Cairo University
Faculty of Computers and Artificial Intelligence Department of Software Engineering

Green Wave

Supervised By

Prof Dr. Khaled Wassef
TA. Ashraf Mohey

Implemented By

Ziad Diaa Ezzat	***20206107***
Mawada Mohamed	***20206079***
Malak Tarek Ahmed	***20206075***
Nada Mohamed Samy	***20206083***
Farah Mohamed	***20206054***

Graduation Project
Academic Year 2023-2024
Final-year Documentation

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Motivation (Abstract)

The project aims to develop a comprehensive platform that addresses various aspects of sustainable living and environmental conservation. **The platform**, named **GreenWave**, will offer a wide range of functionalities to promote eco-friendly practices and community engagement.

Users will be able to register and authenticate their accounts securely, ensuring the protection of their personal data. **GreenWave** will facilitate waste collection services, allowing users to schedule waste collection appointments, access information on acceptable waste types and collection fees, and receive timely reminders and confirmations for their appointments. Additionally, the platform will provide agricultural education resources, including lessons, tutorials, and guides on home gardening and crop cultivation techniques, categorized for easy access.

Furthermore, **GreenWave** will offer an urban greening service, enabling users to request assistance for cultivating green spaces and access information on suitable land areas for urban greening. The platform will also feature an eco-friendly product marketplace, where users can purchase essential tools and supplies for sustainable practices, add products to a shopping cart, make payments, and track their orders.

In addition to these functionalities, **GreenWave** will incorporate waste reporting and reward systems, community engagement features, notifications and reminders, user feedback and support mechanisms, an admin dashboard for managing user accounts and platform analytics, guidance on e-waste recycling and disposal, information on green certifications and labels, providing users with a comprehensive solution for embracing eco-friendly practices and making a positive impact on the environment.

The Technologies that will be used in our project are:

- Front-end (React)
- Back-end (NodeJS)
- Database (MongoDB)
- (Chat Bot)

➢ Background

- The background for the **GreenWave** project stems from a growing global awareness of environmental sustainability and the need for collective action to address ecological challenges. With increasing concerns about climate change, resource depletion, and environmental degradation, there is a pressing need for innovative solutions that promote sustainable living and empower communities to engage in eco-friendly practices.

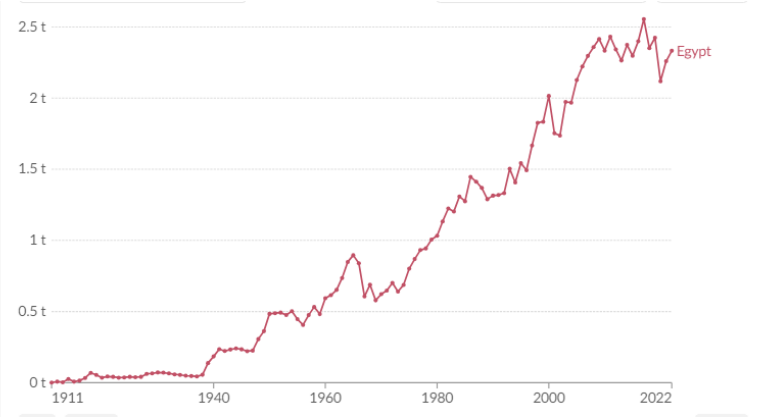
- The **GreenWave** project is inspired by the desire to create a comprehensive platform that not only educates and informs users about sustainable practices but also provides practical tools and resources to facilitate their adoption. The project recognizes the importance of addressing various aspects of sustainable living, including waste management, urban greening, eco-friendly product access, and sustainable transportation options.

➢ Problem Definition

- Urban living has brought about challenges related to waste management, lack of agricultural knowledge, limited green spaces, and a need for sustainable practices. Additionally, there is a growing interest in eco-friendly products and a lack of centralized platforms that address these interconnected issues. Users require a comprehensive solution that not only facilitates efficient waste collection but also provides agricultural education, urban greening services, and access to eco-friendly products. The existing landscape lacks a unified platform that seamlessly integrates these diverse services, limiting the ability of individuals to adopt sustainable and environmentally friendly practices in their daily lives.

Data

How much CO_2 does the average person emit in Egypt?



Annual emissions figures are often used to compare countries' contribution to climate change. But this metric often reflects differences in population size across the world. To understand the 'footprint' of the average person in a given country, this chart shows per capita emissions. These figures reflect 'production-based' emissions, so do not correct for traded goods.

CLIMATE CHANGES AND CONSEQUENCES:

TEMPERATURE INCREASING:

Climate change will lead to heightened temperatures and unpredictable rainfall patterns in Egypt. This weather shift is already underway as the North African country warms at an accelerating speed. Though average annual temperatures have increased at an overall rate of 0.1°C per decade between 1901 and 2013, the rate of temperature rise in the past thirty years alone has shot up to an alarming 0.53°C per decade. Surface temperatures in Egypt and elsewhere in the Nile River Basin have also kept pace with this rise, increasing by an average of 0.16°C to 0.4°C, with El-Minya Governorate in Upper Egypt reaching temperatures of between 40°C and 44°C in recent years.

WATER SCARCITY:

Even without factoring in the effects of climate change, Egypt faces a dire water scarcity crisis. Since the late 1970s, Egypt's "total demand on water outstripped supply [from] the Nile," even though the country is already "fully utilizing the available resources of the Nile River." This deficit is especially alarming given that the Nile accounts for 98 percent of Egypt's renewable water supply.

➢ Our Solution

- **GreenWave** offers a unified platform for sustainable living, integrating waste management, urban greening, eco-friendly products, and sustainable transportation. Users gain access to educational resources, tools, and products, fostering informed choices and sustainable behaviors. Community engagement, robust security measures, and cohesive user experience further empower individuals to embrace eco-friendly lifestyles and contribute to environmental conservation efforts.

➢ Overview

GreenWave is a comprehensive platform designed to promote sustainable living and environmental conservation. It offers a variety of services aimed at educating and empowering individuals to adopt eco-friendly practices. Key features include:

- **User Registration and Authentication:** Secure account creation and login using email and password, with robust security measures to protect user data.
- **Waste Collection Service:** Schedule waste collection appointments, receive confirmation emails, and earn redeemable points.
- **Agricultural Education:** Access a rich library of categorized and searchable educational resources, including courses, articles, and books on agricultural skills.
- **Urban Greening Service:** Request assistance from experts for cultivating green spaces and receive guidance on gardening practices.
- **Eco-friendly Product Marketplace:** Purchase essential tools and supplies for sustainable living, with secure payment and order tracking.
- **Waste Reporting and Reward System:** Report garbage collection sites, earn points for community cleanliness efforts, and manage reported sites.
- **User Feedback and Support:** Provide feedback, report issues, and access support for queries and concerns.
- **Admin Dashboard:** Manage user accounts, educational resources, waste collection appointments, and reported sites, with analytics and reporting capabilities.
- **Machine Learning Chatbot:** Enhance user interaction with a chatbot powered by Python, for instant support and guidance.

GreenWave's mission is to educate and empower individuals on environmental conservation, fostering a community committed to sustainable living practices.

➢ Objective

The objective of the GreenWave project is to create a comprehensive and user-centric platform that empowers individuals to adopt sustainable practices, make informed decisions, and play an active role in environmental conservation. The project aims to provide users with easy access to a wide range of eco-friendly services and resources, including waste management solutions, agricultural education, urban greening initiatives, and an eco-friendly product marketplace.

Additionally, the project prioritizes robust security measures to protect user data and ensure a safe online environment. By integrating modern technologies such as React, Node.js, MongoDB, and machine learning chatbots, GreenWave strives to deliver a cohesive and engaging user experience that builds trust and confidence in the platform.

Ultimately, the goal is to promote environmental sustainability, encourage community participation, and enable users to lead eco-friendly lifestyles, contributing to a healthier planet for future generations.

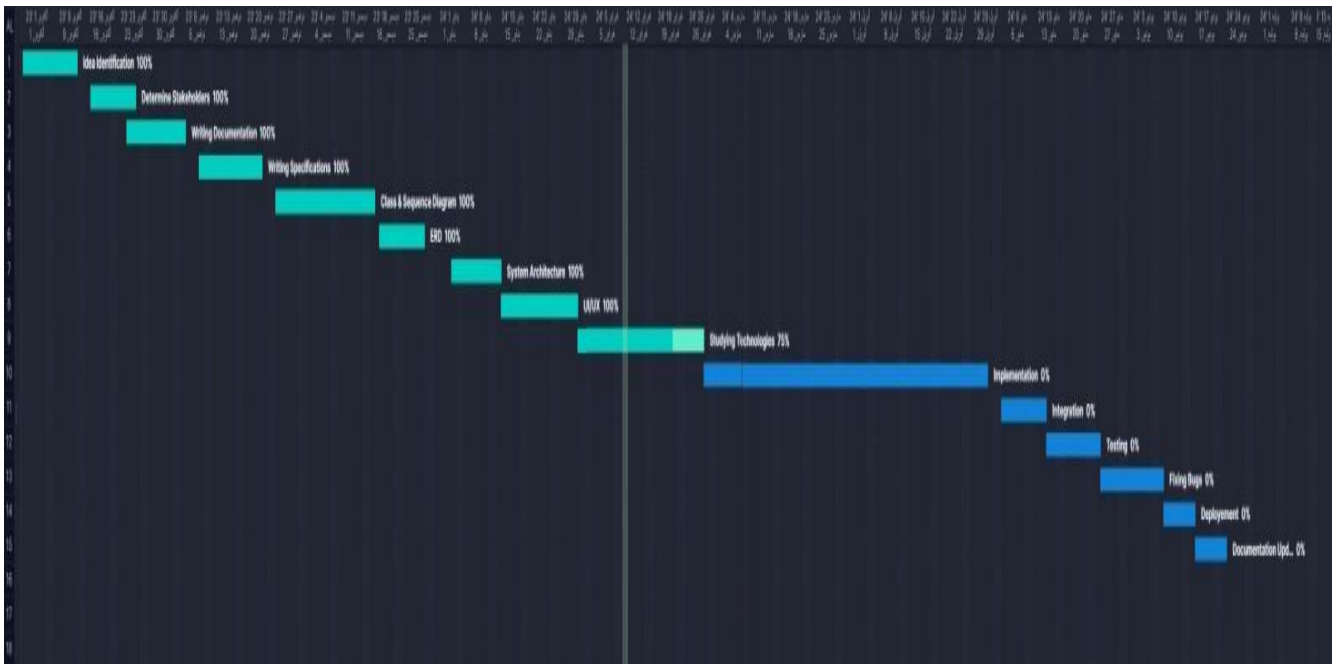
Gantt chart of project time plan:

Task	Task Title	Description	Task status (completed/expected in time)
1	Idea Identification	We made brainstorming to settle for the best ideas with the Doctor.	Start 2/10/2023 End 13/10/2023
2	Determine Stakeholders	Analyzing & studying the market for finding competitors that will have features like us & determine how can we attract & enrich our users.	Start 17 /10 / 2023 End 26/10/2023
3	Writing Documentation	We started in documenting background, problem definition & search for other related work.	Start 25/10 /2023 End 9/ 11 /2023
4	Writing Specifications	We worked on the project specification by determining functional, non-functional requirements & use case diagrams.	Start 20/11/2023 End 11/ 12 /2023
5	Class & Sequence Diagram	We continued in working on project specification by designing class & sequence diagrams.	Start 20/11/2023 End 11/12 / 2023
6	ERD	We designed the ERD	Start 12 / 12 /2023 End 21/12 / 2023
7	System Architecture	We designed our System Architecture.	Start 22/12/2023 End 1/1/2024
8	UI/UX	We designed our UI/UX	Start 16/1/2024 End 1/2/2024
9	Studying Technologies	Learning some new technologies (Nodejs & React)	Start 1/2/2024 End 29/2/2024
10	Implementation	We wrote the code of the app.	Start 1/3/2024. End 2/5/2024
11	Integration	We integrate the Front-End with Back-end (& if there is any Machine Learning Model).	Start 6/5/2024. End 15 / 5 / 2024
12	Testing	We wrote test cases to the whole code.	Start 16/5/2024. End 27/5/2024
13	Fixing Bugs	Fixing all the bugs found during testing & re-implementing.	Start 28/5/2024. End 10/6/2024
14	Deployment	Final Website deployment	Start 11/6/2024. End 1/7/2024
15	Documentation Update	Updating Documentation to a new architecture.	Start 18/6/2024. End 3 /7/ 2024

Gantt chart:

Project Gantt chart is divided into two parts to be more clear and these two parts are:

(analysis and implementation).



Project Development Methodology

Waterfall

Waterfall The Waterfall methodology is a linear and sequential approach to software development. It is characterized by a series of distinct phases that are followed in a strict order, with each phase building upon the results of the previous one. The phases typically include requirements gathering and analysis, system design, implementation, testing, deployment, and maintenance.

Requirements In the requirements phase of the software development process, the main goal is to gather a thorough understanding of the user's needs and define the specifications of the software product. This phase is crucial as it lays the foundation for the entire development process.

Design In the design phase of software development, the goal is to translate the requirements gathered in the previous phase into a detailed design that will guide the implementation of the software. The design phase focuses on defining the structure, architecture, and components of the final solution.

Implementation In the implementation phase of software development, the focus is on translating the design specifications into actual working code. This is where the development team takes the detailed design and builds the software system.

Testing Is a critical phase in the software development process, aimed at identifying and fixing any defects or issues in the system before it is deployed to the users. A test plan will be followed. Different test cases will be executed according to three types: unit testing, integration testing, and usability testing.

Deployment and Maintenance are two essential stages in the software development process. Deployment involves releasing the developed software system and making it available for use by the intended users. It includes activities such as configuring the system, setting up servers, and ensuring a smooth transition to the production environment. On the other hand, Maintenance focuses on ensuring the ongoing operation and performance of the software system. It involves monitoring, bug fixing, applying updates, and providing technical support to users.

➢ Our Report Organization

The Next Chapters Will Discuss **GreenWave** in More Details

Chapter 2

- Some Similar Solutions to the Same Problem and the Difference from our Project

Chapter 3

- Details of the Requirements in the System and Use Cases

Chapter 4

More About the Design of **GreenWave** in Details:

- System Architecture Design
- System Classes UML
- Sequence Diagrams
- System ERD
- System GUI

Chapter 5

- Implementation and Testing of the Application

Chapter 2: Related Work

Our website offers a comprehensive suite of services, including a Marketplace for Eco-Friendly products, Agricultural and Greening needs, Waste Collection Service, Educational Resources (books, courses, articles), and Gardening Services. Below are similar implementations in Egypt and a comparison of their offerings versus ours.

Similar Implementations to the Idea

1. **Mashtal**
    - **Description:** An e-commerce platform specializing in indoor plants, gardening supplies, and plant care information. Established in 2017, Mashtal provides a wide range of gardening products and articles to enhance user knowledge.
    - **Services:**
        - E-commerce for indoor plants and gardening supplies.
        - Informative articles on plant care and gardening.
2. **GrowPro**
    - **Description:** An e-commerce website focused on urban gardening solutions, including home systems, organic edible seedlings, and gardening supplies. Established in 2020, GrowPro also provides articles to enrich users' knowledge.
    - **Services:**
        - E-commerce for urban gardening solutions and supplies.
        - Informative articles on gardening and plant care.
3. **PlantnMore**
    - **Description:** PlantnMore is an online store that provides a variety of plants, gardening tools, and accessories. They also offer indoor and outdoor plant solutions tailored to customer needs.
    - **Services:**
        - E-commerce for plants and gardening supplies.
        - Tips and advice for plant care and gardening.
4. **Shagara**
    - **Description:** Shagara is an initiative focused on promoting urban gardening and tree planting. They provide tools and resources for individuals and communities to engage in green projects.
    - **Services:**
        - Tree planting initiatives.
        - Community gardening projects.
        - Workshops and educational resources on gardening and environmental conservation.
5. **Urban Greens Egypt**
    - **Description:** Urban Greens Egypt specializes in hydroponic and vertical farming solutions for urban spaces. They provide equipment and support for individuals and businesses looking to start their own urban farms.
    - **Services:**
        - Hydroponic and vertical farming systems.
        - Workshops and training on urban farming techniques.
        - Consulting services for setting up urban farms.

