**Project Name: Eco-Friendly Products**

**Project Member:**

**Tanisha Patil 210943120109**

**Aparna Sawant 210943120015**

**Aparna Sonwane 210943120016**

**Trupti Patil 210943120112**

**Abstract:**

The web based “Eco-Friendly Products” project is an attempt to stimulate the basic concepts of Grocery Store. The system enables the customer to do the things such as search for Eco-Friendly Products category wise, choose products based on description and add that product into cart.

The system provides you details about product. If user wants to buy product, he must have registered account. The system shows the product that are available or that are in stock, the system displays price, quantity, availability of product to user as well as employees of store Here we provided products by category wise that allows customer to choose a particular product easily.

The product which are not in stock, user cannot add that products into cart. If the products are available then the system allows the user to add product into cart. To place order system, ask user to select the address and payment mode.

If address is not provided the user can’t place order, User have to specify the address before placing order. After selecting address and payment mode user can place order and the same updates will be done in database.

The system has admin who can add new employees or can remove employees; also admin can add or can remove products. Also, he can see the availability of products. Admin can change other employees Role to "Moderator" or "Admin. The System has employee also who can perform tasks like add or remove products. Also, employee can add delivery person. Employee can edit product details.

**Implementation Technologies:**

1. **Spring Framework:**

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

Spring enables you to build applications from “plain old Java objects” (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

**1.1 Features of Spring Framework:**

**1. Lightweight**

Spring is modular lightweight framework which allows you to selectively use any of its modules on the top of Spring Core.

**2. Inversion of Control (IOC)**

This is another top feature of spring framework where application dependencies are satisfied by the framework itself. Framework creates the object in runtime and satisfies application dependencies.

**3. Aspect Oriented Programming (AOP)**

Aspect Oriented Programming (AOP) is very popular in programming world and in Spring it is well implemented. Developer can use Aspect Oriented Programming (AOP feature of Spring to develop application in which business logic is separated from system services.

**4. Container**

Spring provides their own container for managing the bean lifecycle.

**5. MVC Framework**

Spring MVC Framework is used for developing MVC based web applications.

**6. Transaction Management**

Spring framework provides generic Transaction Management layer which can be used with or without J2EE (JEE) environment.

**7. JDBC Exception Handling**

Spring provides their own abstraction of JDBC exception which further simplifies the exception handling in program.

**1.2 Advantages of Spring Framework:**

**1. Solving difficulties of Enterprise application development**

Spring is solving the difficulties of development of complex applications, it provides Spring Core, Spring IoC and Spring AOP for integrating various components of business applications.

**2. Support Enterprise application development through POJOs**

Spring supports development of Enterprise application development using the POJO classes which removes the need of importing heavy Enterprise container during development. This makes application testing much easier.

**3. Easy integration other frameworks**

Spring designed to be used with all other frameworks of Java; you can use ORM, Struts, Hibernate and other frameworks of Java together. Spring framework does not impose any restriction on the frameworks to be used together.

**4. Application Testing**

Spring Container can be used to develop and run test cases outside enterprise container which makes testing much easier.

**5. Modularity**

Spring framework is modular framework and it comes with many modules such as Spring MVC, Spring ORM, Spring JDBC, Spring Transactions etc. which can used as per application requirement in modular fashion.

**6. Spring Transaction Management**

Spring Transaction Management interface is very flexible it can configure to use local transactions in small application which can be scaled to JTA for global transactions.

1. **The JDBC Template**

The central class of the Spring JDBC abstraction framework is the **JdbcTemplate** class that includes the most common logic in using the JDBC API to access data, such as handling the creation of connection, statement creation, statement execution, and release of resource. The **Jdbc-Template**class can be found in the **org.springframework.jdbc.core**package.

The **JdbcTemplate** class instances are thread-safe once configured. A single **JdbcTemplate** can be configured and injected into multiple DAOs.

We can use the **JdbcTemplate** to execute the different types of SQL statements. **Data Manipulation Language** (**DML**) is used for inserting, retrieving, updating, and deleting the data in the database such as **SELECT**, **INSERT**, or **UPDATE** statements

**2.1** **MySQL**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**Features of MySQL:**

* **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

* **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

* **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

* **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

* **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

1. **Hardware and Software Requirements (Minimum):**

**Hardware:**

1. Intel i5 processor

2. 8 GB ram.

3. Windows 10

4. Data Connection 150 Mbps

**Software:**

1. Eclipse IDE 2021-09(4.21.0)
2. MySQL 8.0.26 with Workbench 8.0
3. Microsoft Edge Version 100.0.1185.39 (Official build) (64-bit)
4. Apache Tomcat Server 9.0
5. Maven Dependencies
6. **ER Diagram:**

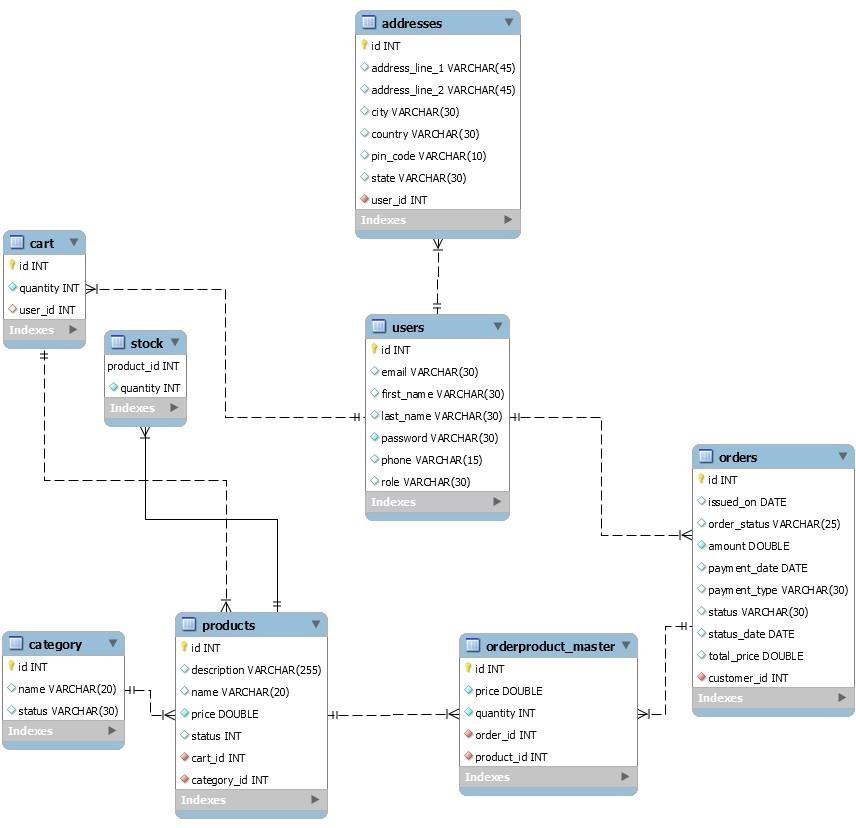


Figure 1: ER Diagram

1. **Table Structures:**
   * 1. Table Name: **Users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int | No | PRI | NULL | auto\_increment |
| email | varchar(30) | Yes | UNI | NULL |  |
| First\_name | varchar(30) | Yes |  | NULL |  |
| Last\_name | varchar(30) | Yes |  | NULL |  |
| password | varchar(30) | No |  | NULL |  |
| phone | varchar(15) | Yes |  | NULL |  |
| role | varchar(255) | Yes |  | Null |  |

* + 1. Table Name: **Addresses**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int | No | PRI | NULL | auto\_increment |
| address\_line\_1 | varchar(30) | Yes | UNI | NULL |  |
| address\_line\_2 | varchar(30) | Yes |  | NULL |  |
| city | varchar(30) | Yes |  | NULL |  |
| country | varchar(30) | No |  | NULL |  |
| Pin\_code | varchar(15) | Yes |  | NULL |  |
| state | varchar(255) | Yes |  | Null |  |
| User\_id | int | No | MUL | NULL |  |

* + 1. Table Name: **Cart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** | **Extra** |
| id | int | NO | PRI | NULL | auto\_increment |
| quantity | int | NO |  | NULL |  |
| customer\_id | int | YES | MUL | NULL |  |
| Product\_id | int | YES | MUL | NULL |  |

* + 1. Table Name: **Categories**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** | **Extra** |
| id | int | NO | PRI | NULL | auto\_increment |
| name | varchar(30) | YES |  | NULL |  |
| status | Varchar(30) | YES |  | NULL |  |

* + 1. Table Name: **Products**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int | No | PRI | NULL | auto\_increment |
| description | varchar(100) | Yes | UNI | NULL |  |
| name | varchar(30) | Yes |  | NULL |  |
| price | double | Yes |  | NULL |  |
| status | varchar(30) | No |  | NULL |  |
| category\_id | int | Yes |  | NULL |  |

* + 1. Table Name: **Stocks**

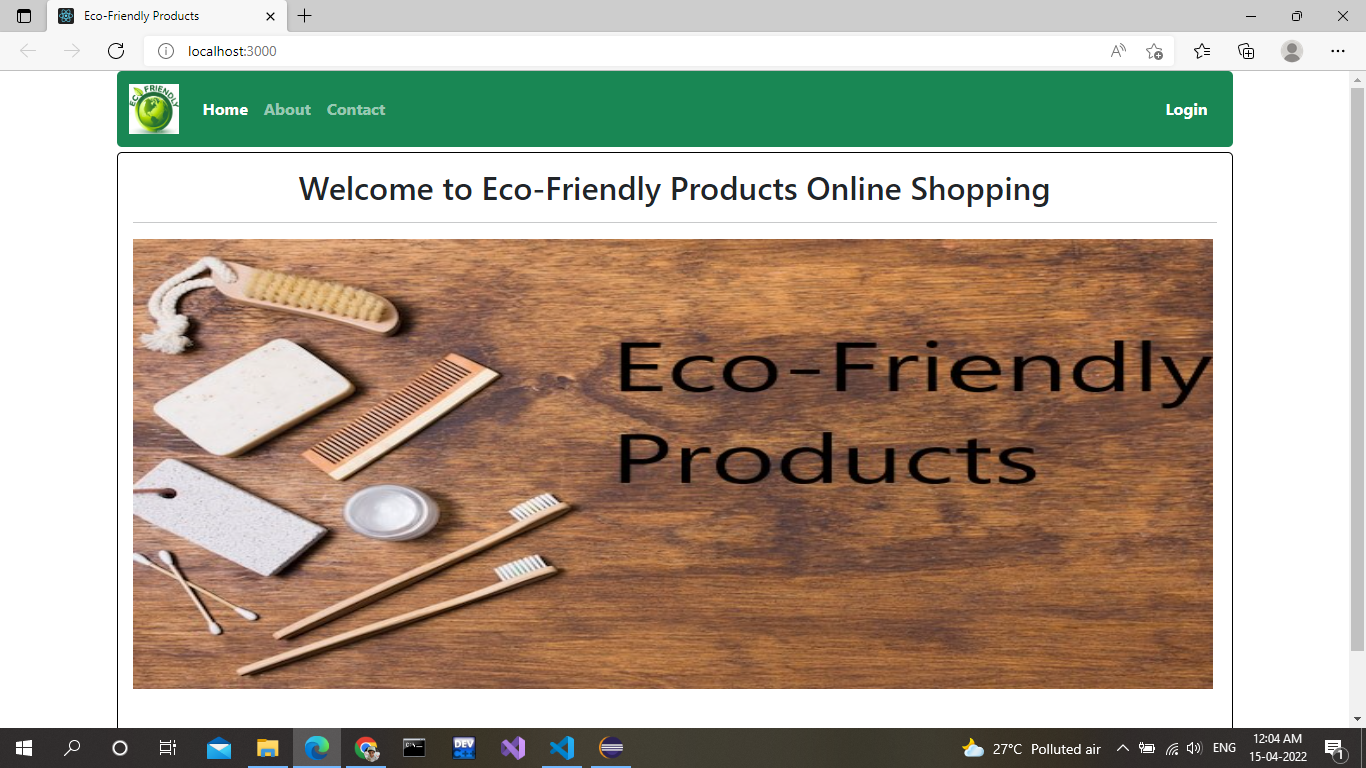
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Product\_id | int | No | PRI | NULL |  |
| quantity | int | No |  | NULL |  |
| unit | varchar(10) | Yes |  | NULL |  |

