage of the main website and all-in-one platform for college event registration.

Since the e-commerce industry has experienced exponential expansion, a new choice will easily join this competition of commercial websites. The e-commerce website will offer the ability to shop online for a variety of products from a single web location. Multiple shopping sellers will be able to sell their products online thanks to the suggested web application. The system's product management will take the shape of categories. Information security is the system's primary requirement, and it will be handled as such.

Reference:

1. https://companies.naukri.com/olx-careers/about-us/
2. https://www.techtarget.com/whatis/definition/Facebook-Marketplace
3. https://pypi.org/project/Pillow/
4. https://www.javatpoint.com/create-rest-api-using-django-rest-framework
5. https://www.w3schools.com/django/django_intro.php
6. https://sqlitebrowser.org/