The Main Differences Between Them and Us

While these platforms offer valuable services related to indoor plants, urban gardening, and innovative farming techniques, our platform differentiates itself by providing a broader range of services and features:

1. **Educational Resources:**
   - **Courses and Books:** Unlike Mashtal, GrowPro, PlantnMore, Shagara, and Urban Greens Egypt, our platform includes a comprehensive library of educational resources, offering courses and books on various agricultural and greening topics.
2. **Waste Collection Service:**
   - **Scheduled Appointments:** We offer a waste collection service where users can schedule appointments for waste pickup, promoting responsible waste management.
3. **Reward System:**
   - **Points Redemption:** Our platform incorporates a reward system where users earn points for engaging in eco-friendly activities. These points can be redeemed for products in our marketplace.
4. **Eco-Friendly Products:**
   - **Sustainable Daily Use Products:** Our marketplace includes a variety of eco-friendly products for daily use, promoting sustainable living beyond just gardening supplies.
5. **Urban Greening Assistance:**
   - **Delegate Support:** We provide assistance from delegates for urban greening projects, offering guidance and support for cultivating green spaces in urban environments.
6. **Waste Reporting Feature:**
   - **Reporting Sites Needing Attention:** Users can report sites that require waste collection, contributing to community cleanliness and environmental conservation.

Chapter 3: System Analysis

Functional Requirements

1. Basic Requirements

1.1. User Registration and Authentication

1.1.1. The user should be able to register to the system with their email and password.
1.1.2. The user should log in to the system if they are already registered.
1.1.3. *The password is hashed before storing it in the database to ensure security.*
1.1.4. User authentication should be secure and protect user data.
1.1.5. *The phone number provided by the user should be encrypted.*
1.1.6. *The user must confirm their email through a verification link sent to their email before gaining access to the website.*
1.1.7. *The user should be able to log in to the system if they are already registered and have confirmed their email.*
1.1.8. *All sensitive data, such as passwords and phone numbers, should be appropriately hashed and encrypted.*
1.1.9. *Upon updating the password, ensure the data remains consistent and follows the same hashing and encryption protocols*

1.2. User Main Operations

1.2.1. The user should be able to schedule waste collection appointments.
1.2.2. The platform should provide a list of acceptable waste types for collection.
1.2.3. The user should receive confirmation mail for their scheduled appointments.
1.2.4. The user should be able to earn points for scheduling waste collection appointments.
1.2.5. The user should be able to redeem earned points for products from the marketplace.
1.2.6. The user should be able to buy products such as books and courses from the Market place.
1.2.7. The user should be able to access articles for free.
1.2.8. The user can access Some books and courses available for free access.
1.2.9. The user should be able to order products and complete payment operations securely.
1.2.10. The user should be able to add products to their cart, review their order, and proceed to checkout.
1.2.11. The user should be able to apply coupons for discounts on their purchases according to his Points.

1.3. Agricultural Education

1.3.1. The platform should provide a library of educational resources such as courses, articles, and books.
1.3.2. The educational content should be categorized and easily searchable.

1.4. Urban Greening Service

1.4.1. The user should be able to request assistance from delegates for cultivating green spaces.
1.4.2. The user should receive guidance and support from delegates on gardening and greening practices.

1.5. Eco-friendly Product Marketplace

1.5.1. The platform should have an online store where users can purchase essential tools and supplies for sustainable practices.
1.5.2. Products should be categorized and searchable.
1.5.3. The user should be able to add products to a shopping cart.
1.5.4. The user should be able to make payments and track their orders.

1.6. Waste Reporting and Reward System

1.6.1. The user should be able to report garbage collection sites that require attention.
1.6.2. The platform should track and manage reported sites.
1.6.3. The user should earn points for reporting sites.

1.7. User Feedback and Review

1.7.1. Users should be able to access a support page with FAQs, contact information, and a support request form.
1.7.2. Users should be able to leave reviews on products, services, and educational resources available on the platform.

2. Admin Dashboard

2.1. Admins should be able to add books, courses, articles, waste collection appointments, and reported sites.
2.2. Admins should be able to Update books, courses, articles.
2.3. Admins should be able to Delete books, courses, articles.

3. Notifications

3.1. The user must be notified of their scheduled waste collection appointments by sending mail.
3.2. The user must receive notifications about their earned points and redeemed rewards.

Non – Functional Requirements

1- Security:

1.1. The system will be authenticated, ensuring that only registered users can access their accounts.

1.2. User accounts will be secured using a unique email address and a strong password with at least 6 characters.

1.3. Passwords will be hashed and stored securely in the database to protect user credentials.

1.4. Regular security audits and updates will be conducted to protect against vulnerabilities and ensure the security of user data.

2- Availability:

2.1. The system will be available 24/7, ensuring continuous access for users.

2.2. The website will achieve an availability target of 97%, minimizing downtime and ensuring reliability for users.

3- Usability:

3.1. The system will provide a flexible and intuitive user interface, making it easy for users to navigate and perform tasks.

3.2. Comprehensive help documentation and user guides will be available to assist users in using the platform effectively.

3.3. The platform will be responsive, ensuring compatibility with various devices such as desktops, tablets, and smartphones.

4- Performance:

4.1. The system will be responsive and load quickly to minimize waiting time for users.

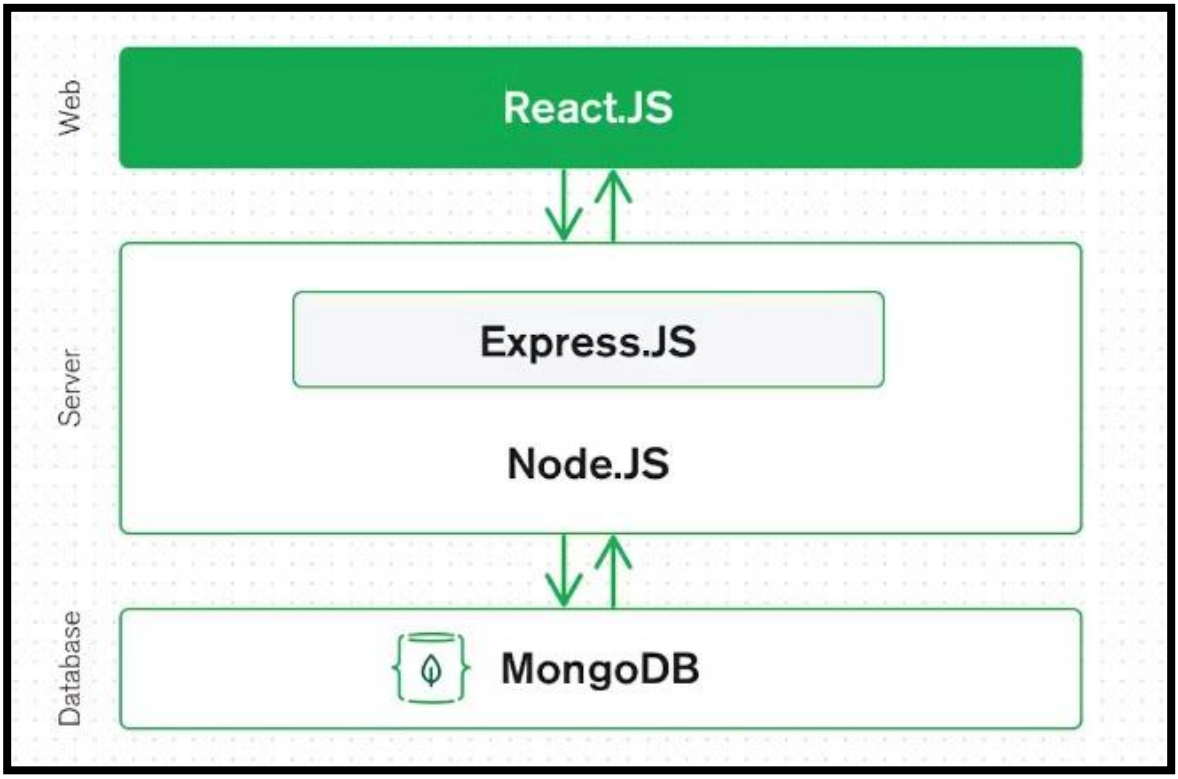
4.2. The platform will ensure that all user requests are processed within 5 seconds under normal load conditions.

4.3. The system will be optimized for performance to handle high traffic and large numbers of concurrent users without degradation.

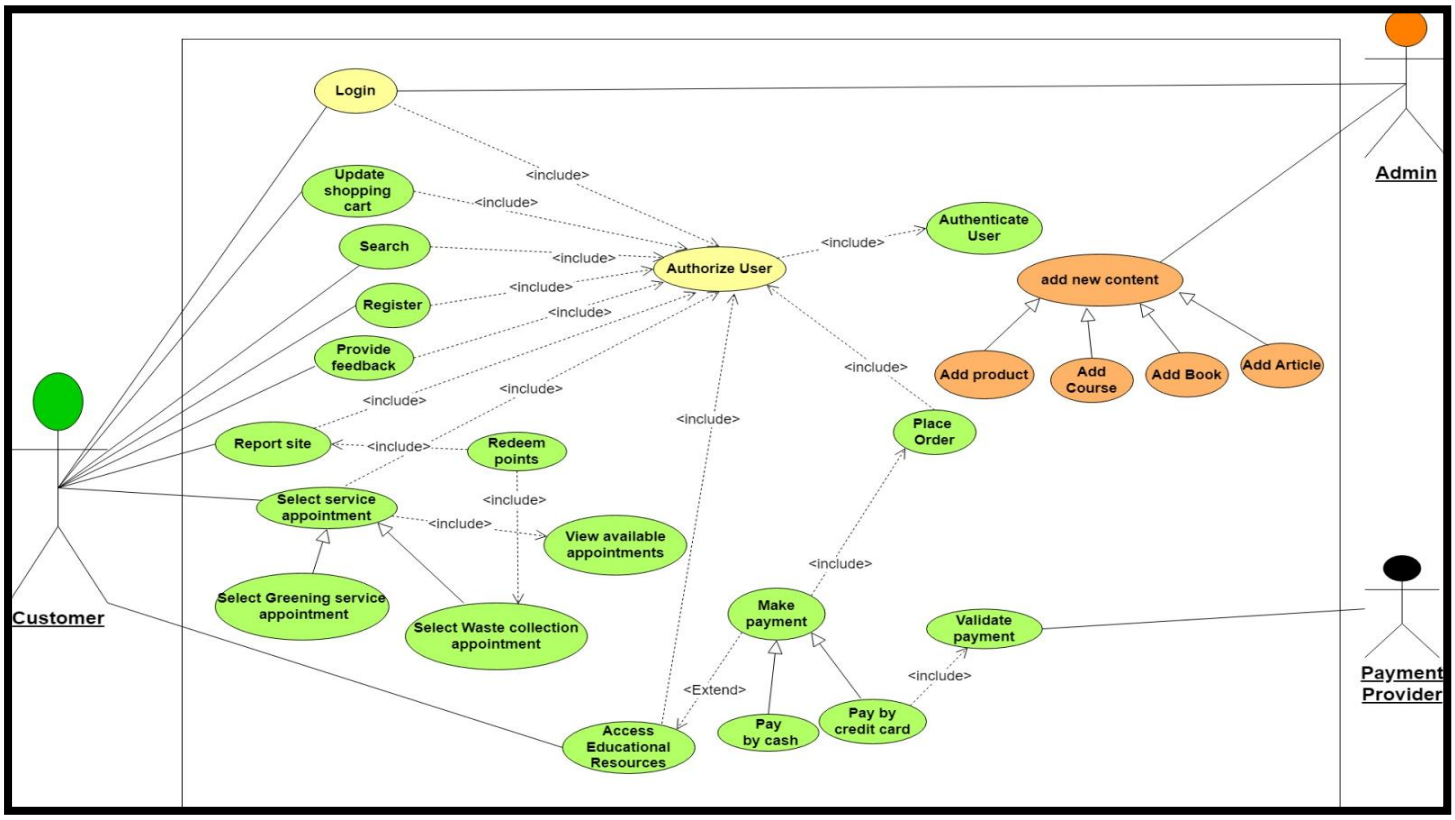
Stakeholders:

- **Users**: Individuals who will utilize the platform to access educational resources, tools, and products for sustainable living, schedule waste collection appointments, engage in urban greening activities.
- **Educational Content Providers**: Organizations or individuals contributing educational resources on agriculture for the platform.
- **Environmental Organizations**: Non-profit organizations, environmental groups, and sustainability advocates who may collaborate with the project to promote eco-friendly practices and environmental conservation.
- **Regulatory Authorities**: Government agencies and regulatory bodies responsible for overseeing environmental regulations, data privacy, and security standards that may impact the platform's operations.
- **Investors and Sponsors**: Individuals or organizations providing financial support, resources, or expertise to facilitate development.
- **Product Suppliers**: Suppliers and vendors of eco-friendly products listed on the platform's marketplace.
- **Shipping Institution**: Organization or service responsible for handling the shipping and delivery of eco-friendly products purchased on the platform.
- **External Partners:** Companies facilitating secure online payments for purchases made in the eco-friendly product marketplace.

System Architecture



Use Case Diagram



Use Case Tables:

Table 1: Use Case 1 --> Sign Up

Use case name	Sign up
Actors	user
Description	This use case describes the process of a user signing up for an account on the platform.
Pre-condition	1.The user accesses the platform's registration page. 2.The user has a valid email address. 3.User is not registered on the website with the same email
Post-condition	1.The user has a registered account. 2.The user receives a confirmation email.
Flow of events	1.User enters registration information: Full Name, Email, Password, and Confirm Password. 2.System validates the entered information for correctness and completeness. 3.User clicks on the "Sign Up" button. 4.System verifies the uniqueness of the email address. 5.If the email address is unique, the system creates a new user account and sends a confirmation email. 6.System displays a success message and prompts the user to check their email for confirmation. 7.User receives the confirmation email and clicks on the provided link. 8.System verifies the confirmation link and activates the user's account. 9.System displays a confirmation message, and the user is now registered. User can log in using their credentials.
Extension	1.If the entered email address is already associated with an existing account, the system displays an error message and prompts the user to choose a different email address. 2.If the user does not confirm within a specified timeframe or the confirmation link is invalid, the system prompts the user
Alternative path	If the user decides not to complete the registration, they can navigate away from the registration page.

Table 2: Use Case 2 --> Login.

Use case name	login
Actors	user
Description	This use case describes the process of a user logging into their account on the platform.
Pre-condition	The user has a registered account on the platform.
Post-condition	1.The user is successfully logged into their account. 2.The user gains access to personalized features and information.
Flow of events	1-User navigates to the platform's login page. 2-System displays the login form. 3-User enters their registered email address and password. 4-System validates the entered credentials. 5-User clicks on the "Login" button. 6-System verifies the email and password combination. 7-If the credentials are valid, the system logs the user in and redirects them to the user dashboard. 8-User gains access to personalized features. 9-System displays a welcome message and relevant user information. 10-User successfully logged in.
Extension	If the entered email address or password is invalid, the system displays an error message and prompts the user to enter valid credentials.
Alternative Path	If the user forgot their password, they can click on the "Forgot Password" link, triggering a password reset process.

Table 3: Use Case 3 --> Request a waste collection.