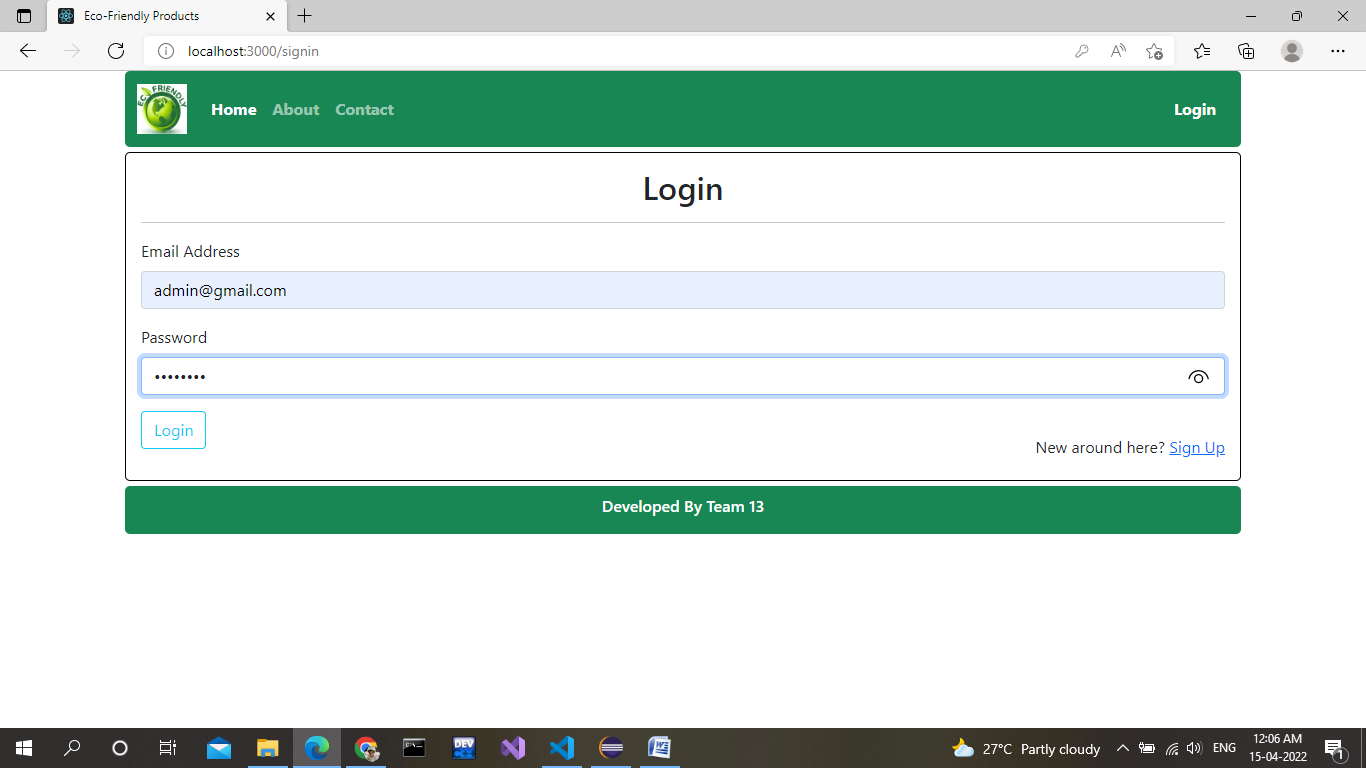
1. **UML Diagrams:**

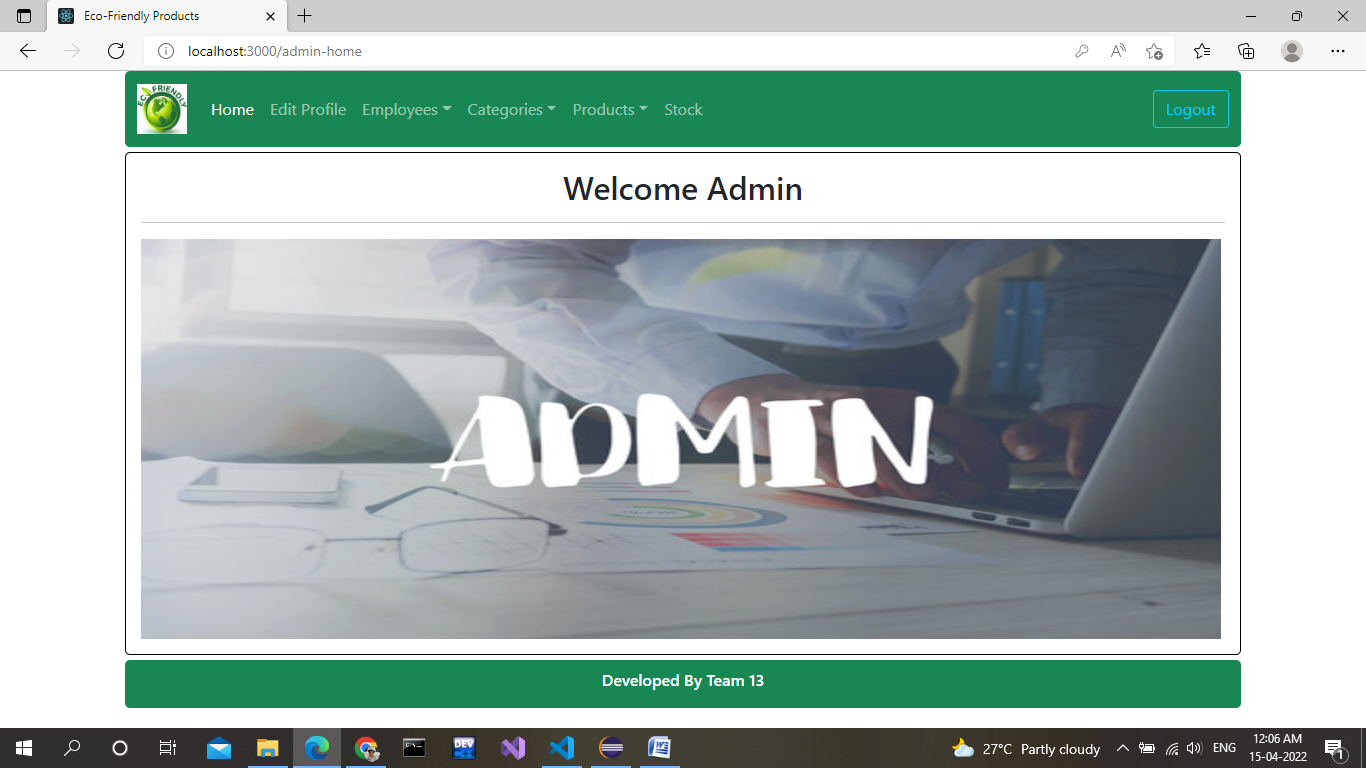
****

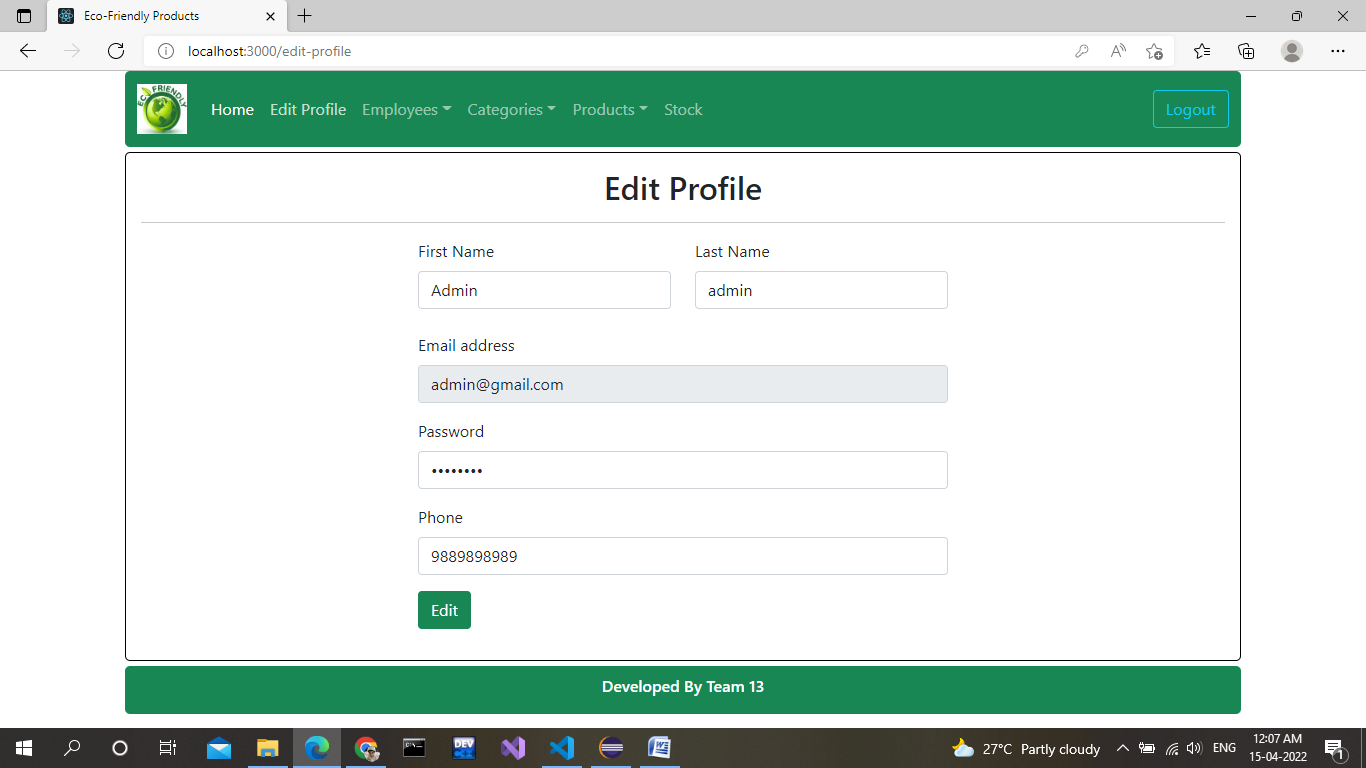
Figure 2: UML Diagram

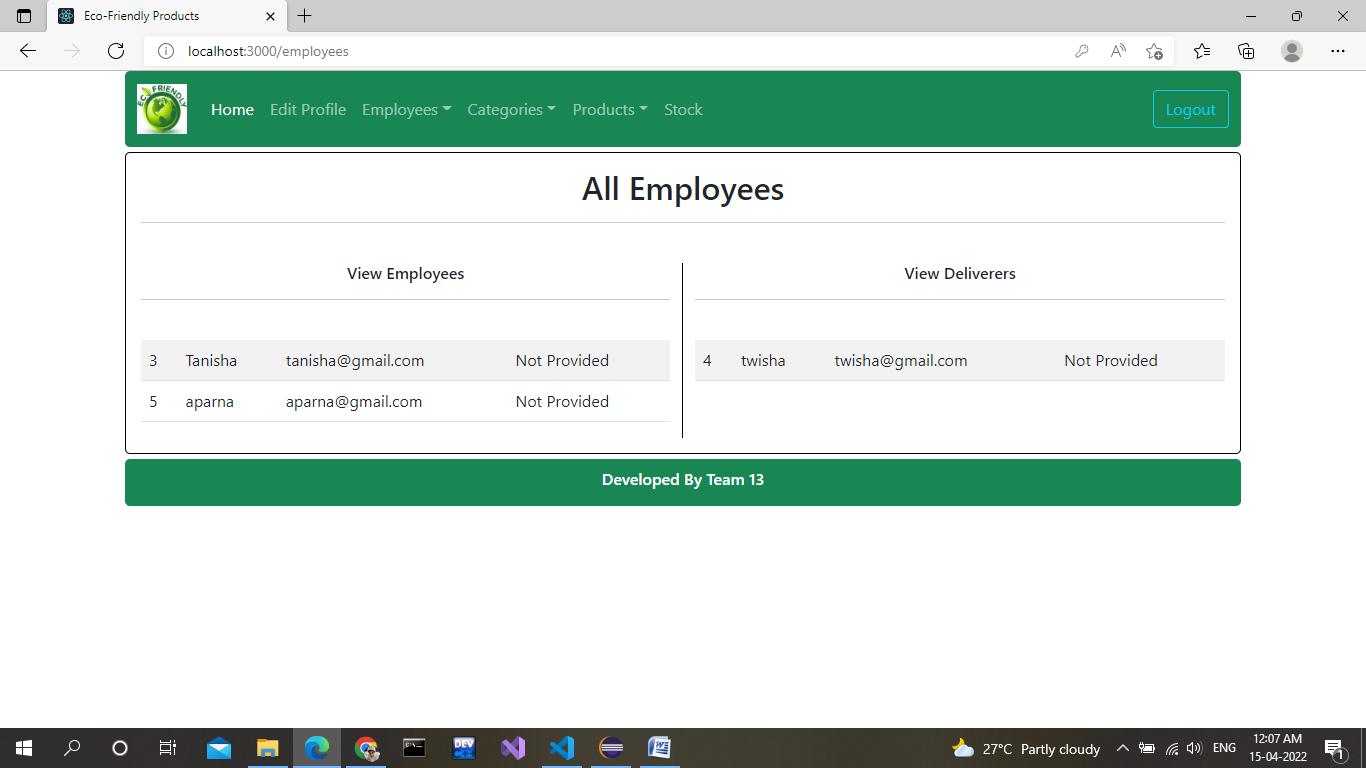
1. **Project Screen Shots**

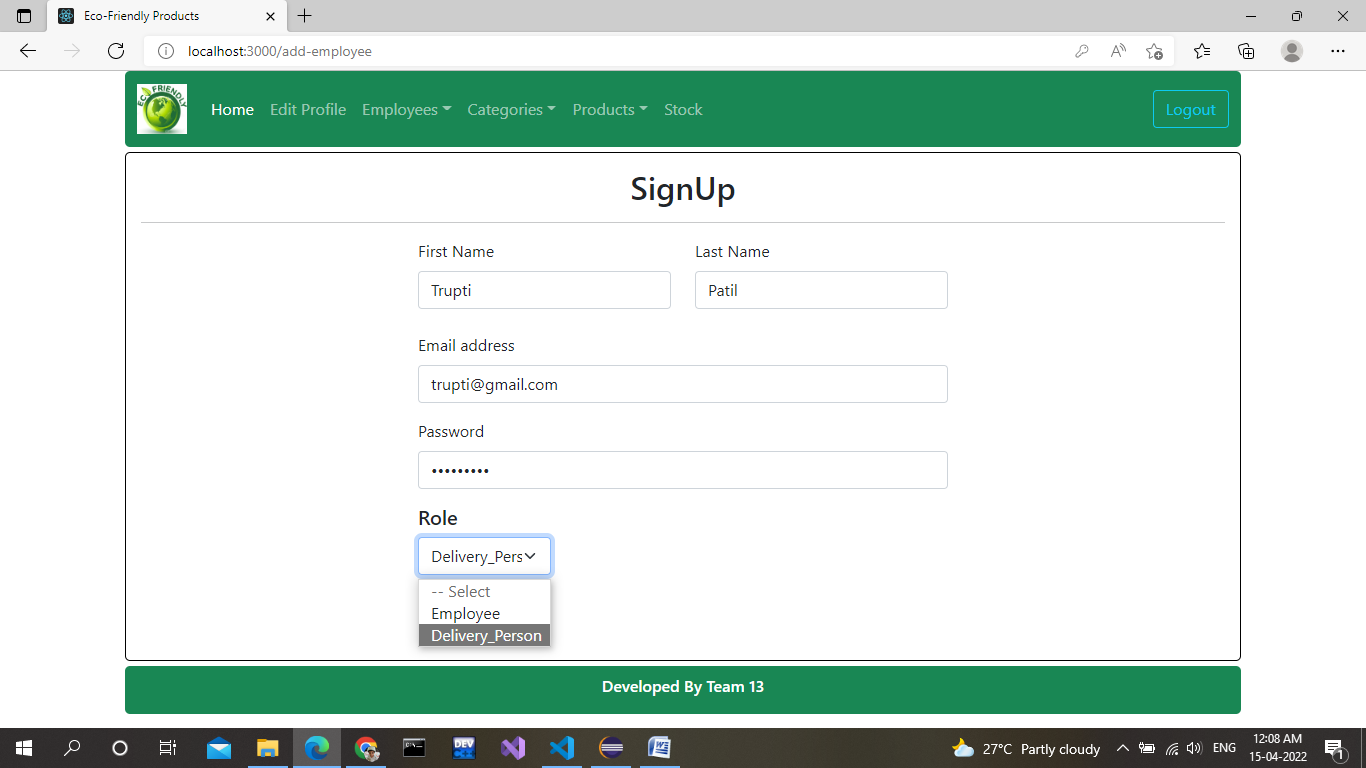
****

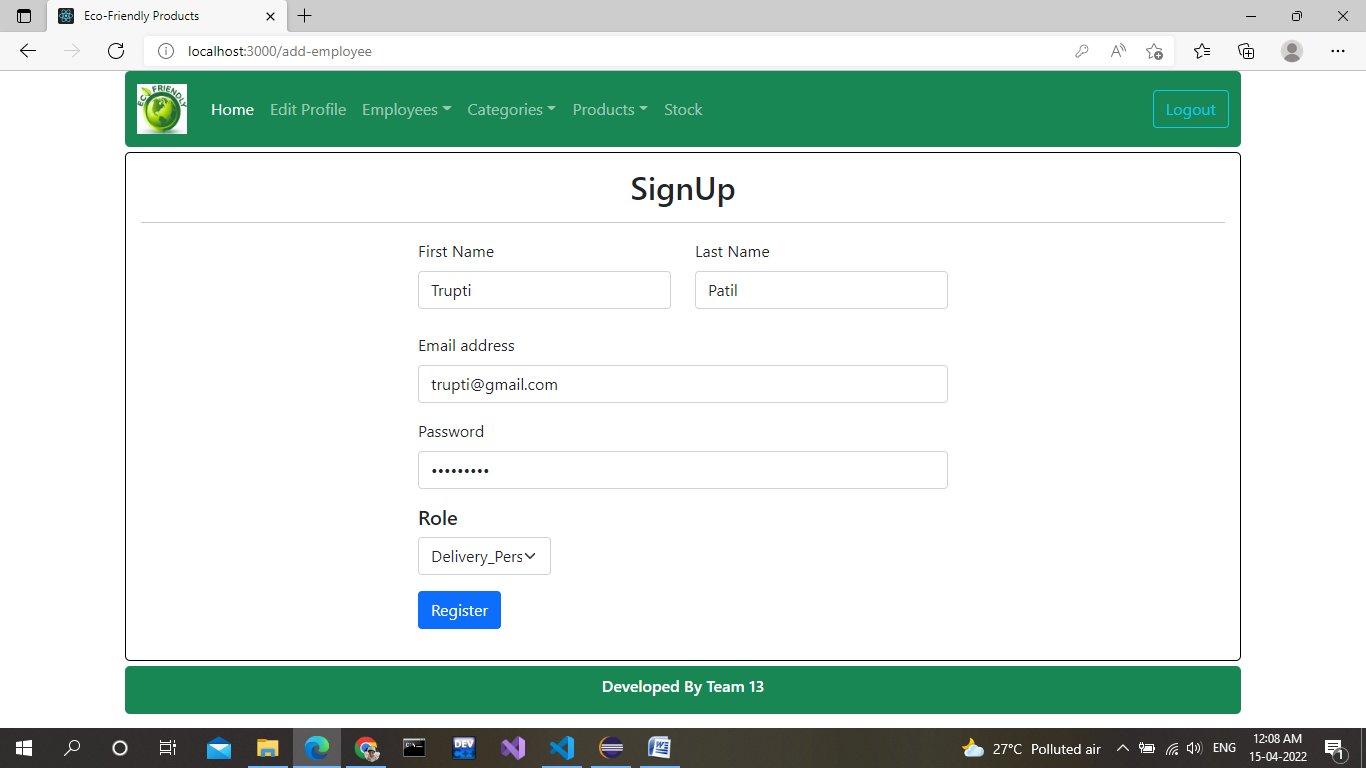
****

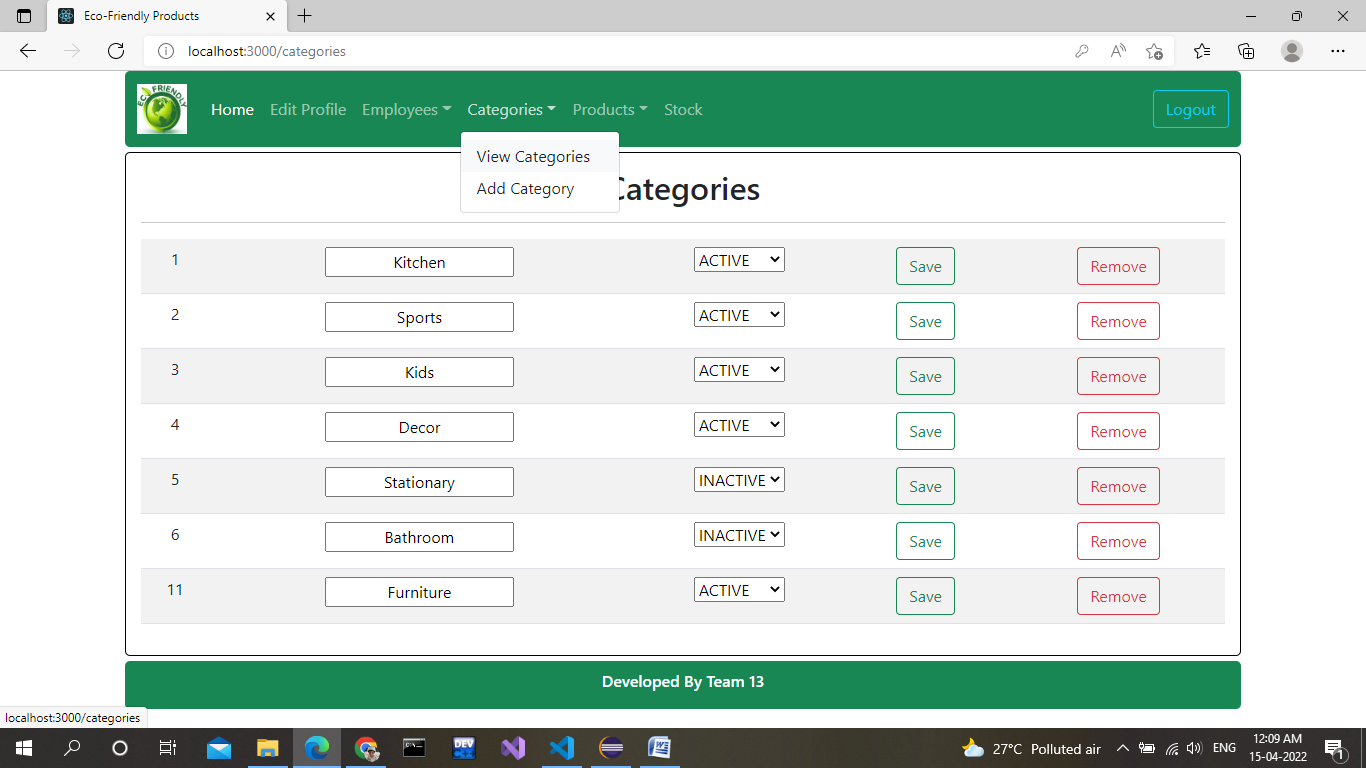
****

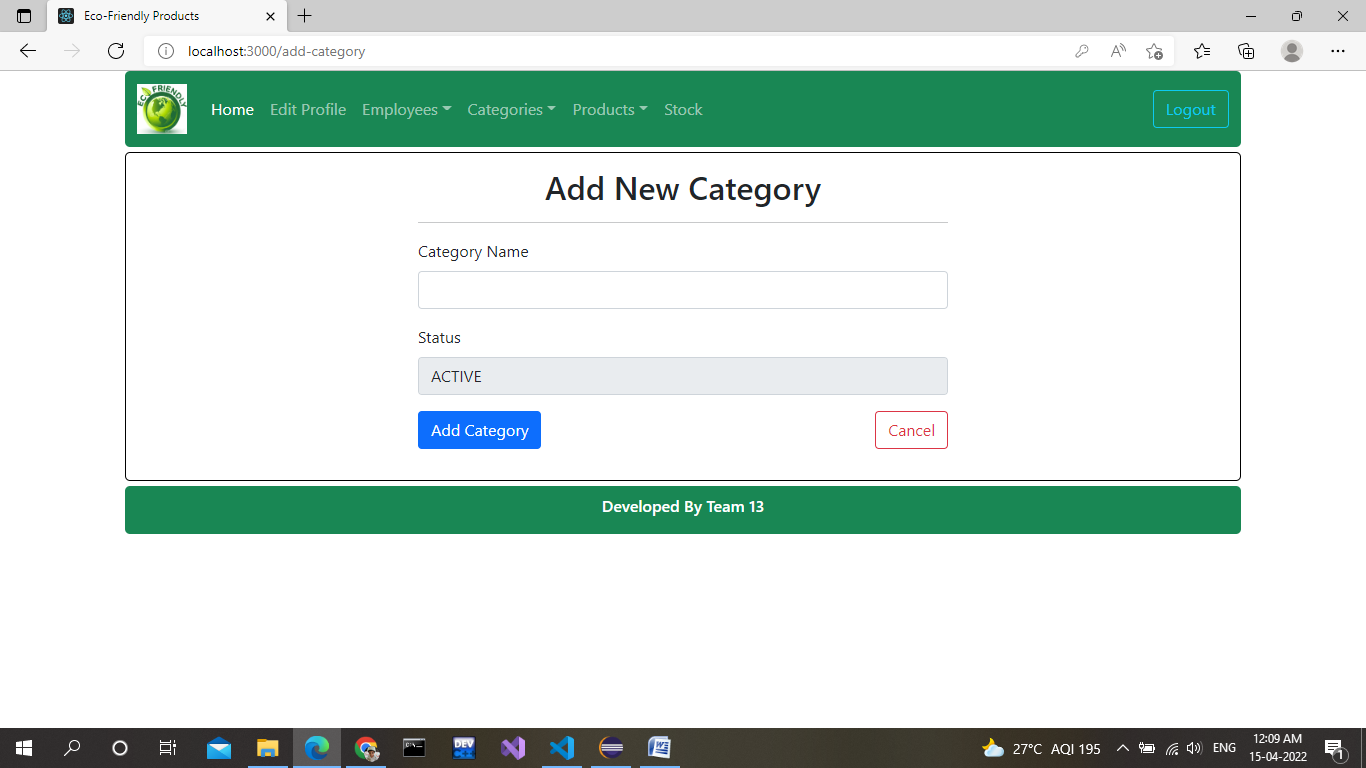
****

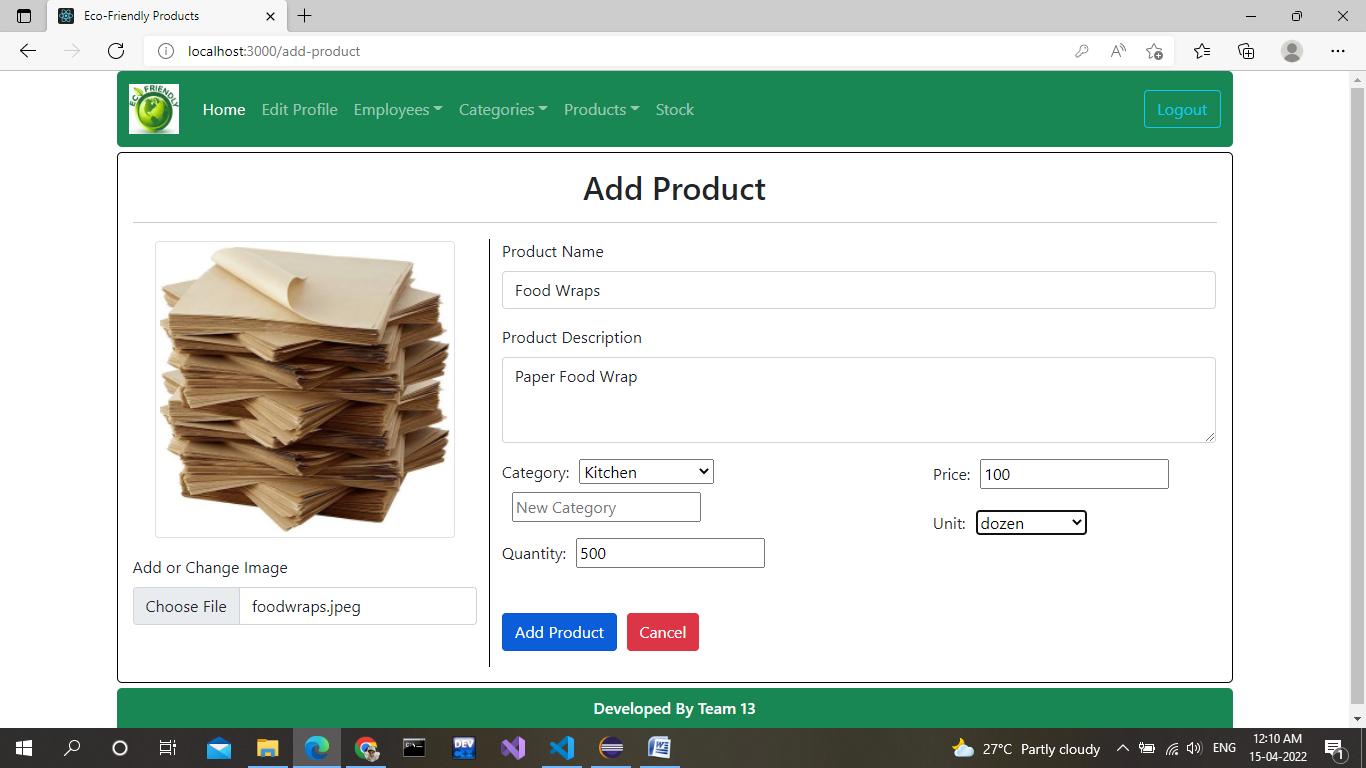
****

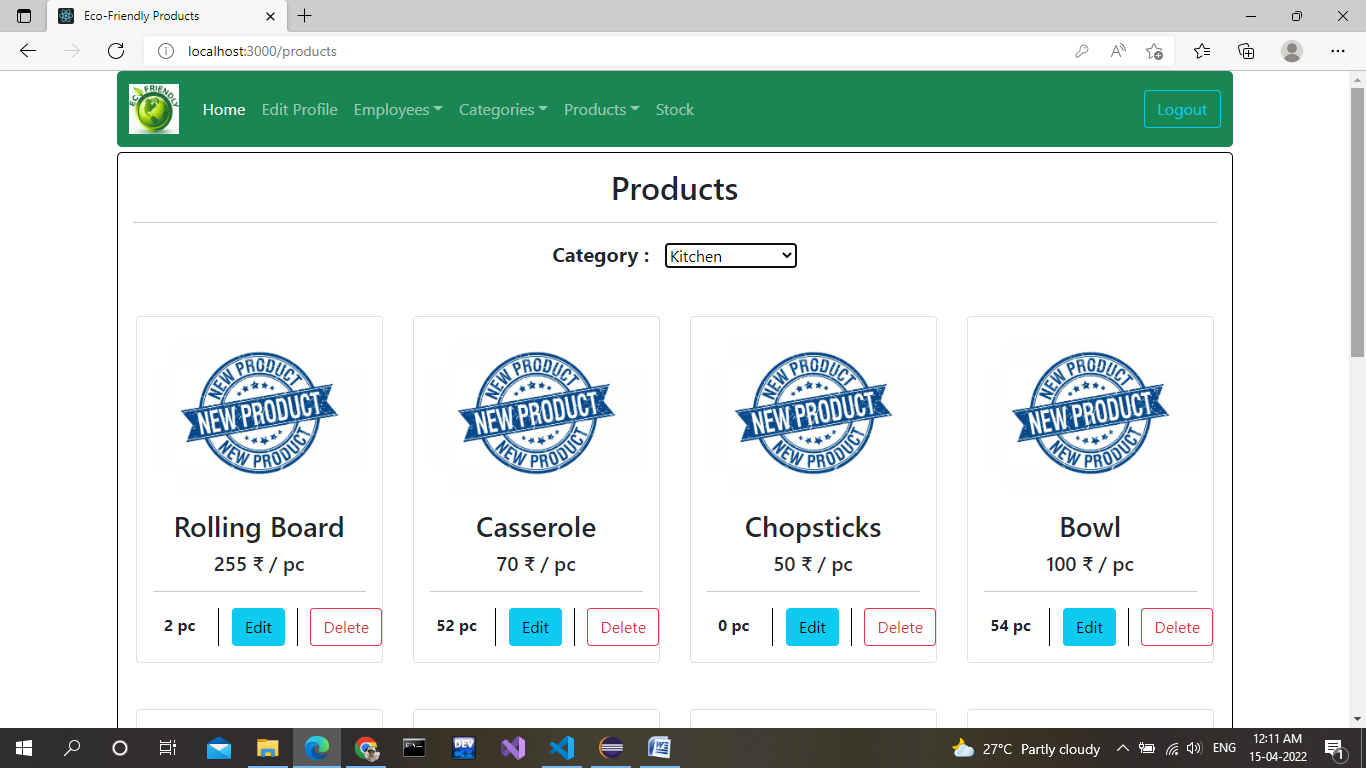
****

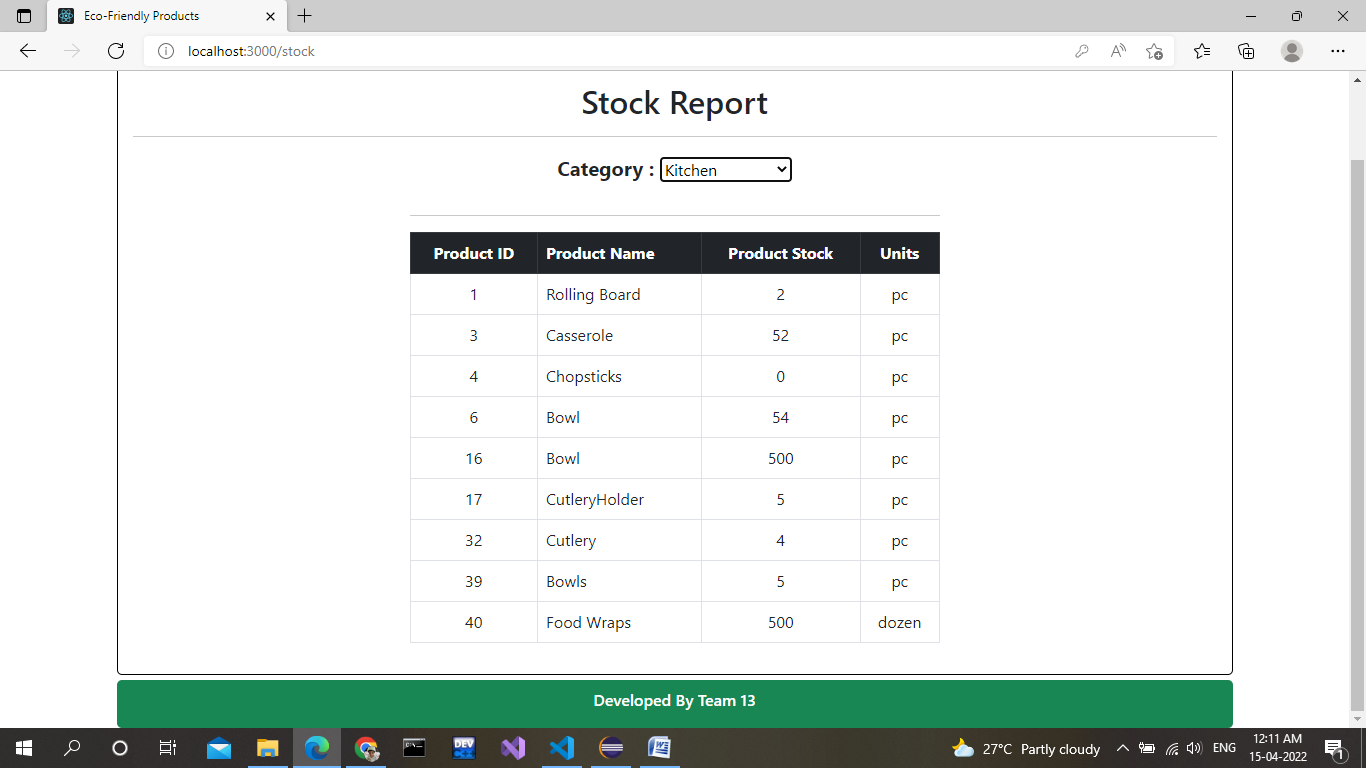
****

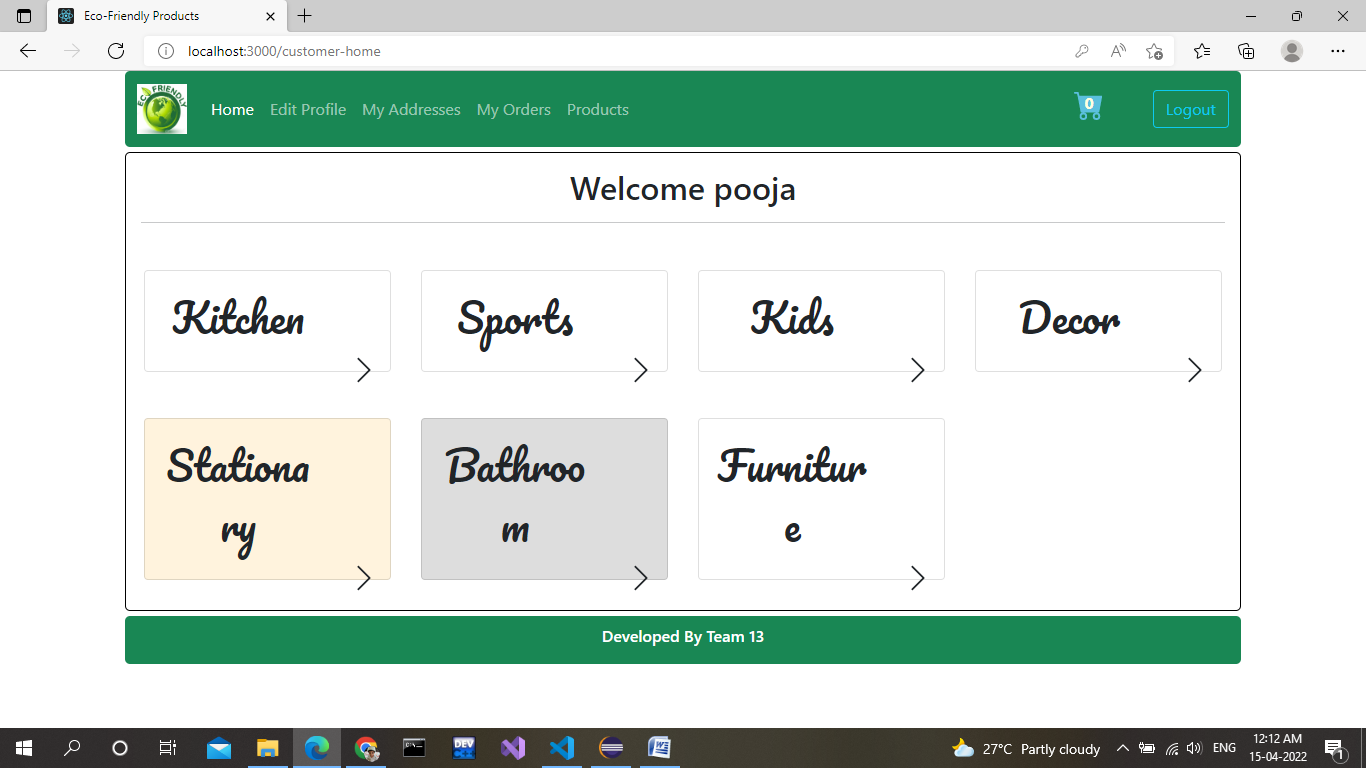
****

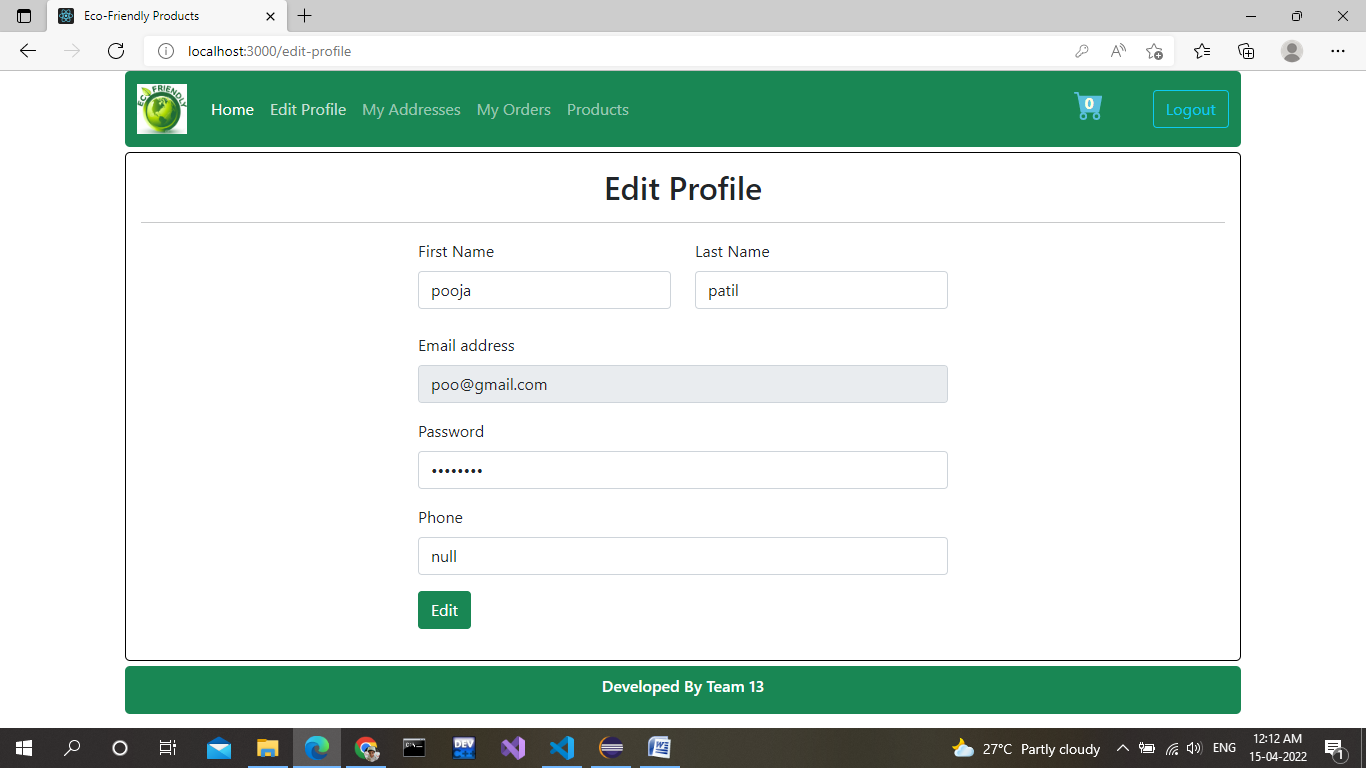
****

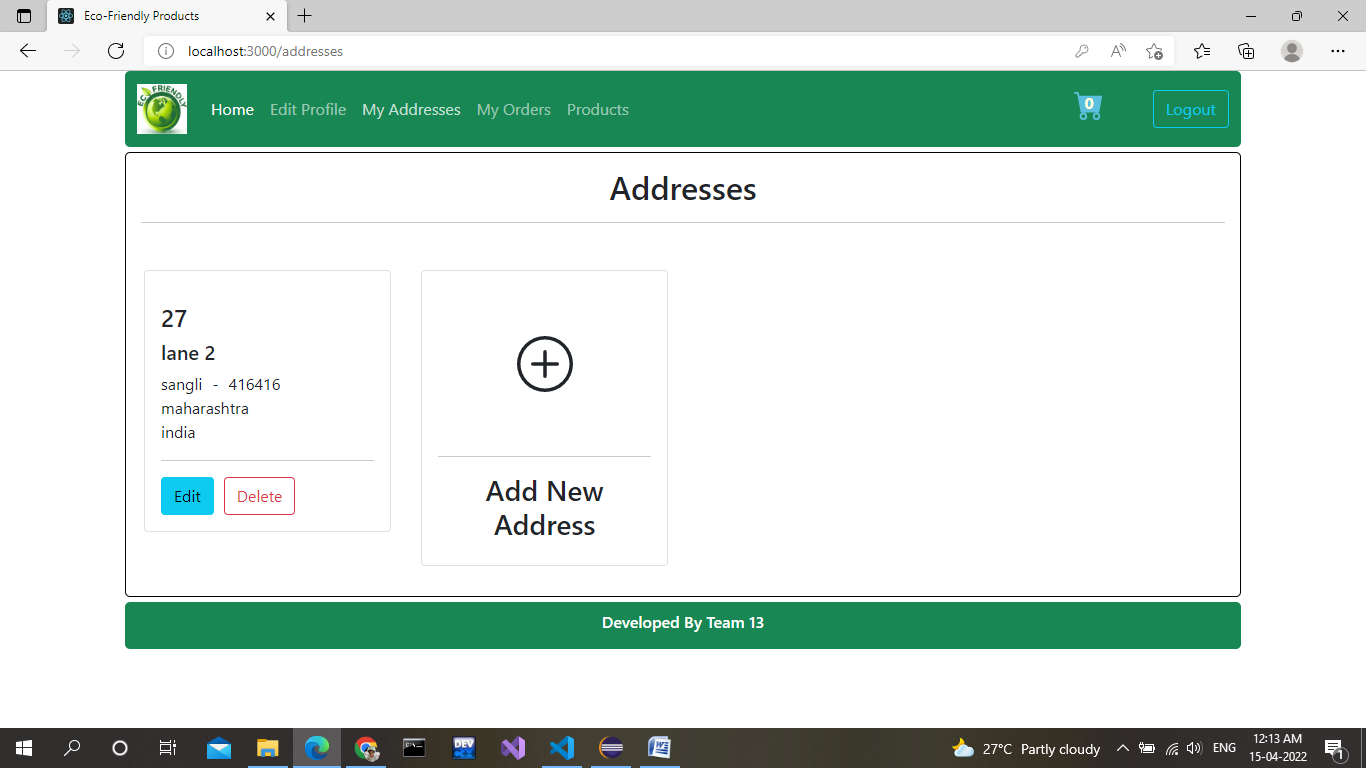
****

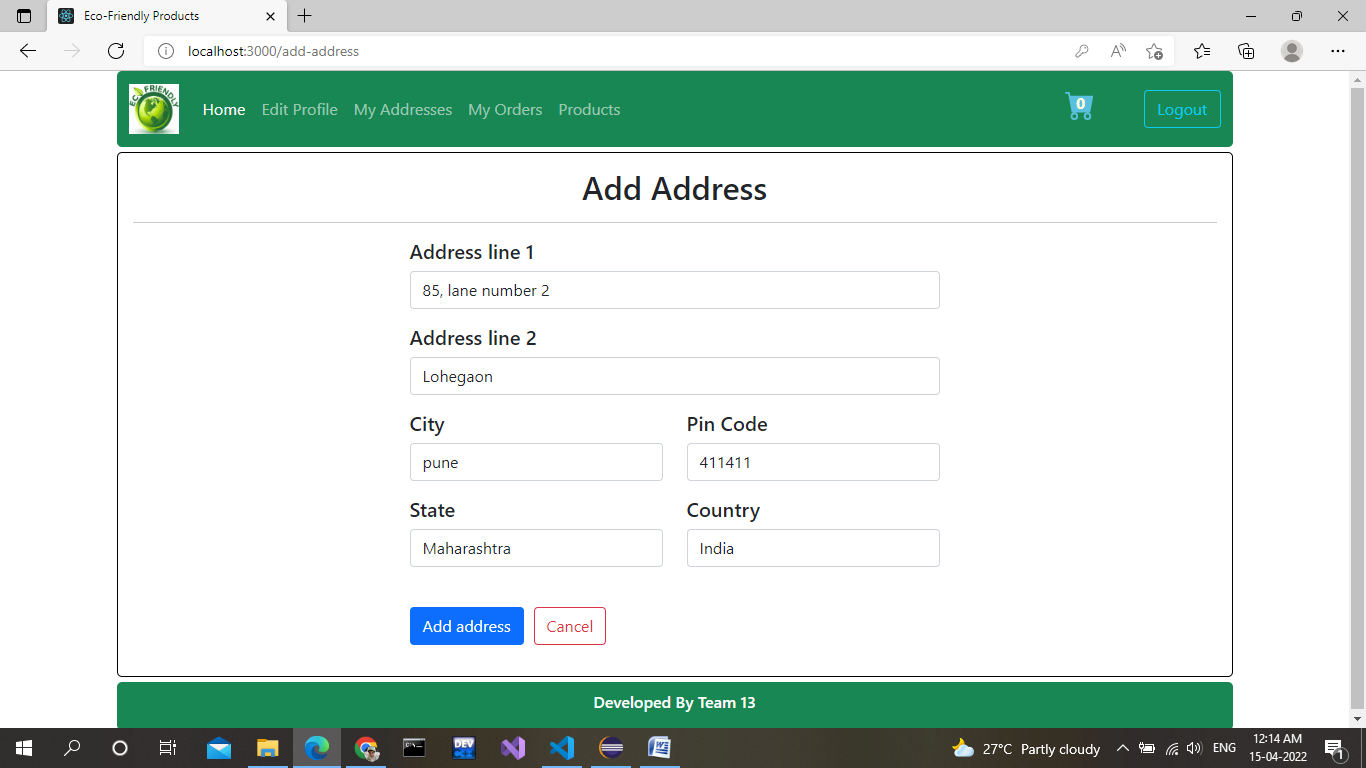
****

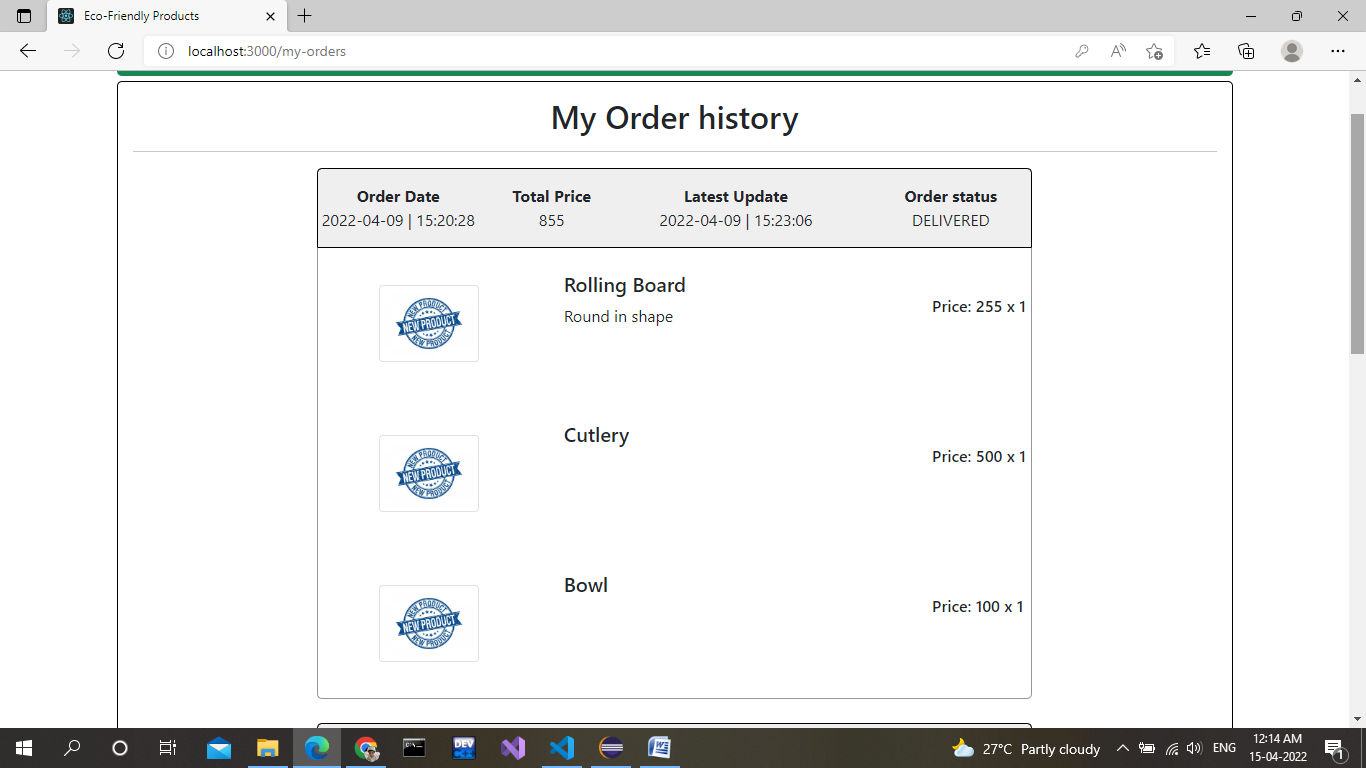
****

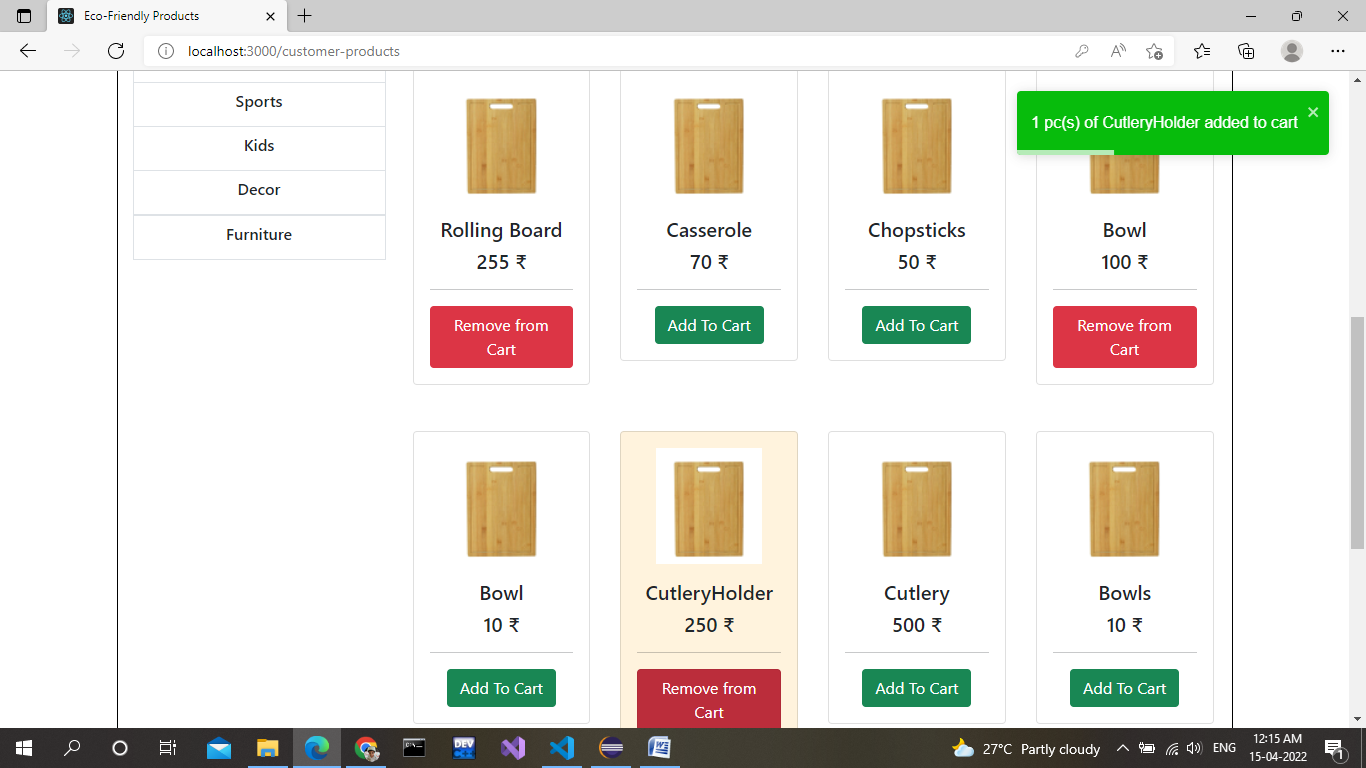
****

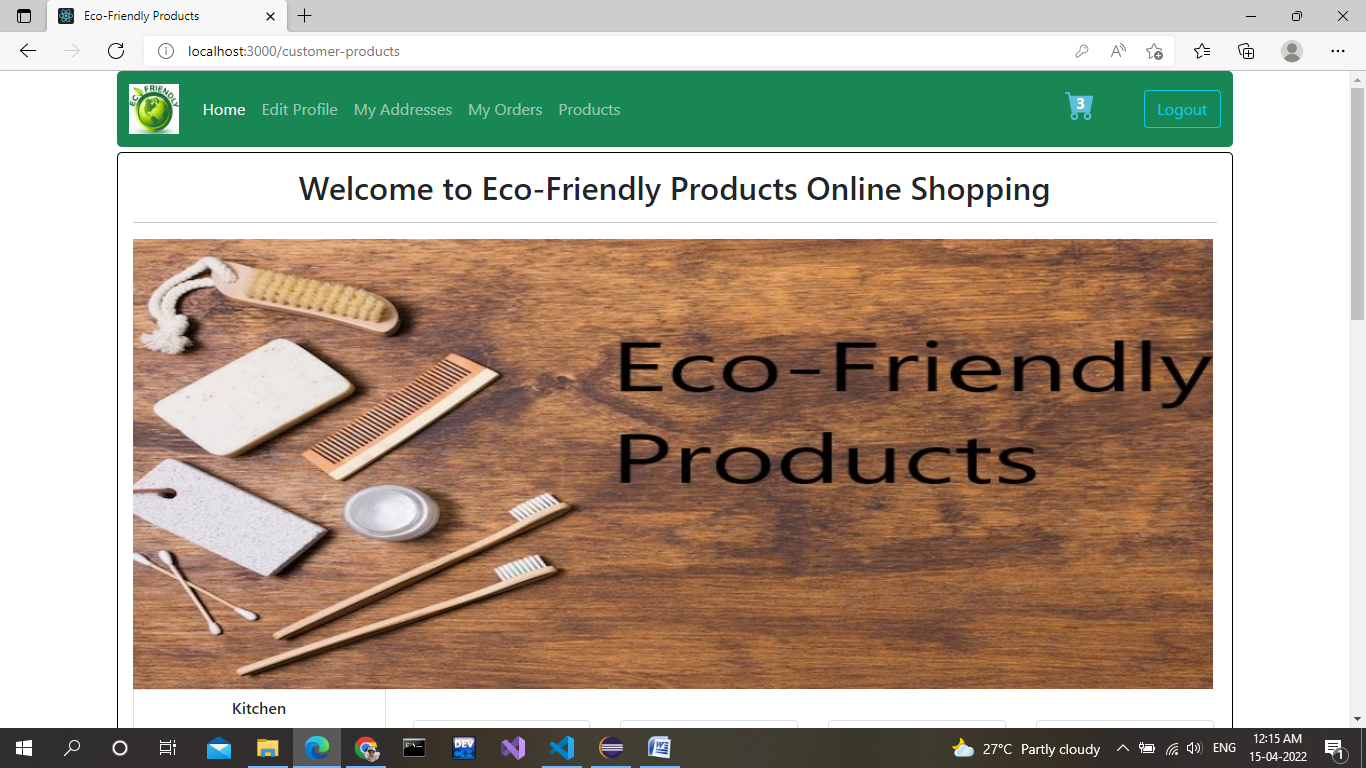