Use case name	Request waste collection
actor	user
Description	This use case describes the process of a user requesting a waste collection appointment, specifying waste details, and ensuring the weight meets the minimum requirement.
Pre-condition	1. The user is logged into their account on the platform. 2.The user has navigated to the waste collection section.
Post-condition	1.The user successfully schedules a waste collection appointment (if available). 2.The system updates the user's appointments.
Flow of events	1.User navigates to the waste collection section and views the list of all available appointments. 2.System displays a list of all upcoming waste collection appointments. 3.User checks the list of available appointments. 4.System registers the user's intention to request an appointment. 5.If there are available appointments, the user selects a preferred appointment from the list. 6.System registers the user's selection. 7.User specifies waste details: waste type, weight, location, and uploads a photo. 8.System captures and validates the provided information. 9.System checks if the weight of the waste is more than the minimum required. 10.If weight meets the minimum requirement, proceed; otherwise, inform the user. 11.User clicks on the "Request Appointment" button. 12.System schedules the selected appointment for the user. 13.System sends a confirmation email to the user with details of the scheduled appointment. 14.User receives an email confirming the requested waste collection appointment.
Extension	If the entered email address or password is invalid, the system displays an error message and prompts the user to enter valid credentials
Alternative path	If the user forgot their password, they could click on the "Forgot Password" link, triggering a password reset process.

Table 4: Use Case 4 --> --> Course Enrollment.

Use case name	course enrollment
actor	user
description	This use case describes the process of a user enrolling in a course on the platform and making the enrollment payment using a credit card.
Pre-condition	1.The user successfully enrolls in the selected course. 2.The system processes the credit card payment and updates the user's enrollment status.
Post-condition	1.The user successfully enrolls in the selected course. 2.The system processes the credit card payment and updates the user's enrollment status.
Flow of events	1.User navigates to the course enrollment section and views the list of available courses. 2.System displays a list of courses with enrollment details. 3.User selects a preferred course from the list. 4.System displays detailed information about the selected course. 5.User clicks on the "Enroll Now" button. 6.System prompts the user to confirm their enrollment and proceed to payment. 7.User confirms enrollment and chooses the payment method as credit card. 8.System prompts the user to enter credit card details. 9.User enters credit card details (card number, expiration date, CVV). 10.System validates the entered credit card information. 11.System processes the credit card payment for the course enrollment. 12.System confirms the user's enrollment in the course. 13.User receives a confirmation message and is officially enrolled in the selected course.
Extensions	If the credit card payment fails, the system informs the user about the payment failure and prompts them to retry or use an alternative payment method.
Alternative path	If the user has previously enrolled in courses, they can view their enrollment history before considering a new enrollment

Table 5: Use Case 5 --> Request gardening service.

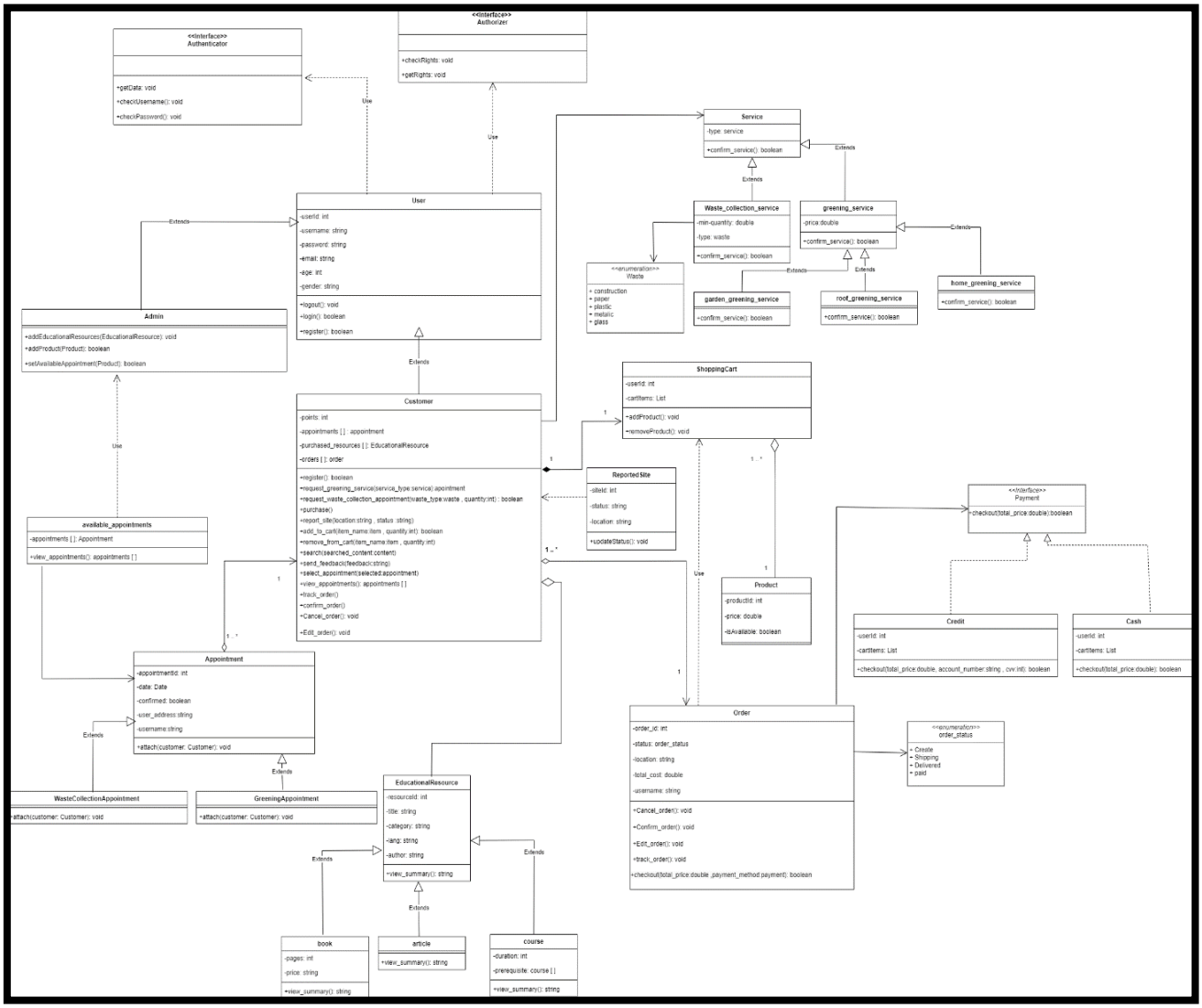
Use case name	Request gardening service
actor	user
Description	This use case describes the process of a user requesting a gardening service on the platform by selecting an appointment, choosing a specific gardening service, and filling out the request form.
Precondition	1.The user is logged into their account on the platform. 2.The user has navigated to the gardening services section.
Postcondition	1.The user successfully requests a gardening service. 2.The system updates the user's appointments.
Flow of events	1.User navigates to the gardening services section and views the list of available appointments. 2.System displays a list of upcoming gardening service appointments with details. 3.User selects a preferred appointment from the list. 4.System registers the user's intention to request a gardening service. 5.User clicks on the "Request Service" button. 6.System presents a list of available gardening services for the selected appointment. 7.User selects a specific gardening service from the list. 8.System registers the user's choice of gardening service. 9.User fills out the gardening service request form (providing additional details such as garden size, specific requirements, etc.). System validates the entered information and prompts the user to review their request. 10.User reviews the request details and clicks on the "Submit Request" button. 11.System schedules the selected appointment for the user. 12.System sends a confirmation email to the user with details of the scheduled appointment. 13.User receives an email confirming the requested Gardening appointment.
Extension	If there are errors or missing information in the request form, the system informs the user and prompts them to correct the issues before submission

Table 6: Use Case 6--> place order (cash om delivery).

Use case name	Place order
Actors	user
Description	This use case describes the process of a user shopping on the platform's marketplace, adding items to the shopping cart, and placing an order.
Pre-condition	1.The user is logged into their account on the platform. 2.The user has navigated to the marketplace section.
Post-condition	1.The user successfully places an order for the selected items. 2.The system processes the order, updates inventory, and records the user's purchase history.
Flow of events	1.User navigates to the marketplace section and views the list of available products. 2.System displays a list of products with prices. 3.User clicks on the "Add to Cart" button. System adds the selected product to the user's shopping cart. 4.User continues shopping and repeats steps 3 for additional products. 5.System updates the shopping cart with the selected products. 6.User clicks on the "Shopping Cart" button to review the selected items. 7.System displays the contents of the user's shopping cart with a total order summary. 8.User clicks on the "Proceed to Checkout" button. 9.System prompts the user to confirm the order details and select a shipping method. 10.User selects the "Cash on Delivery" payment option and clicks on the "Place Order" button. 11.System processes the order, updates inventory, and records the user's purchase history. 12.System sends a confirmation email to the user with order details. 13.User receives an email confirming the placed order with the "Cash on Delivery" payment method.
Extension	1.If the selected product is unavailable, the system informs the user and prompts them to choose an alternative product. 2.If there is an error during the order processing, the system informs the user and prompts them to retry or contact customer support.