****

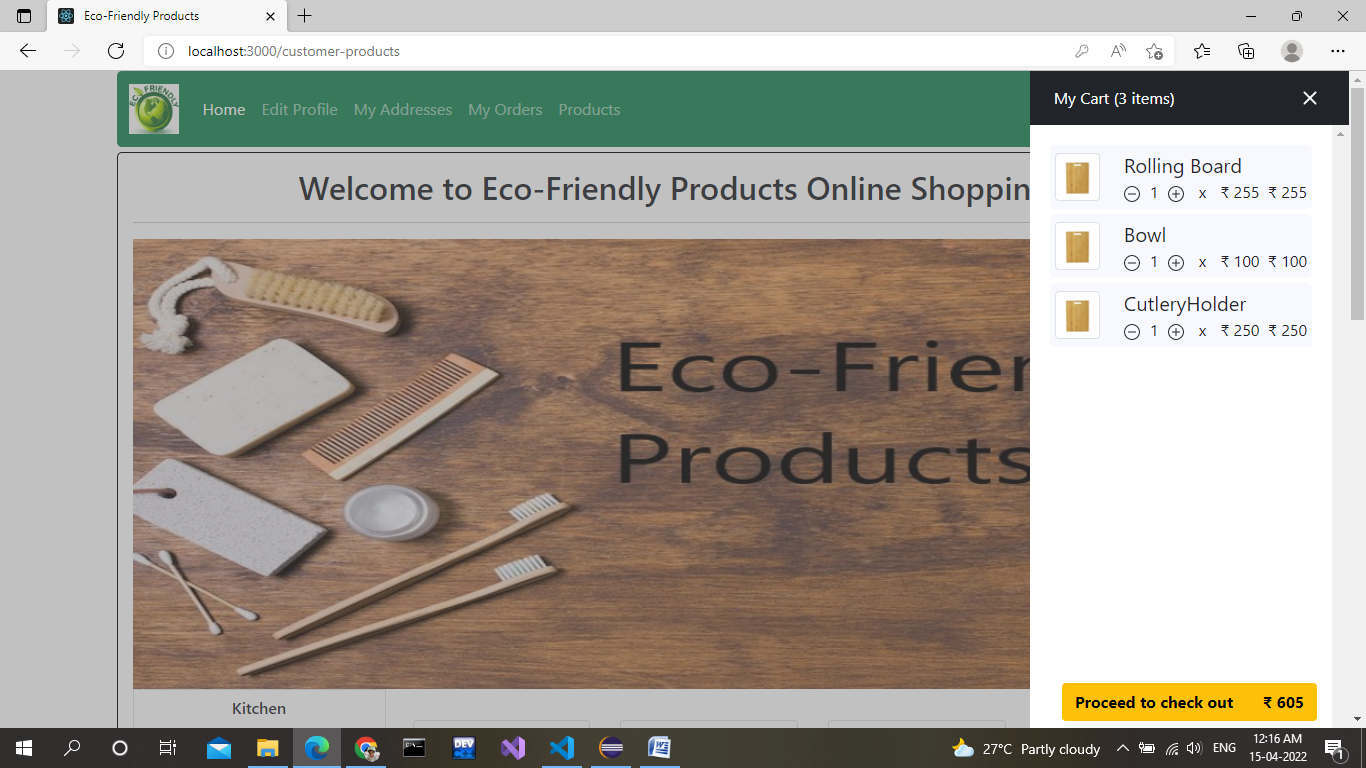
****

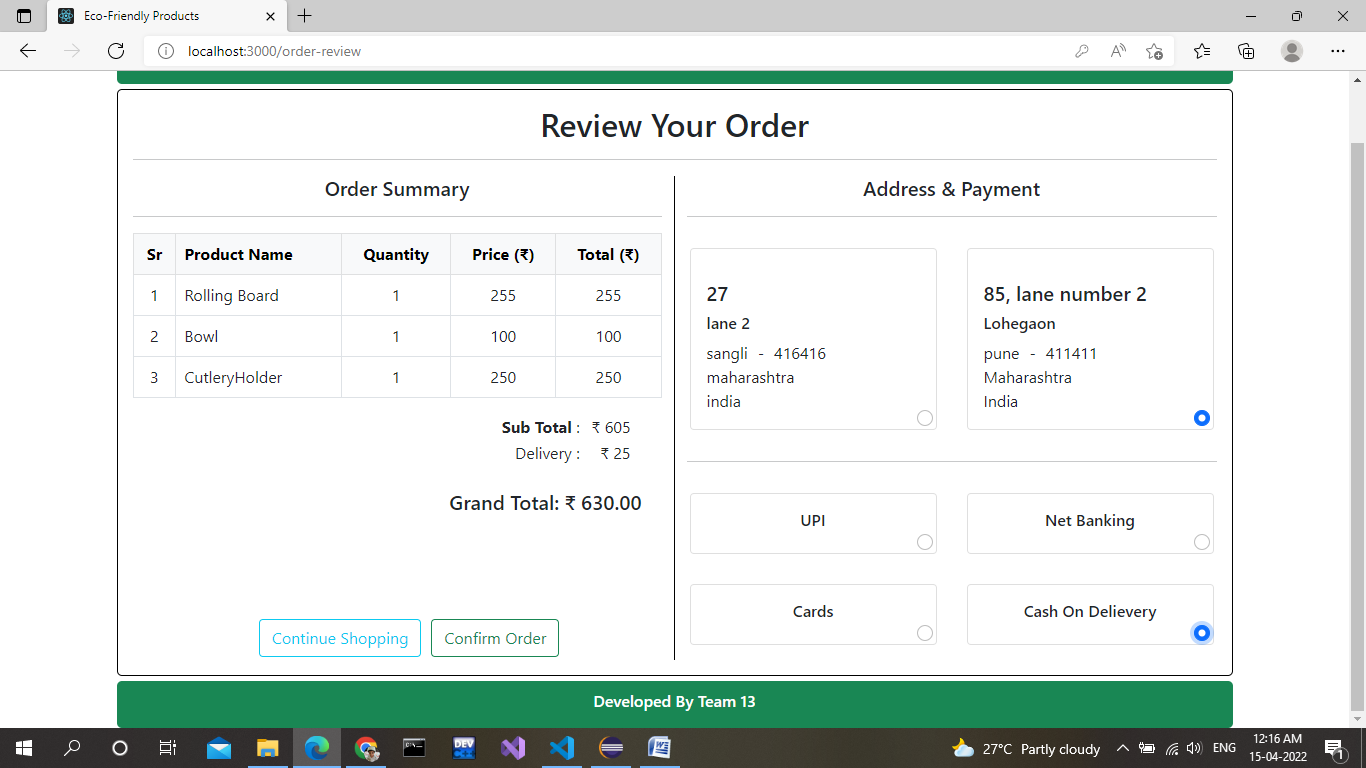
****

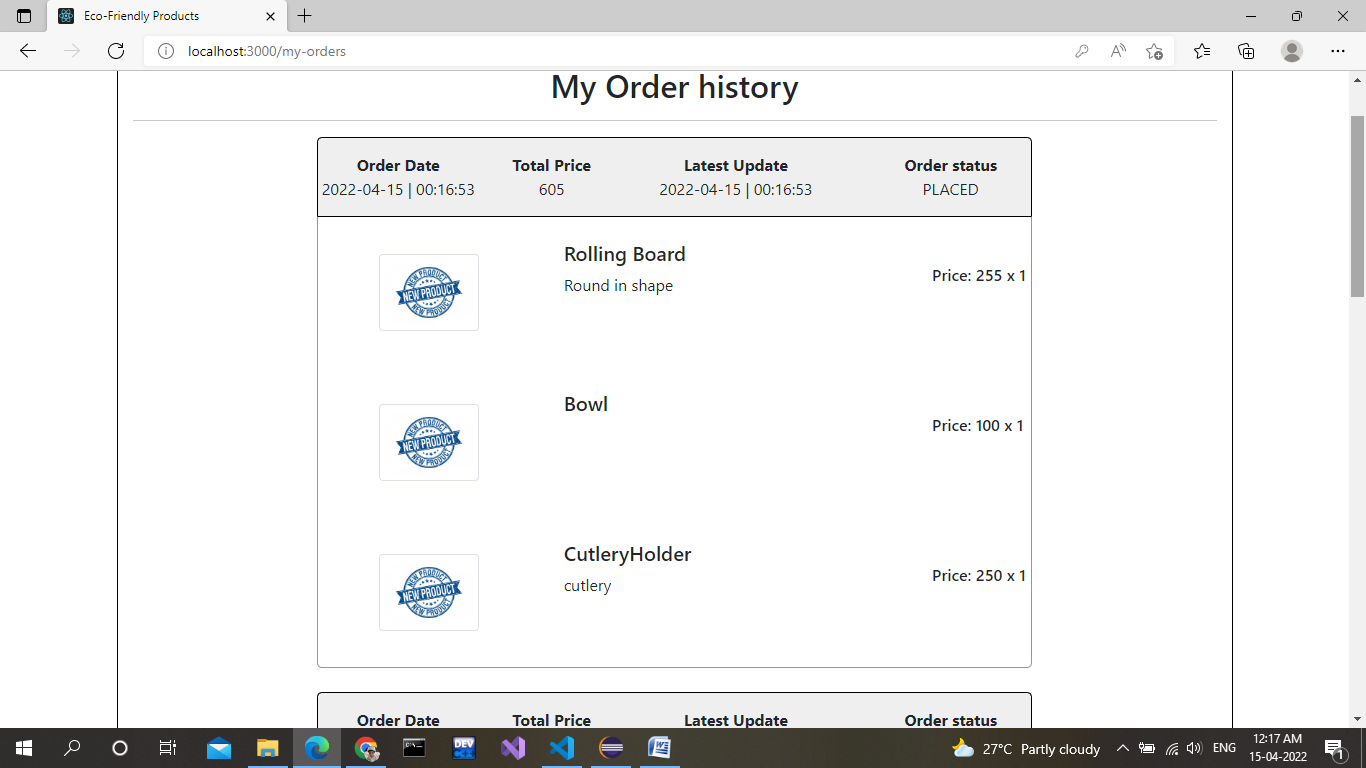
****

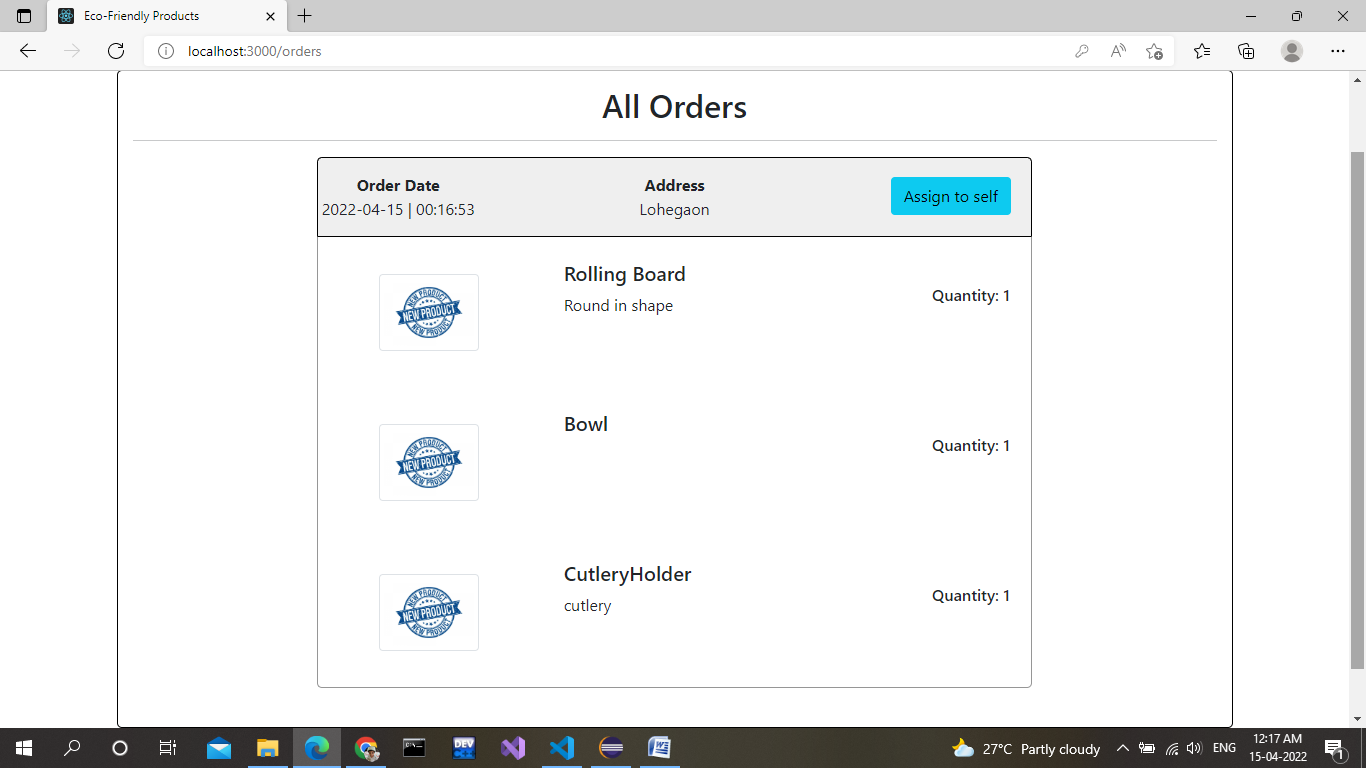
****

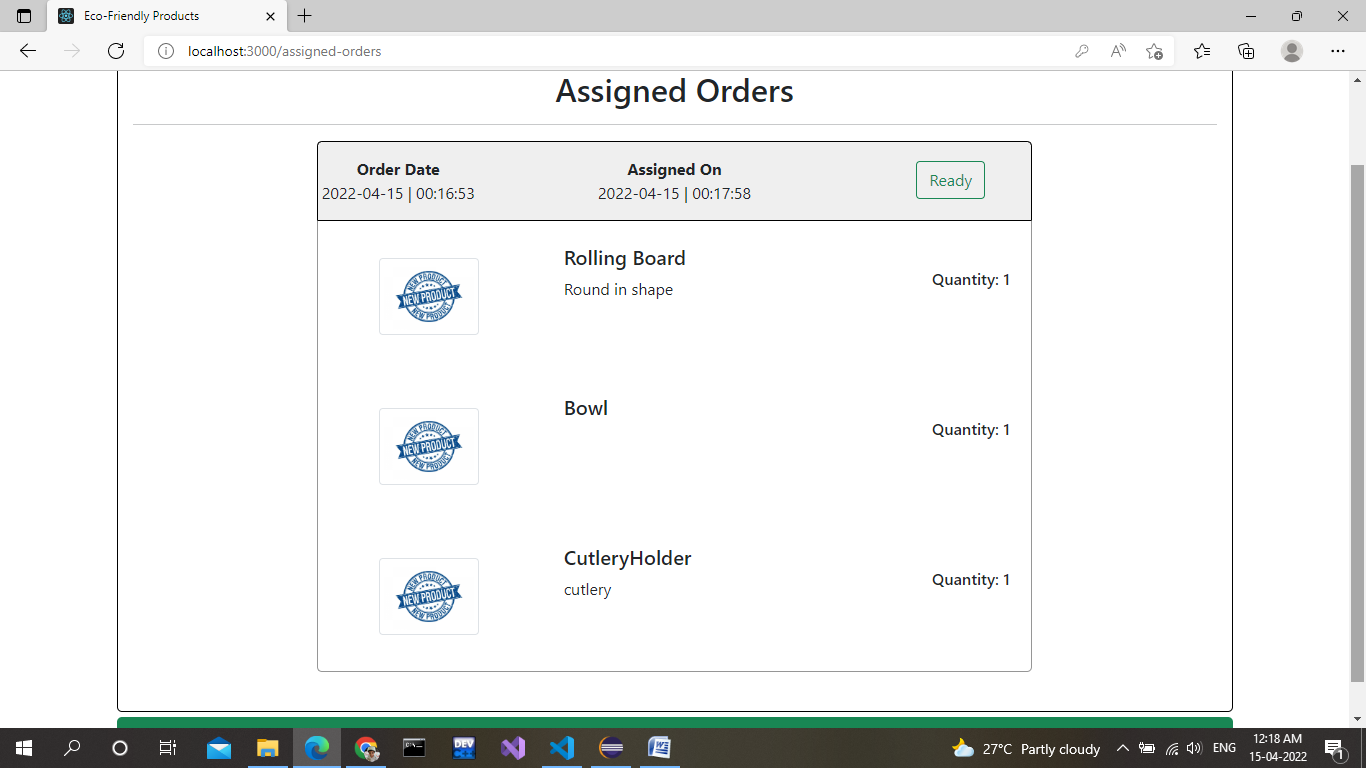
****

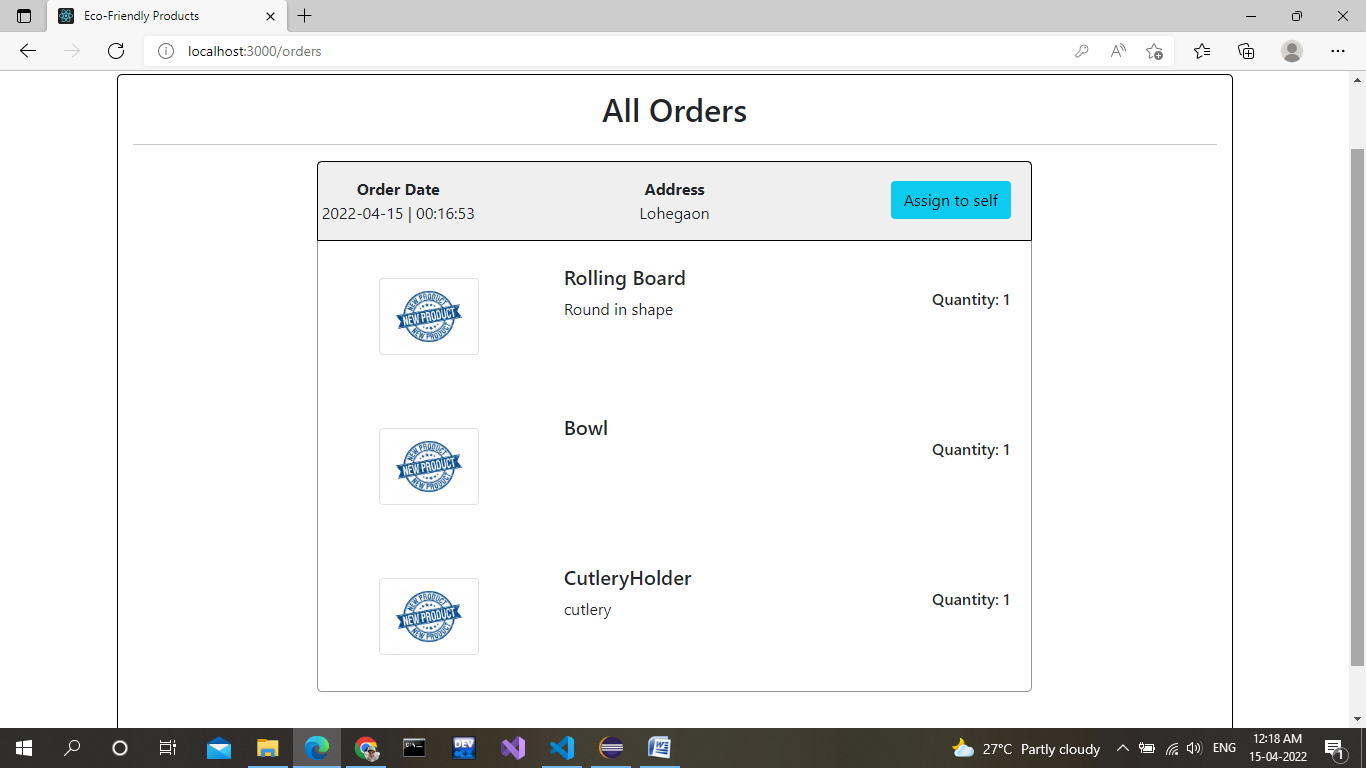
****

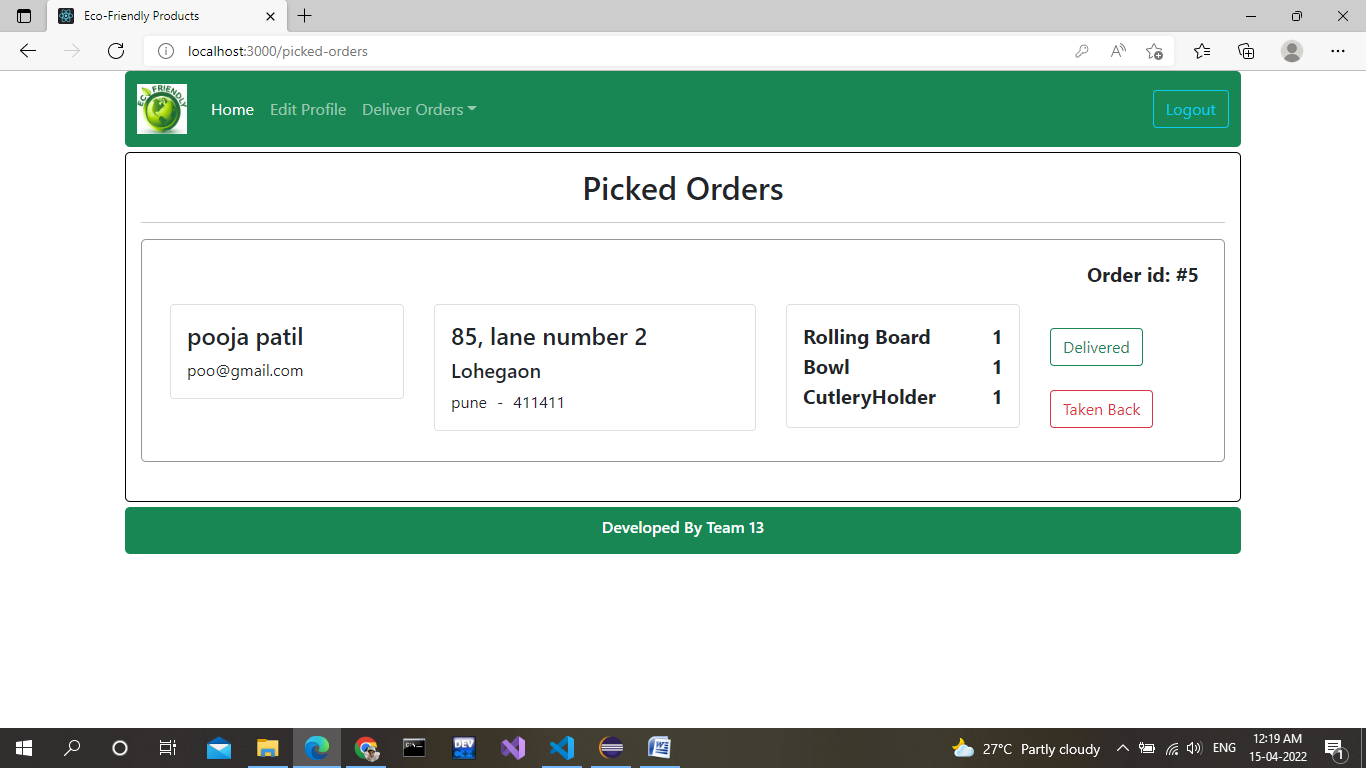
****

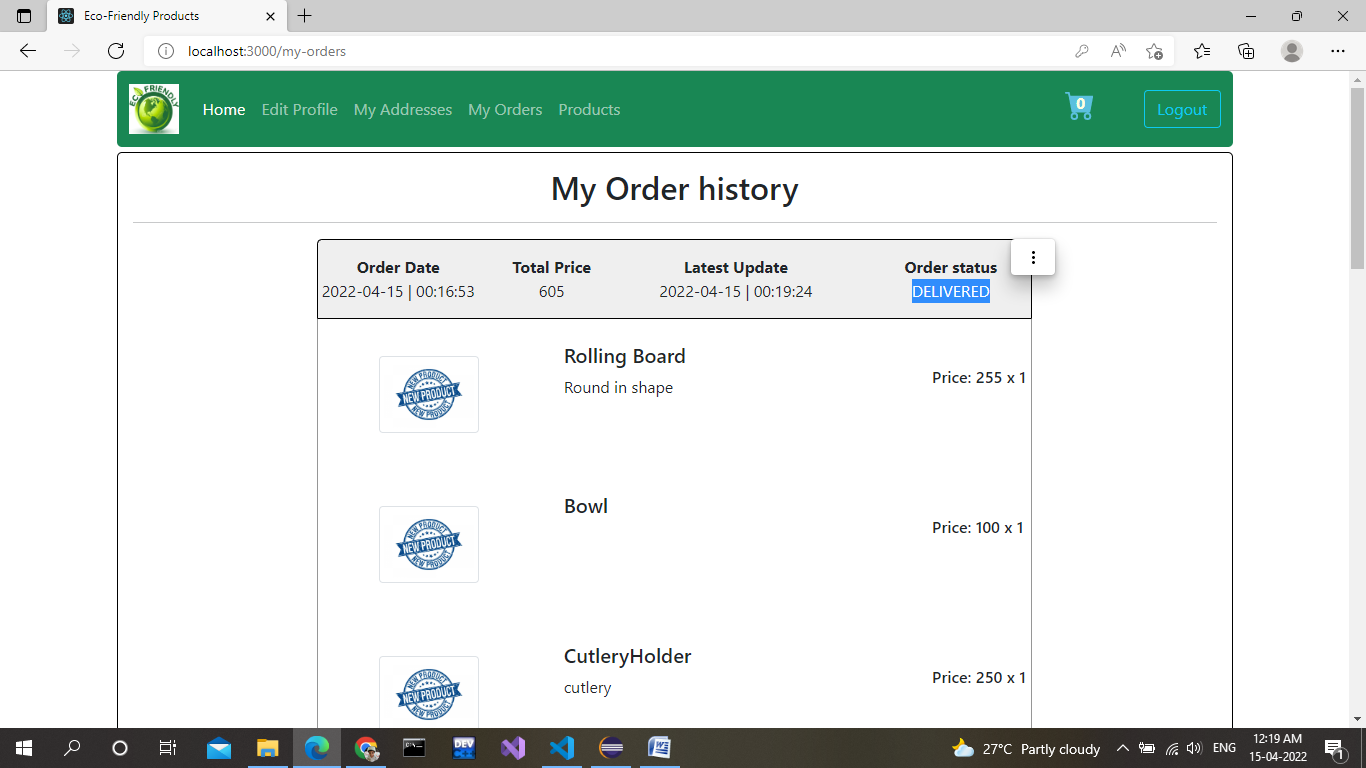
****

****

****

****

****

****

1. **End to End Flow of Application:**

**User:**

* 1. User will login to the portal or will have to register if he is not a registered user.
  2. After registration User will login and Dashboard page will be displayed to him which will display the categories of Eco-Friendly Products.
  3. From that page can User can click on the categories from which he want to order products.
  4. After that he can add the products to the cart.
  5. From cart he can proceed to order those products by selecting address and payment method.
  6. User can also see his order history and status of his order.

**Admin:**

1. Admin will login as Admin from the home page and Admin page will open.
2. Admin can edit his own profile and he can even add employees and delivery person.
3. Admin can add, remove and update Categories.
4. He can even make categories Active and inactive as per products availability.
5. He can add, remove and update Products. Products are added category wise.
6. He can even check stock of the products.

**Employees:**

1. Employee can login to his account. He cannot register. Admin will add the employee.
2. Employee can add remove and update categories as well as products.
3. He can assign the order to the delivery person.
4. He can even edit his own profile.

**Delivery-Person:**

1. Delivery person can login to his profile. He also cannot register. Admin will add the delivery-person.
2. He can edit his own profile.
3. He can accept the order assigned by the employee.
4. And he can select whether he has to deliver or take it back.
5. Future Scope of Project:

Future scope of this project is we can make mobile application to

Access this facility remotely. And also we can allow local vendors to add their products to our website.

**Thank You!**