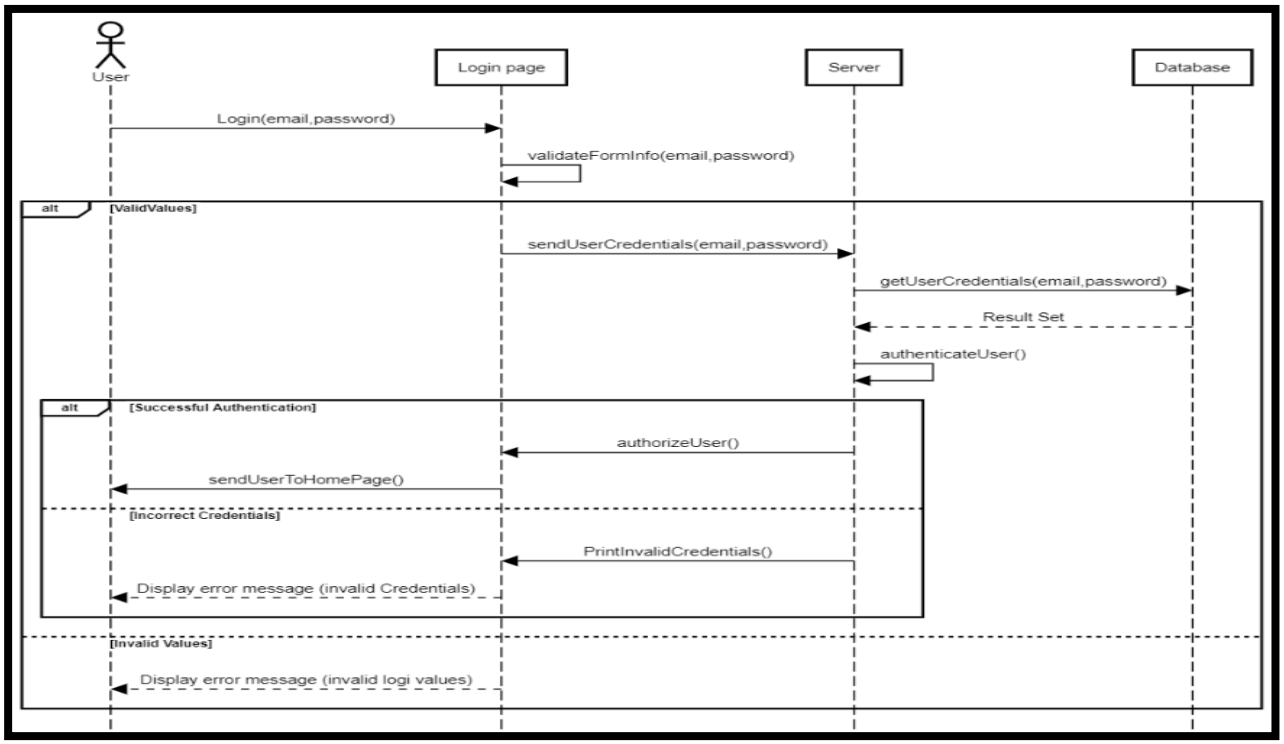
Chapter 4: System Design

Class Diagram

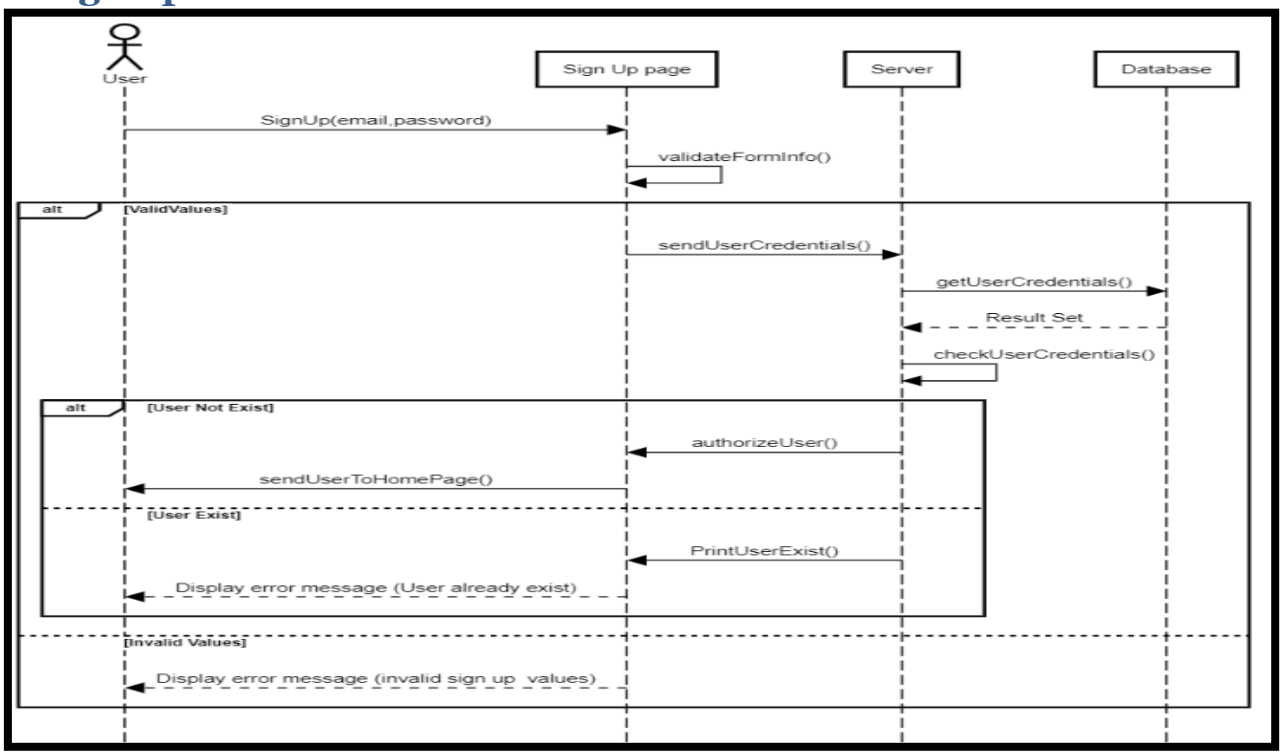


Sequence Diagrams

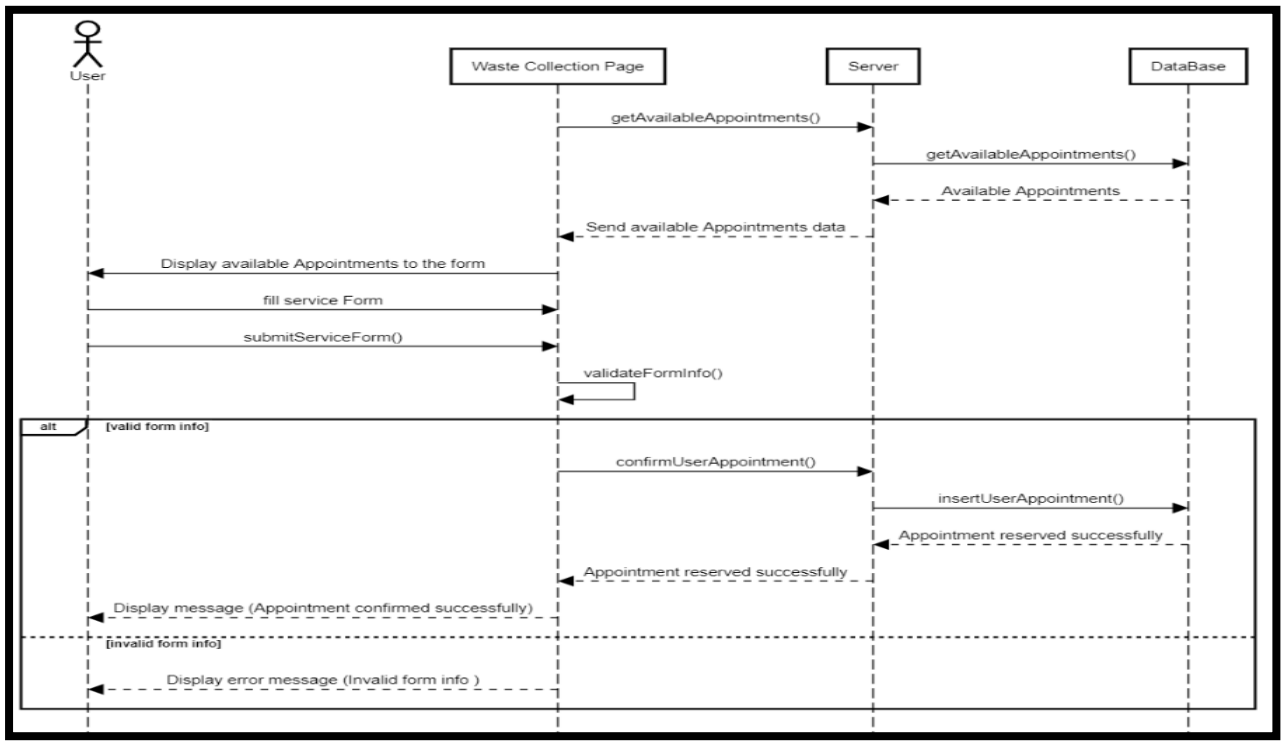
1-Login



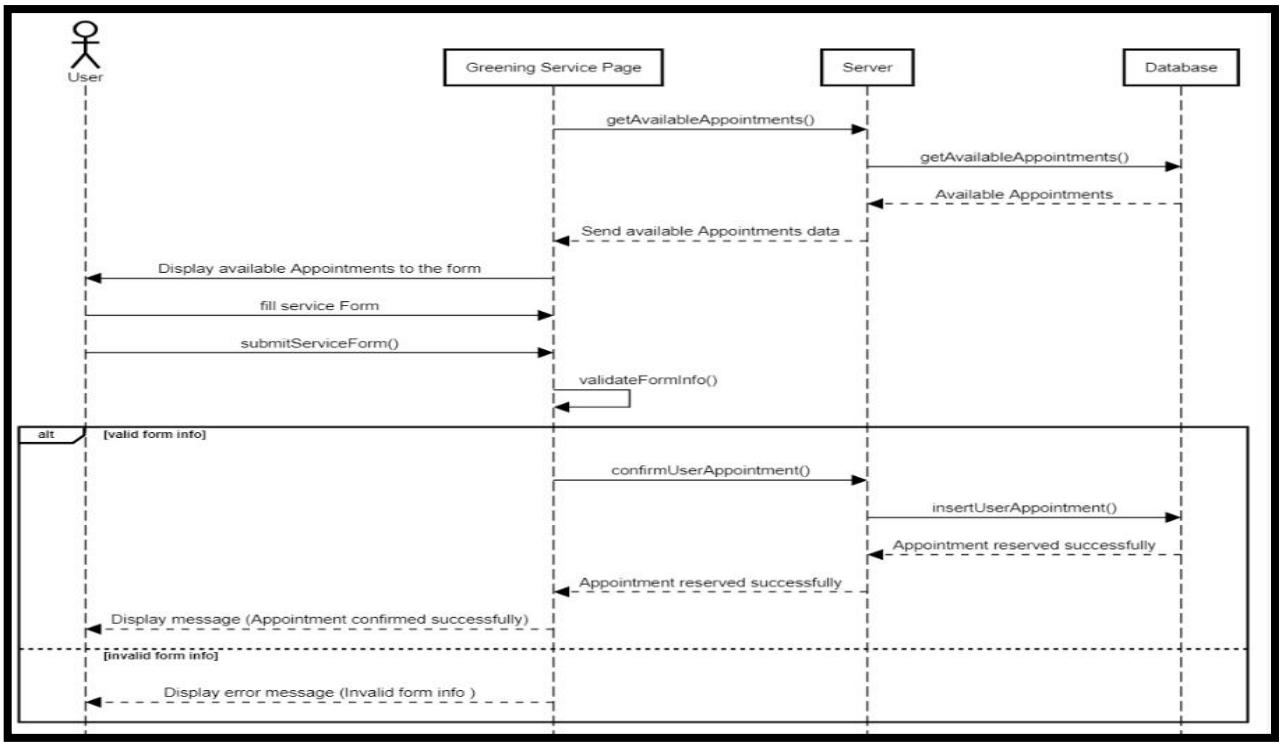
2-Sign up



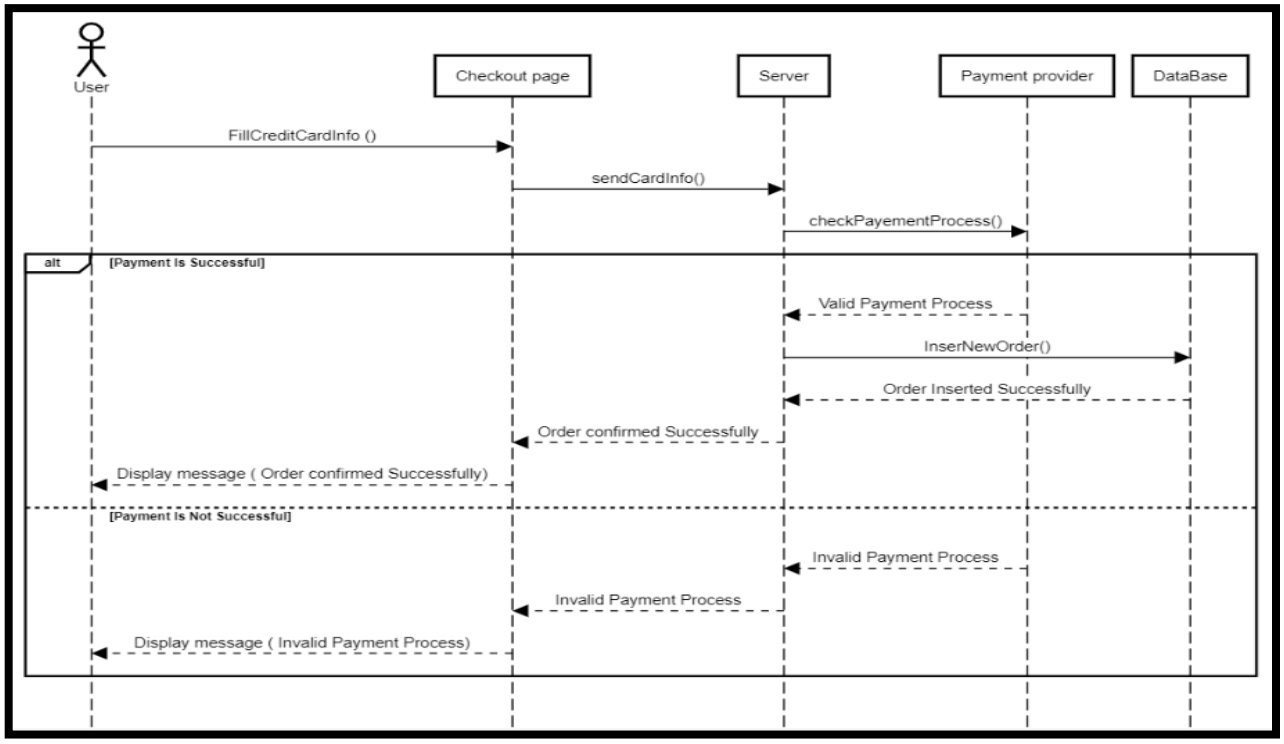
3- Waste Collection Service



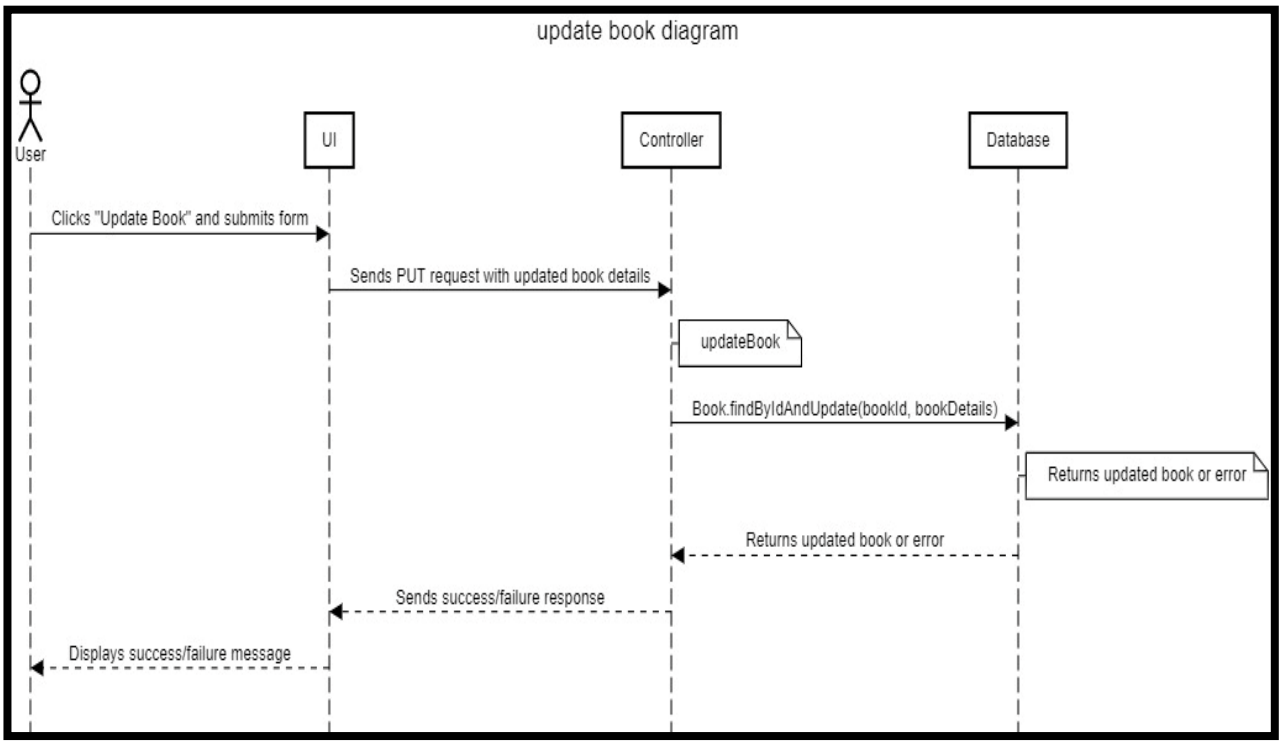
4-Greening Service



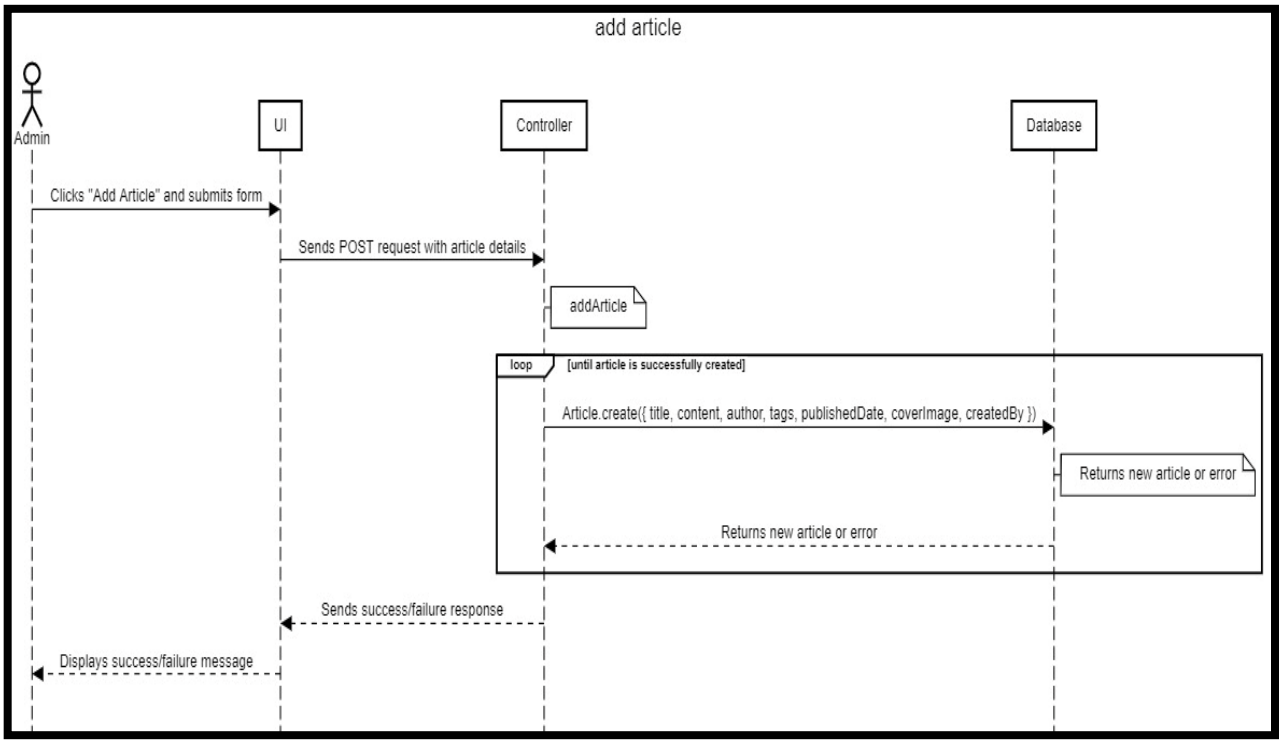
5- Checkout Service



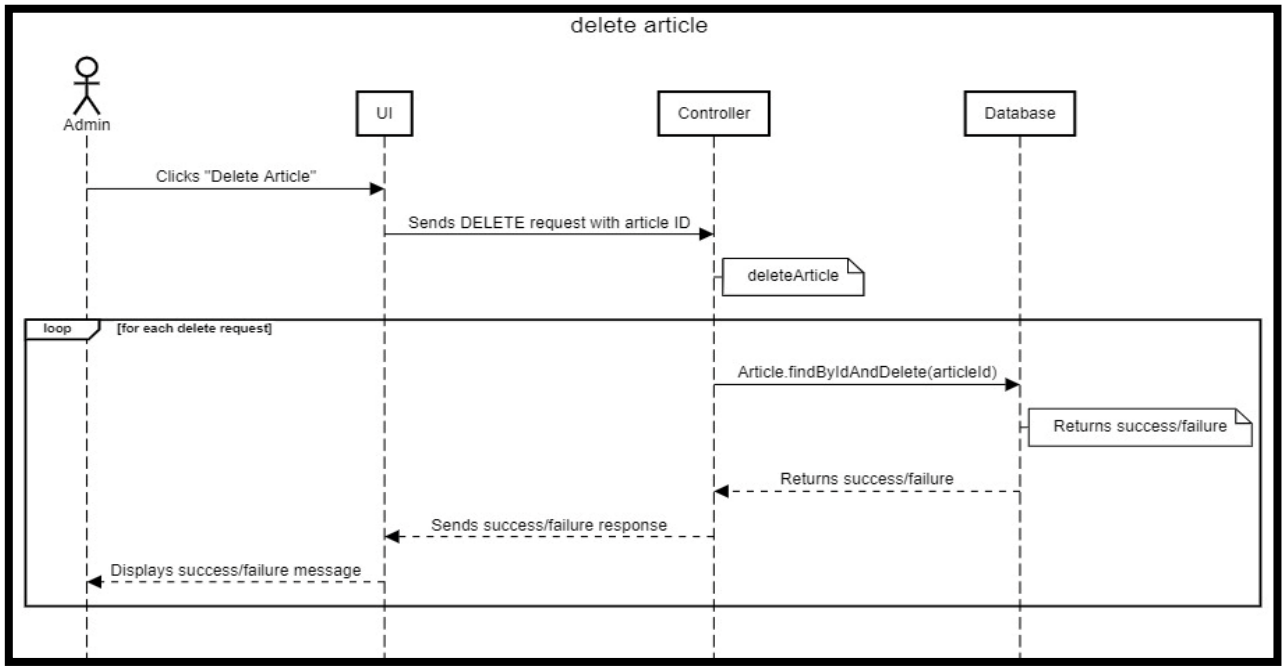
Update Book :



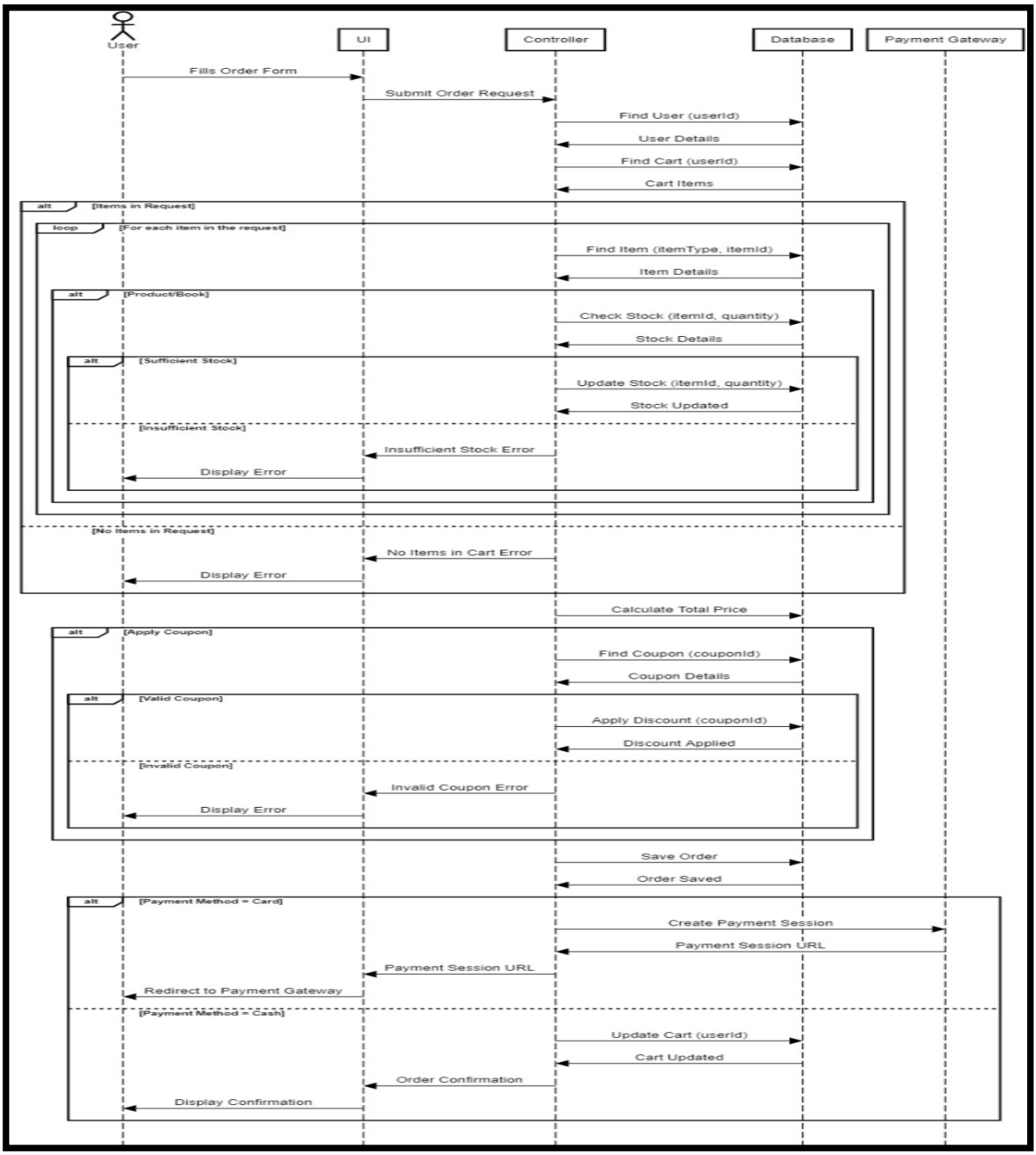
Add Article:



Delete Article:

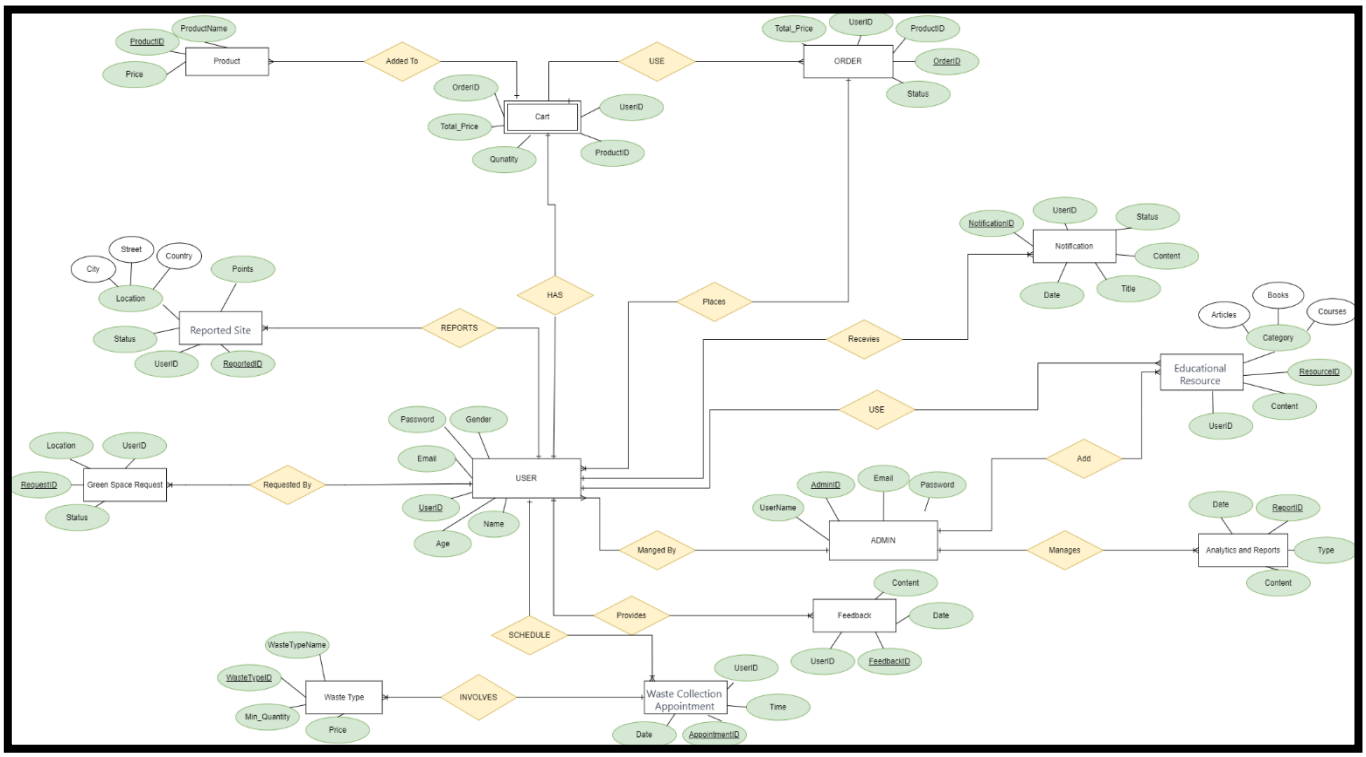


Payment and Order:

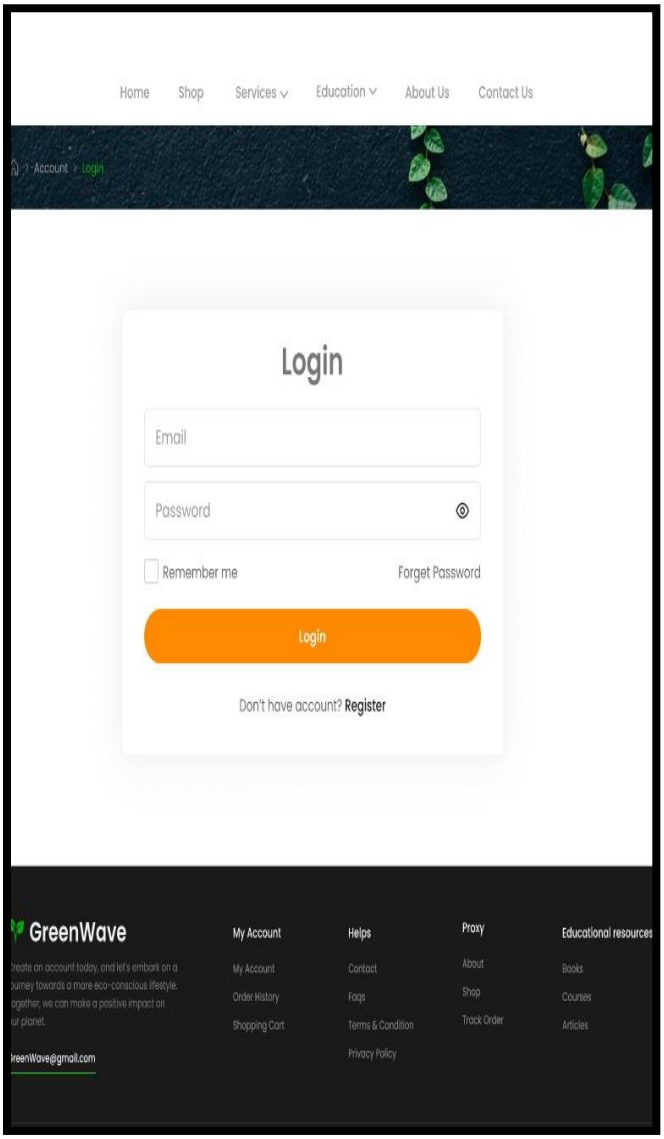


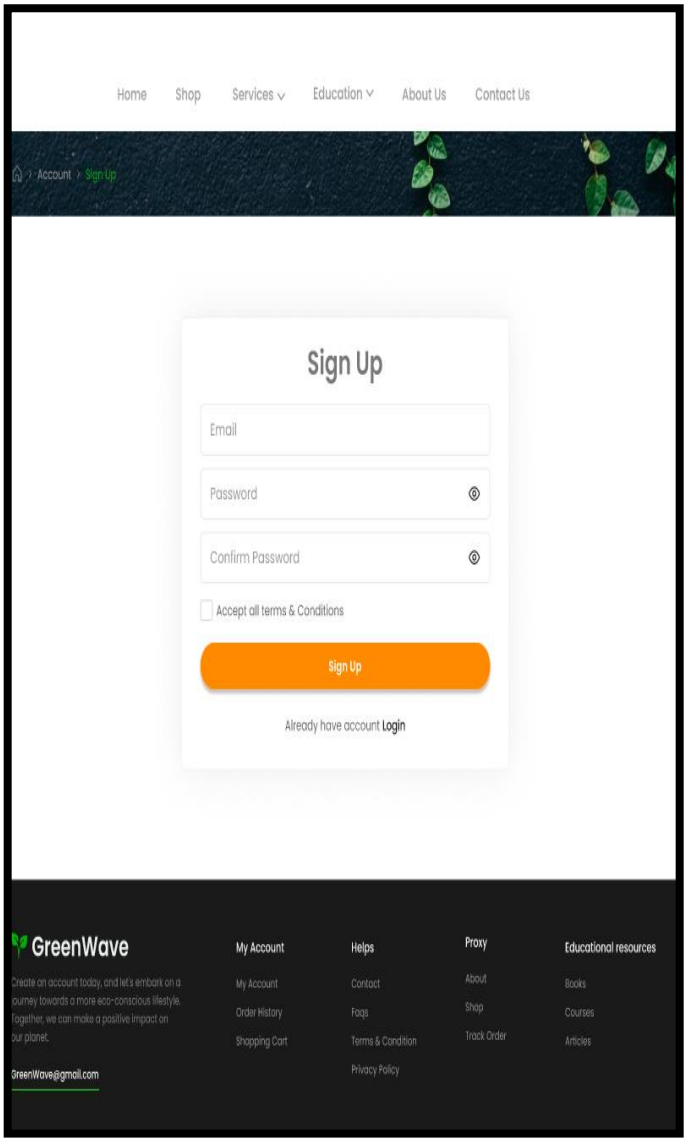
Entity Relationship Diagram (ERD):

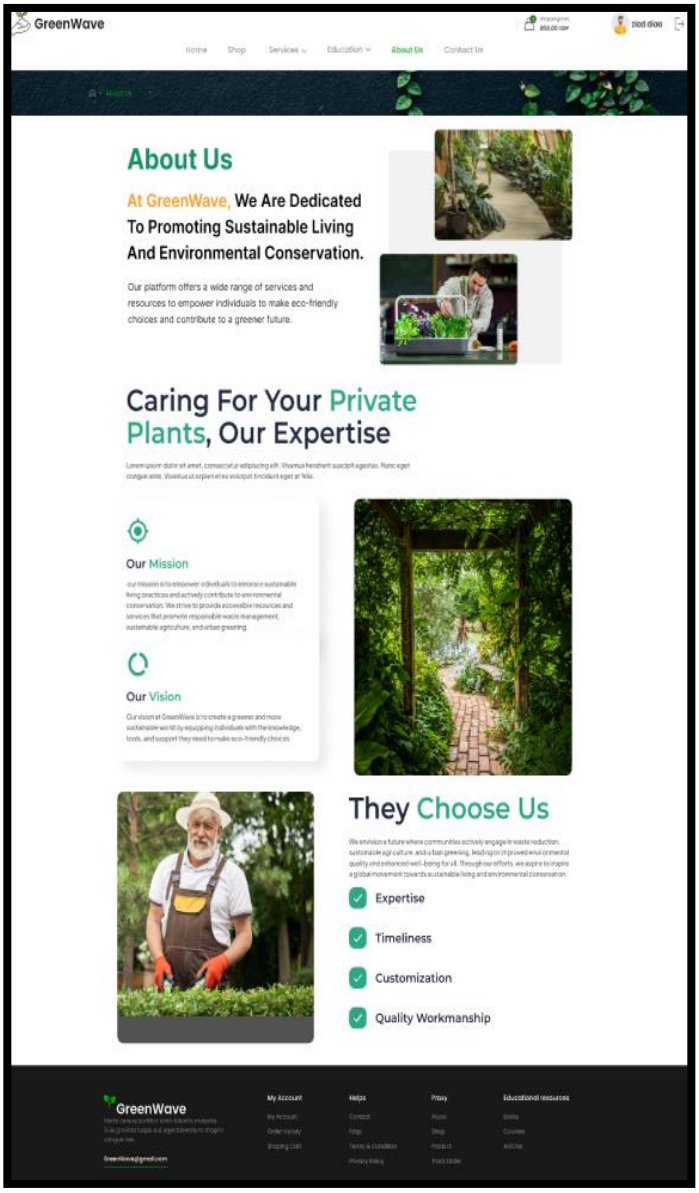
- The following Diagram Represents the Relation in the Database

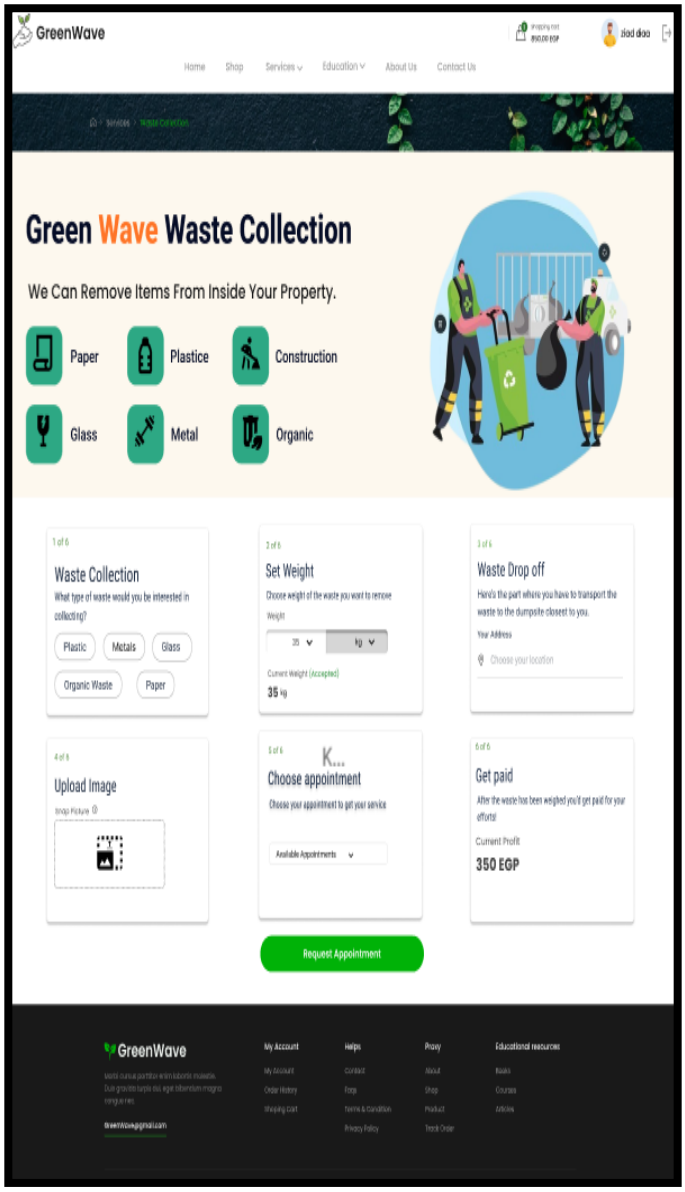


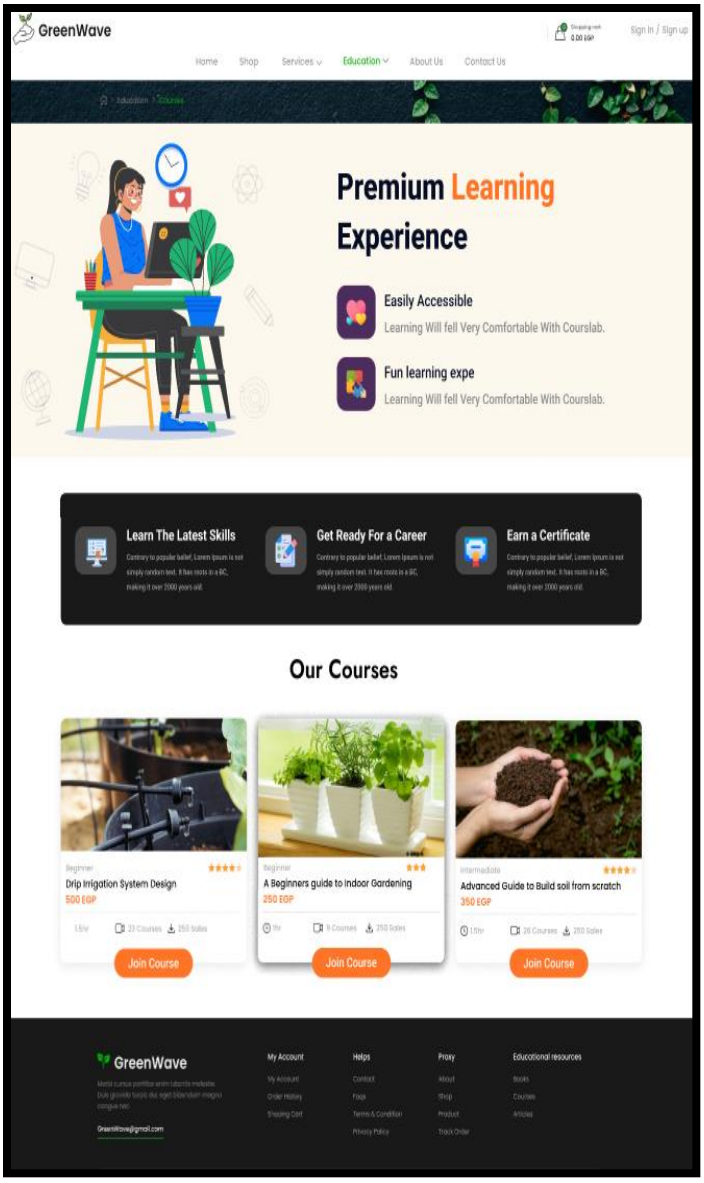
Chapter 6: Figma Snippet

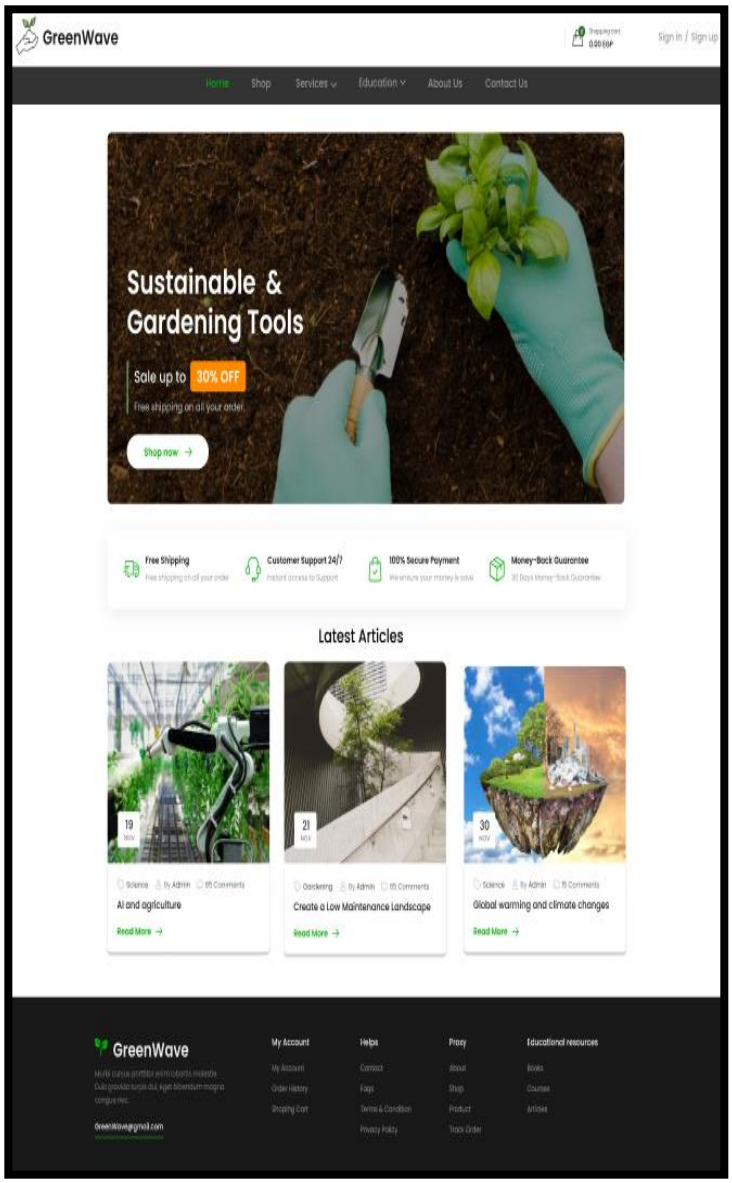


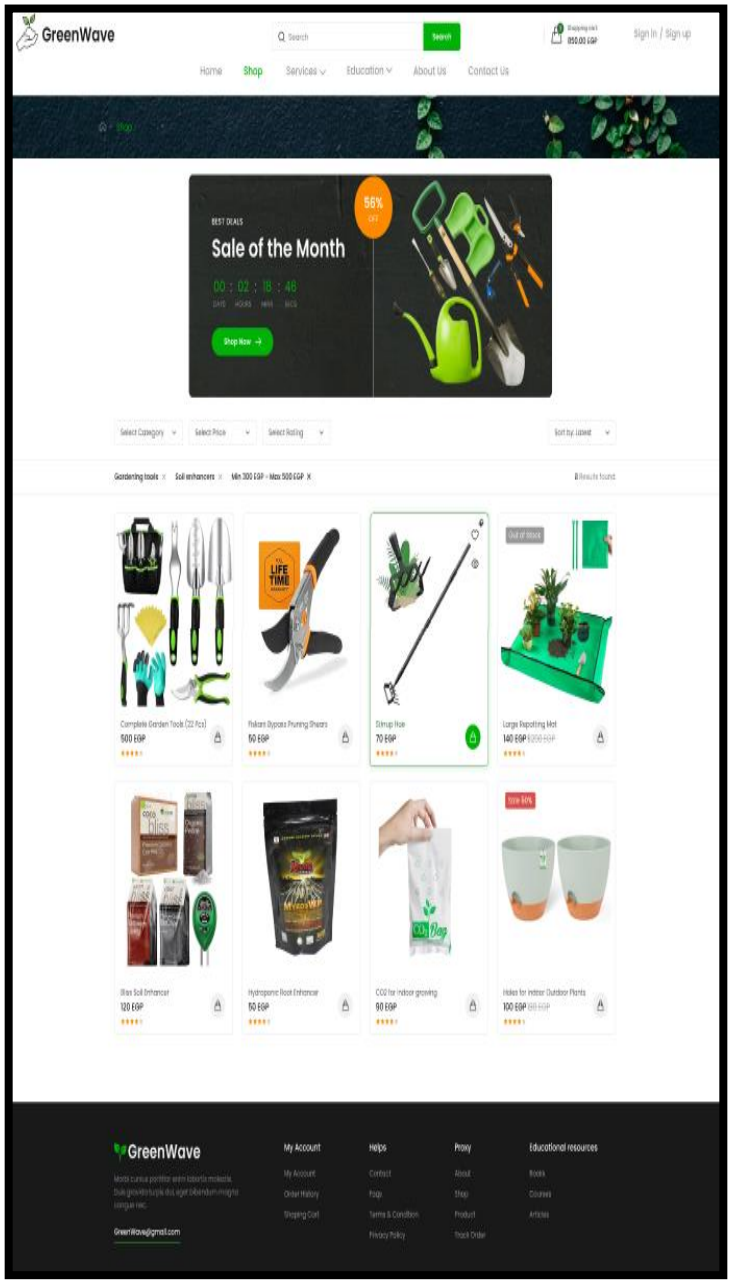


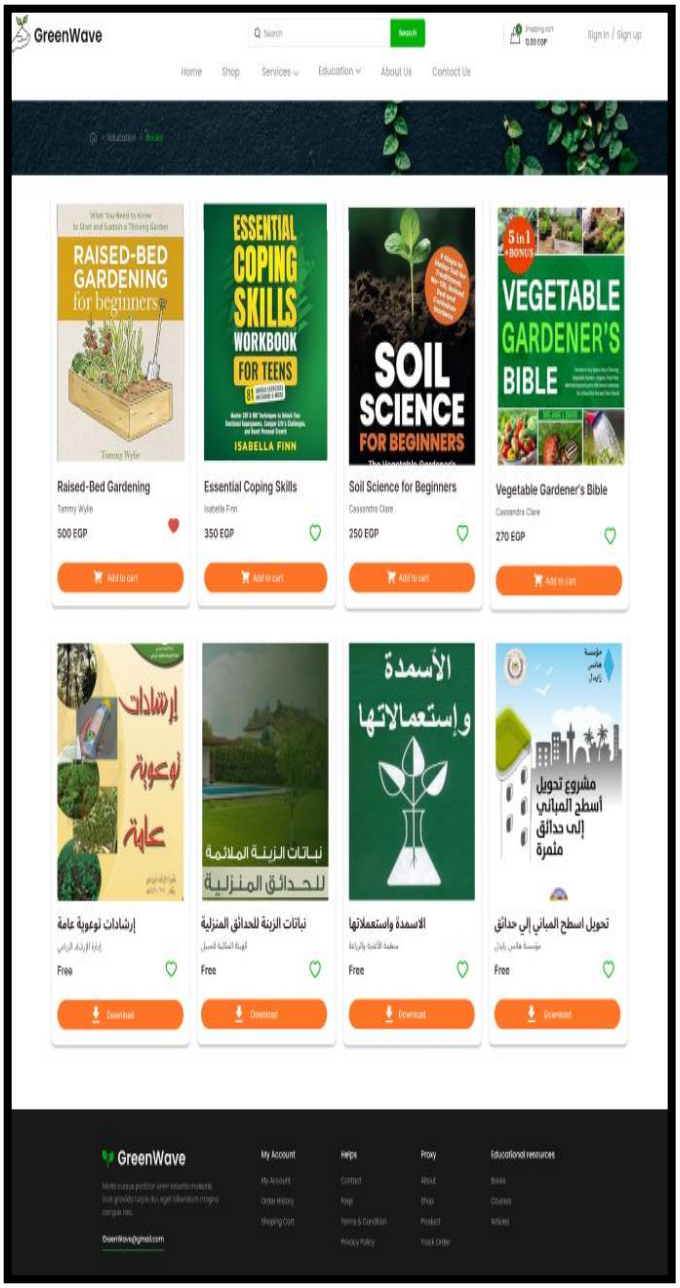

GreenWave
Home
Shop
Services
Education
About Us
Contact Us
About Us
At GreenWave, We Are Dedicated To Promoting Sustainable Living And Environmental Conservation.
Our platform offers a wide range of services and resources to empower individuals to make eco-friendly choices and contribute to a greener future.
Caring For Your Private Plants, Our Expertise
Our Mission
Our Vision
They Choose Us
Expertise
Timeliness
Customization
Quality Workmanship
GreenWave

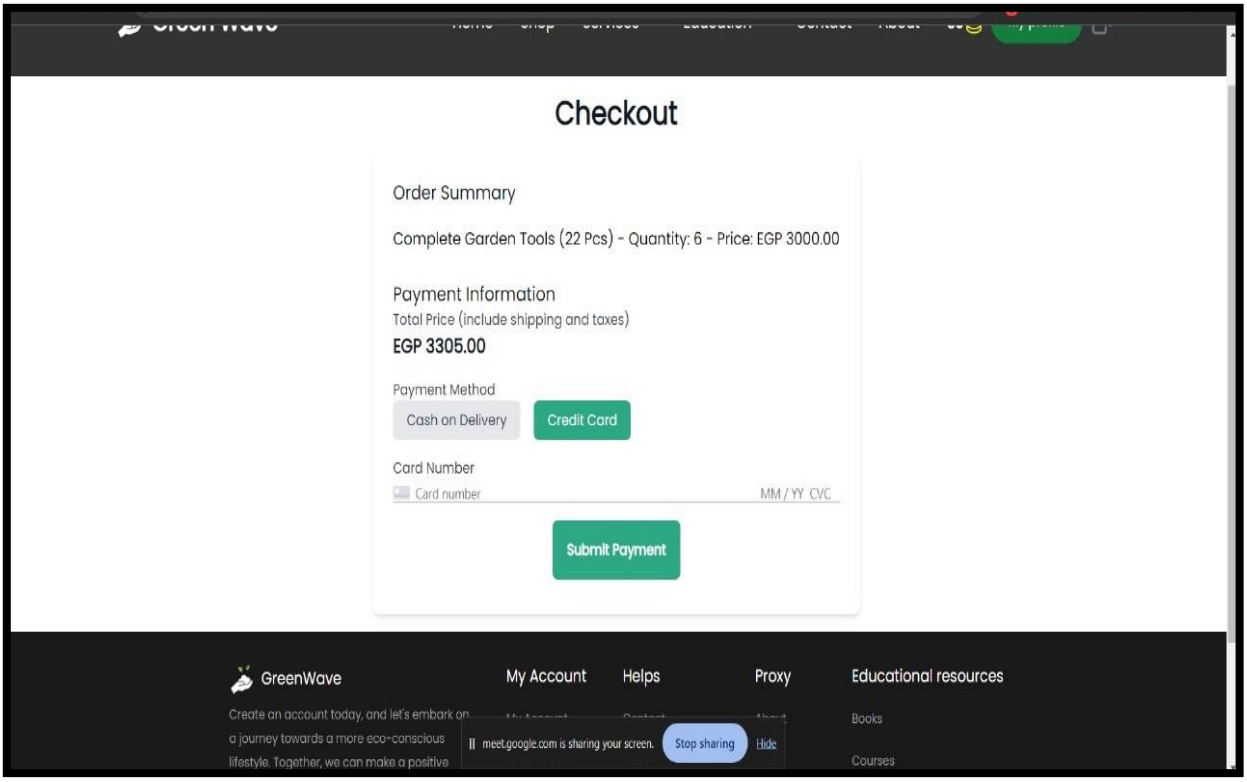

GreenWave
Home
Shop
Services
Education
About Us
Contact Us
Green Wave Waste Collection
We Can Remove Items From Inside Your Property.
Paper
Plastice
Construction
Glass
Metal
Organic
Waste Collection
What type of waste would you be interested in collecting?
Plastic
Metals
Glass
Organic Waste
Paper
Set Weight
Choose weight of the waste you want to remove
Waste Drop off
Upload Image
Choose appointment
Get paid
After the waste has been weighed you'll get paid for your efforts!
Current Profit
350 EGP
Request Appointment
GreenWave

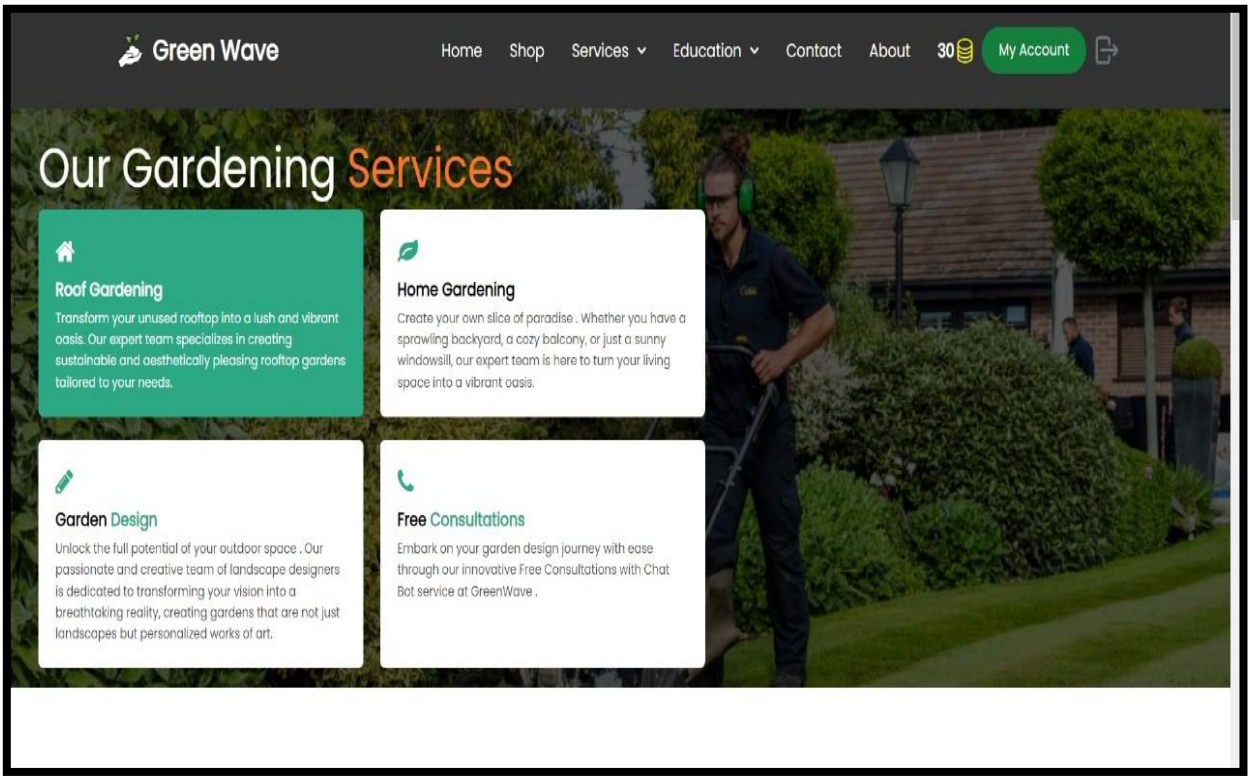

GreenWave
Sign In / Sign up
Home
Shop
Services
Education
About Us
Contact Us
Premium Learning Experience
Easily Accessible
Learning Will fell Very Comfortable With Courslab.
Fun learning expe
Learning Will fell Very Comfortable With Courslab.
Learn The Latest Skills
Get Ready For a Career
Earn a Certificate
Our Courses
Drip Irrigation System Design
500 EGP
A Beginners guide to Indoor Gardening
250 EGP
Advanced Guide to Build soil from scratch
350 EGP
Join Course
Join Course
Join Course
GreenWave
My Account
Helps
Proxy
Educational resources


GreenWave
Sign In / Sign up
Home
Shop
Services
Education
About Us
Contact Us
Sustainable & Gardening Tools
Sale up to 30% OFF
Free shipping on all your order.
Shop now
Free Shipping
Customer Support 24/7
100% Secure Payment
Money-Back Guarantee
Latest Articles
AI and agriculture
Read More
Create a Low Maintenance Landscape
Read More
Global warming and climate changes
Read More
GreenWave
My Account
Helps
Proxy
Educational resources


GreenWave
Search
Sign In / Sign up
Home
Shop
Services
Education
About Us
Contact Us
BEST DEALS
Sale of the Month
66%
Shop Now
Select Category
Select Price
Select Rating
Sort by: Latest
Gardening tools
Soil enhancers
Min 300 EGP - Max 500 EGP
8 Results found
Complete Garden Tools (22 Pcs)
500 EGP
Fiskars Bypass Pruning Shears
90 EGP
Stirrup Hoe
70 EGP
Large Repotting Mat
140 EGP
Out of Stock
Bliss Soil Enhancer
120 EGP
Hydroponic Root Enhancer
90 EGP
CO2 for indoor growing
90 EGP
Holes for Indoor Outdoor Plants
100 EGP
GreenWave
My Account
Helps
Proxy
Educational resources


GreenWave
Search
Sign In / Sign up
Home
Shop
Services
Education
About Us
Contact Us
Raised-Bed Gardening
Tammy Wylie
500 EGP
Add to cart
Essential Coping Skills
Isabella Finn
350 EGP
Add to cart
Soil Science for Beginners
Cassandra Clare
250 EGP
Add to cart
Vegetable Gardener's Bible
Cassandra Clare
270 EGP
Add to cart
إرشادات نوعية عامة
Free
Download
نباتات الزينة للحدائق المنزلية
Free
Download
الاسمدة واستعمالاتها
Free
Download
تحويل اسطح المباني إلى حدائق
Free
Download
GreenWave
My Account
Helps
Proxy
Educational resources


Checkout
Order Summary
Complete Garden Tools (22 Pcs) - Quantity: 6 - Price: EGP 3000.00
Payment Information
Total Price (include shipping and taxes)
EGP 3305.00
Payment Method
Cash on Delivery
Credit Card
Card Number
Card number
MM / YY CVC
Submit Payment
GreenWave
Create an account today, and let's embark on a journey towards a more eco-conscious lifestyle. Together, we can make a positive
My Account
Helps
Proxy
Educational resources
Books
Courses
meet.google.com is sharing your screen.
Stop sharing
Hide


Green Wave
Home
Shop
Services
Education
Contact
About
30
My Account
Our Gardening Services
Roof Gardening
Transform your unused rooftop into a lush and vibrant oasis. Our expert team specializes in creating sustainable and aesthetically pleasing rooftop gardens tailored to your needs.
Home Gardening
Create your own slice of paradise . Whether you have a sprawling backyard, a cozy balcony, or just a sunny windowsill, our expert team is here to turn your living space into a vibrant oasis.
Garden Design
Unlock the full potential of your outdoor space . Our passionate and creative team of landscape designers is dedicated to transforming your vision into a breathtaking reality, creating gardens that are not just landscapes but personalized works of art.
Free Consultations
Embark on your garden design journey with ease through our innovative Free Consultations with Chat Bot service at GreenWave .

Chapter 5: Implementation and Testing

Implementation

Frontend

React Our main technology for the front-end of our web application is React. We chose React because it offers the following benefits:

- **Component-based Architecture:** React's component-based architecture allows for reusability of UI components, leading to faster development and easier maintenance.
- **Virtual DOM:** React's virtual DOM improves performance by minimizing direct manipulation of the actual DOM.
- **Strong Ecosystem:** React has a strong ecosystem with a variety of libraries and tools that can be integrated to enhance functionality.

Backend

Node.js + Express.js Our main technology for the back end of our application is Node.js with Express.js. We chose Node.js and Express.js because they offer the following benefits:

- **Asynchronous and Event-driven:** Node.js handles multiple requests efficiently with its non-blocking I/O model.
- **Single Programming Language:** Using JavaScript for both frontend and backend allows for a consistent development experience.
- **Rich Ecosystem:** Node.js has a rich ecosystem of modules and packages, available through npm, which can be used to add functionality quickly.

Database

MongoDB We chose MongoDB as our database because it offers the following benefits:

- **Schema Flexibility:** MongoDB's flexible schema design allows for quick iterations and changes in data structure.
- **Scalability:** MongoDB can handle large volumes of data and scale horizontally by adding more servers.
- **Rich Query Language:** MongoDB supports a rich query language and aggregation framework for data analysis.
- **NoSQL**: MongoDB is a NoSQL database, which means it stores data in a flexible, JSON-like format, making it easier to handle diverse data types and structures.

Version Control

GitHub

To ensure efficient collaboration and version management throughout the development process, we have utilized Git as our version control system. Git allows us to track changes, manage code repositories, and collaborate seamlessly with team members. With Git, we can create branches for different features or bug fixes, merge code changes, and maintain a comprehensive history of our project's evolution. By utilizing Git's branching and merging capabilities, we can work concurrently on different aspects of the project, facilitating efficient collaboration and ensuring a streamlined development workflow.

Endpoint Testing

Postman

To verify the functionality and integrity of our API endpoints, we have utilized Postman for endpoint testing. Postman is a popular API development and testing tool that enables us to send requests to our API endpoints, inspect responses, and validate the expected behaviour. With Postman, we can simulate various scenarios, test different endpoints, and ensure the proper functioning of our backend API. This allows us to identify and resolve any issues or bugs, ensuring that our system's API endpoints perform as intended and provide accurate and reliable responses to client applications.

By incorporating React for the frontend, Node.js + Express for the backend, MongoDB for the database, Git for version control, and Postman for endpoint testing, we have established a robust and efficient development environment. These technologies and tools contribute to the overall success of our project by ensuring high-quality code, effective collaboration, and reliable functionality.

Gemini AI: Chatbot Integration

In our project, we have integrated Gemini AI as a chatbot to enhance user interaction and support. Gemini AI provides the following benefits:

1. Natural Language Processing (NLP): Gemini AI leverages advanced NLP techniques to understand and process user queries effectively. This allows the chatbot to interpret user inputs accurately and provide relevant responses, enhancing the user experience.

2. 24/7 Availability: With Gemini AI, our platform offers support to users. The chatbot is always available to answer questions, assist with navigation, and provide information, ensuring users receive immediate assistance whenever needed.

3. Contextual Understanding: Gemini AI can maintain context throughout a conversation, enabling more meaningful and coherent interactions. This context-awareness helps the chatbot understand follow-up questions and provide responses that are relevant to the ongoing conversation.

4. Seamless Integration: Gemini AI is seamlessly integrated into our platform, ensuring a smooth and consistent user experience. Users can interact with the chatbot through the website interface without the need for additional installations or configurations.

5. Continuous Learning: The AI continuously learns and improves over time based on user interactions. This ensures that the chatbot remains up-to-date with the latest information and can provide increasingly accurate and helpful responses.

By incorporating Gemini AI into our platform, we aim to provide an enhanced, interactive, and supportive user experience. The chatbot not only improves user engagement but also helps in efficiently managing user queries and support requests.

Testing

1- User Login Test Case Table

Test Scenario	Verify login with email and password
Pre-condition	Sign up

Table 1: t Login Test Scenario

Test Steps:
1.Enter Email
2.Enter Password
3.Click "Sign In" button

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
Login_1	Enter Valid Email & Password	Email mady112018@gmail.com Password ******	Return JWT Token Redirect to Home Page	**Return JWT Token Redirect to Home Page**
Login_2	Enter invalid Email & valid Password	Email doaa32@gmail.comPassword *******	Print: Invalid Email OR Password	**Print: Invalid Email OR Password**
Login_3	Enter valid Email & invalid Password	Email Ziaddiaa@gmail.comPassword *******	Print: Wrong Password	**Print: Invalid Email OR Password**
Login_4	**Enter Email That does notexist & Valid Pass**	**Email a.samy@gmail.com Password ******* **	**Print: Email does not exist**	**Print: Email does not exist**

2- User Sign up Test Case Table

Test Scenario	Create a new user and add it to Database
Pre-condition	-

Table 2: User Sign up Test Scenario

Test Steps:

- *Enter Email*
- *Enter Password*
- *Enter Full Name*
- *Enter Birth Date and Phone*
- *Click "Sign Up" button*

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
Signup_1	Enter ValidData	Email nadasamyy@gmail.comPassword ******* Phone 1234567	Return User Registered successfully. Redirect to Login Page	**Return User Registered successfully.Redirect to Login Page**
SignUp_2	Enter Email that already exist	Email ahla.m@gmail.com	Print: Email Already Exist	**Print: Email Already Exist**
SignUp_3	Leave a missing data	Email Empty Pass *******	Print: Cannot Signup, please complete any missing Data	**Print: Cannot Signup, please complete anymissing Data**
SignUp_4	Enter InvalidEmail (Without '@')	Email Example.com	Print: Cannot Signup, please enter valid email	**Print: Cannot Signup, pleaseenter valid email**
SignUp_5	**Enter short Password 4>**	**Password *****	**Print: Please choose stronger Password**	**Print: Please choose stronger Password**

1- Add New Book

Test Scenario	Add Book for Specific Category
Pre-condition	Admin

Table 3: Add New Book Test Scenario

Test Steps:

- Add Book (Name, Author, genre , Publish Date, Description)
- Enter Book (Type, Book Price)
- Enter Book Provider
- Click “ADD” button

ID	Description	Test Data	Expected Results	Actual Results
TC_001	Add a new book with valid data	"title": "green", "author": "مؤسسة هانس زاي", "genre": "Gardening", "publishedDate": "2009-01-01T00:00:00.000Z" , "price": 0, "description": " يأتي إصدار هذا الكتيب في إطار أنشطة مؤسسة هانس النيل ا", "coverImage": "", "createdBy": "667df670039c83f0af48249f"}	HTTP 200, { message: 'Book added successfully', book: { ... } }	{ message: 'Book added successfully', book: { ... } }
TC_002	Add a new book with missing required field	"title": "green", "author": "مؤسسة هانس زاي", "genre": "Gardening", "price": 0, "description": " يأتي إصدار هذا الكتيب في إطار أنشطة مؤسسة هانس النيل ا", "coverImage": "", "createdBy": "667df670039c83f0af48249f"}	HTTP 400, { message: 'publishedDate is required' }	Book validation failed: 'publishedDate Path:'publishedDate is required'

TC_003	Update an existing book	{ bookId: "668703d94515995779e96b7", title: "waste collection", author: "Updated Author", genre: "Non-Fiction", publishedDate: "2023-02-01", price: 29.99, description: "An updated great book", coverImage: "updated_cover.jpg" }	HTTP 200, { message: 'Book updated successfully', book: { ... } }	{ message: 'Book updated successfully', book: { ... } }
TC_004	Update a non-existing book	{ bookId: "668703d9", title: "Updated Title" }	HTTP 404, { message: 'Book not found' }	{ message: 'Book not found' }
TC_005	Delete an existing book	{ bookId: "668703d94515995779e96b74" }	HTTP 200, { message: 'Book deleted successfully' }	message: 'Book deleted successfully'
TC_006	Delete a non-existing book	{ bookId: "668703d9" }	HTTP 404, { message: 'Book not found' }	message: 'Book not found' }
TC_007	Get all books	{ }	HTTP 200, [{ ...bookDetails... }]	[{ ...bookDetails... }]

Table3: Add New Book Test Case

2- Add Articles Test Cases

Test Scenario	Add Articles
Pre-condition	Admin

Table 4: Add New article Test Scenario

Test Steps:

- Add a new article.
- Add a new article with missing fields.
- Update an existing article.
- Update a non-existing article.
- Delete an existing article.
- Delete a non-existing article.
- Search for articles by title.

Test Case ID	Test Case Description	**Test Data**	Expected Result	Actual Result
TC_001	Add a new article with valid data	{"title": "waste colloction", "content": "This is the content of the new article.", "author": "Jane Doe", "tags": ["tag1", "tag2"], "coverImage": "", "createdBy": "667df670039c83f0af48249f" }	HTTP 201, { message: 'Article added successfully', article: { ... } }	{ message: 'Article added successfully', article: { ... } }
TC_002	Add a new article with missing field	{"title": "waste colloction", "content": "This is the content of the new article.", "tags": ["tag1", "tag2"], "coverImage": "", "createdBy": "667df670039c83f0af48249f" }	HTTP 400, { message: 'author is required' }	message: 'author is required' }

TC_003	Update an existing article	{articleId: "6686cf5b4515995779e96b3a", "title": "planett", "content": "This is the updated content of the article.", "author": "Jane Doe", "tags": ["updatedTag1", "updatedTag2"],"coverImage": ""}	HTTP 200, { message: 'Article updated successfully', article: { ... } }	{ message: 'Article updated successfully', article: { ... } }
TC_004	Update a non-existing article	{ articleId: "999", title: "Updated Article" }	HTTP 404, { message: 'Article not found' }	{ message: 'Article not found' }
TC_005	Delete an existing article	{ articleId: "6686cf5b4515995779e96b3a " }	HTTP 200, { message: 'Article deleted successfully' }	{ message: 'Article deleted successfully' }
TC_006	Delete a non-existing article	{ articleId: "999" }	HTTP 404, { message: 'Article not found' }	{ message: 'Article not found' }
TC_007	Search for articles by title	{ query: "AI and agriculture " }	HTTP 200, [{ ...articleDetails... }]	[{ ...articleDetails... }]

Table 4: Add New Article Test Case

3- Waste Collection Service Test Case Table

Test Scenario	Verify waste collection scheduling
Pre-condition	User is registered and logged in

Table 5:Waste Collection Test Scenario

Test Steps:

- Enter User ID
- Select Waste Type
- Choose Date
- Choose Time
- Click "Schedule" button

ID	Description	Test Data	Expected Result	Actual result
Tc_001	Schedule Waste Collection - Valid Input	{ "userId": "60d21b4667d0d8992e610c85", "wasteType": "Plastic", "date": "2024-07-10", "time": "10:00 AM" }	Appointment scheduled successfully and confirmation email sent	Appointment scheduled successfully and confirmation email sent
Tc_002	Schedule Waste Collection - Missing UserID	{ "wasteType": "Glass", "date": "2024-07-10", "time": "10:00 AM" }	Error scheduling appointment: Path `userId` is required	Error scheduling appointment: Path `userId` is required
Tc_003	Schedule Waste Collection - Invalid WasteType	{ "userId": "60d21b4667d0d8992e610c85", "wasteType": "Wood", "date": "2024-07-10", "time": "10:00 AM" }	Error scheduling appointment: `Wood` is not a valid enum value for path `wasteType`	Error scheduling appointment: `Wood` is not a valid enum value for path `wasteType`
Tc_004	Schedule Waste Collection - Missing Date	{ "userId": "60d21b4667d0d8992e610c85", "wasteType": "Organic", "time": "10:00 AM" }	Error scheduling appointment: Path `date` is required.	Error scheduling appointment: Path `date` is required.
Tc_005	Schedule Waste Collection - Missing Time	{ "userId": "60d21b4667d0d8992e610c85", "wasteType": "Metal", "date": "2024-07-10" }	Error scheduling appointment: Path `time` is required	Error scheduling appointment: Path `time` is required
Tc_006	Get Waste Types	None	["Plastic", "Glass", "Metal", "Organic"	["Plastic", "Glass", "Metal", "Organic"

Table 5: Waste Collection Test Case

4- Urban Greening Service Test Case Table

Test Scenario	Verify waste collection scheduling
Pre-condition	User is registered and logged in

Table 6: Urban Greening Test Scenario

Test Steps:

- Enter User ID
- Select Greening Type
- Choose Date
- Choose Time
- Click "Request Assistance" button

ID	Description	Test Data	Expected Result	Actual Result
TC_1	Request Urban Greening Assistance - Valid Input	{ "userId": "60d21b4667d0d8992e610c85", "greeningType": "Tree Planting", "date": "2024-07-12", "time": "2:00 PM" }	Request submitted successfully and confirmation email sent	Request submitted successfully and confirmation email sent
TC_2	Request Urban Greening Assistance - Missing UserID	{ "greeningType": "Community Garden", "date": "2024-07-12", "time": "2:00 PM" }	Error submitting request: Path `userId` is required.	Error submitting request: Path `userId` is required.
TC_3	Request Urban Greening Assistance - Missing Greening Type	{ "userId": "60d21b4667d0d8992e610c85", "date": "2024-07-12", "time": "2:00 PM" }	Error submitting request: Path `greeningType` is required.	Error submitting request: Path `greeningType` is required.
TC_4	Request Urban Greening Assistance - Missing Date	{ "userId": "60d21b4667d0d8992e610c85", "greeningType": "Tree Planting", "time": "2:00 PM" }	Error submitting request: Path `date` is required.	Error submitting request: Path `date` is required.

TC_5	Request Urban Greening Assistance - Missing Time	{ "userId": "60d21b4667d0d8992e610c85", "greeningType": "Community Garden", "date": "2024-07-12" }	Error submitting request: Path `time` is required.	Error submitting request: Path `time` is required.
TC_6	Get Urban Greening Guidance	None	[{ type: 'Tree Planting', info: 'Guidelines and support for tree planting.' }, { type: 'Community Garden', info: 'Guidelines and support for setting up a community garden.' }]	[{ type: 'Tree Planting', info: 'Guidelines and support for tree planting.' }, { type: 'Community Garden', info: 'Guidelines and support for setting up a community garden.' }]

Table 6: Urban Greening Test Case

5- Make OrderTest Case Table

Test Scenario	Verify order creation, status updates, and coupon application
Pre-condition	User is registered and logged in

Table 6: Urban Greening Test Scenario

Test Steps:

- Enter User ID
- Select Items
- Enter Address
- Enter Phone
- Add Notes (optional)
- Choose Payment Method
- Click "Create Order" button

ID	Description	Test Data	Expected Data	Actual Data
TC_1	Create Order - Valid Input	{ "items": [{ "itemType": "Product", "itemId": "60d21b4667d0d8992e610c85", "quantity": 1, "price": 100 }], "address": "123 Street", "phone": "1234567890", "notes": "Leave at door", "paymentMethod": "Cash" }	Order created successfully with status 'pending'	Order created successfully with status 'pending'
TC_2	Create Order - Missing UserID	{ "items": [{ "itemType": "Product", "itemId": "60d21b4667d0d8992e610c85", "quantity": 1, "price": 100 }], "address": "123 Street", "phone": "1234567890", "notes": "Leave at door", "paymentMethod": "Cash" }	Error: Path `userId` is required.	Error: Path `userId` is required.
TC_3	Create Order - Invalid ItemType	{ "userId": "60d21b4667d0d8992e610c85", "items": [{ "itemType": "InvalidType", "itemId": "60d21b4667d0d8992e610c85", "quantity": 1, "price": 100 }], "address": "123 Street", "phone": "1234567890", "notes": "Leave at door" }	Error: Invalid item type	Error: Invalid item type

TC_4	Create Order - Missing Address	{ "userId": "60d21b4667d0d8992e610c85", "items": [{ "itemType": "Product", "itemId": "60d21b4667d0d8992e610c85", "quantity": 1, "price": 100 }], "phone": "1234567890", "notes": "Leave at door", "paymentMethod": "Cash" }	Error: Path `address` is required.	Error: Path `address` is required.
TC_5	Create Order - Missing ItemID	{ "userId": "60d21b4667d0d8992e610c85", "items": [{ "itemType": "Product", "quantity": 1, "price": 100 }], "address": "123 Street", "phone": "1234567890", "notes": "Leave at door", "paymentMethod": "Cash" }	Error: Path `itemId` is required.	Error: Path `itemId` is required.
TC_6	Get User Orders	None	List of orders for the user	List of orders for the user
TC_7	Update Order Status - Valid Input	{ "orderId": "60d21b4667d0d8992e610c85", "status": "shipped" }	Order status updated successfully	Order status updated successfully
TC_8	Update Order Status - Invalid OrderID	{ "orderId": "invalidOrderID", "status": "shipped" }	Error: Order not found	Error: Order not found
TC_9	Cancel Order - Valid Input	{ "orderId": "60d21b4667d0d8992e610c85", "reason": "Changed my mind" }	Order cancelled successfully	Order cancelled successfully
TC_10	Cancel Order - Already Delivered	"orderId": "60d21b4667d0d8992e610c85", "reason": "Changed my mind" }	Error: this order has already been delivered	Error: this order has already been delivered

TC_11	Apply Coupon - Valid Coupon	{ "userId": "60d21b4667d0d8992e610c85", "items": [{ "itemType": "Product", "itemId": "60d21b4667d0d8992e610c85", "quantity": 1, "price": 100 }], "address": "123 Street", "phone": "1234567890", "notes": "Leave at door", "coupon": "60d21b4667d0d8992e610c85" }	Order created successfully with discount applied	Order created successfully with discount applied
TC_12	Apply Coupon - Invalid Coupon	{ "userId": "60d21b4667d0d8992e610c85", "items": [{ "itemType": "Product", "itemId": "60d21b4667d0d8992e610c85", "quantity": 1, "price": 100 }], "address": "123 Street", "phone": "1234567890", "notes": "Leave at door", "coupon": "invalidCouponID" }	Error: Invalid or expired coupon	Error: Invalid or expired coupon

Table 7: Make Order Test Case

Chapter 6: Conclusion and future Work

Conclusion

In conclusion, the development of our platform aims to empower users with sustainable living practices and environmental conservation efforts. The platform is designed to facilitate a wide range of activities, including scheduling waste collection appointments, accessing educational resources on agriculture, engaging in urban greening activities, purchasing eco-friendly products, and reporting waste sites for community cleanup.

Throughout the project development process, we employed modern programming techniques and best practices to ensure high performance and a user-friendly interface. Our use of React for the frontend, Node.js with Express.js for the backend, and MongoDB for the database has allowed us to create a robust and scalable platform. Additionally, our integration of a chatbot using machine learning technologies such as Python, NLTK, and TensorFlow has enhanced user interaction and support.

The main objectives of the project have been achieved, including providing a comprehensive platform for sustainable living and environmental conservation, aiding users in making informed decisions about waste management and eco-friendly practices, and fostering a community of environmentally conscious individuals.

We believe that our platform will be a powerful and valuable tool for individuals seeking to adopt sustainable living practices and contribute to environmental conservation. The platform enables users to gain access to educational resources, engage in green initiatives, and make eco-friendly purchases, all while earning rewards for their contributions to the environment.

We look forward to sharing the platform with users and receiving their feedback and comments to continue improving and enhancing our services in the future.

Finally, we would like to express our gratitude to all the individuals who helped us in this project, especially **Dr. Khaled Wassef**, who provided us with invaluable support and guidance. We hope that our platform will prove to be useful and effective in promoting sustainable living and environmental conservation for our users.

➢ Future Work and Feature Enhancements

1. **Mobile Application Development:**
    - Develop a mobile application for both Android and iOS platforms to provide users with a more accessible and convenient way to interact with the platform.
2. **Advanced Waste Management Features:**
    - Implement AI and machine learning algorithms to optimize waste collection routes and schedules, reducing operational costs and environmental impact.
    - Introduce a real-time tracking system for waste collection vehicles to provide users with updates on their scheduled appointments.
3. **Expanded Marketplace:**
    - Expand the marketplace to include a wider range of eco-friendly products and services.
    - Introduce a subscription service for regular delivery of eco-friendly products.
4. **Personalized Recommendations:**
    - Implement a recommendation engine that suggests products, courses, and articles based on user behavior and preferences.
    - Provide personalized gardening tips and urban greening advice based on the user's location and interests.
5. **Community Features:**
    - Develop community forums where users can share experiences, tips, and advice on sustainability, waste management, and urban greening.
    - Organize community events and workshops to promote eco-friendly practices and urban greening initiatives.
6. **Enhanced Educational Resources:**
    - Develop interactive courses and webinars on topics related to sustainability, waste management, and urban greening.
    - Collaborate with experts and institutions to provide accredited courses and certifications.

References:

https://carnegieendowment.org/2023/10/26/climate-change-in-egypt-opportunities-and-obstacles-pub-90854

https://www.mashtalegypt.com/

https://growpro-eg.com/

https://www.atlassian.com/agile/project-management/waterfall-methodology

https://theurbanactivist.com/idea/food-security-urban-greens-cultivates-resilience-on-the-rooftops-of-cairo/

https://ourworldindata.org/co2/country/egypt