Project Report

On

"Stock Management system"

SUBMITTED TO

ROURKELA INSTITUTE OF MANAGEMENT STUDIES

(As a Partial fulfilment of the requirement for the award of degree)

FOR

"MASTER IN COMPUTER APPLICATION "

(2023-25)

SUBMITTED BY

TITHI SARANGI

Registration Roll No:- 2305260026

MCA 4th SEMESTER

ROURKELA INSTITUTE OF MANAGEMENT STUDIES

(Affiliated to Biju Pattnaik University of Technology, Odisha)

Rourkela – 769015

Rourkela Institute of Management Studies
Rourkela
Department of Computer Science
Rourkela Institute of Management Studies
Chhend, Rourkela-15, Odisha
Phone:0661 2480482
Fax:91-0661-1480665
Mail: rkl_rimsgrol@sancharnet.in
Visit: www.rims-edu.com

CERTIFICATE OF EXAMINATION

This is to certify that this project report entitled "**Stock Management System**" submitted by **TITHI SARANGI** of 4th Semester, **Rourkela Institute of Management Studies, Rourkela**, is accepted as partial fulfillment of requirements for the degree in **Master in Computer Applications**, under **Biju Pattnaik University of Technology, Rourkela**, this has been verified by us and found be original up to our satisfaction.

External Examiner

Rourkela Institute of Management Studies
Rourkela
Department of Computer Science
Rourkela Institute of Management Studies
Chhend, Rourkela-15, Odisha
Phone:0661 2480482
Fax:91-0661-1480665
Mail: rkl_rimsgrol@sancharnet.in
Visit: www.rims-edu.com

CERTIFICATE OF EXAMINATION

This is to certify that this project report entitled "**Stock Management System**" submitted by **TITHI SARANGI** of 4th Semester, **Rourkela Institute of Management Studies, Rourkela**, is accepted as partial fulfillment of requirements for the degree in **Master in Computer Applications**, under **Biju Pattnaik University of Technology, Rourkela**, this has been verified by us and found be original up to our satisfaction.

Internal Examiner

Rourkela Institute of Management Studies
Rourkela
Department of Computer Science
Rourkela Institute of Management Studies
Chhend, Rourkela-15, Odisha
Phone:0661 2480482
Fax:91-0661-1480665
Mail: rkl_rimsgrol@sancharnet.in
Visit: www.rims-edu.com

CERTIFICATE

This is to certify that this project entitled **"STOCK MANAGEMENT SYSTEM"** has been and submitted by **TITHI SARANGI ,** M.C.A 2023-2025, **Rourkela Institute of Management Studies, Rourkela,** has been examined by us. She is found fit and approved for the award of **"Master in Computer Application "**Degree.
To the best my knowledge this work has not been submitted for the award of any other degree.
I wish all success in his life.

DEAN ACADEMIC RIMS, ROURKELA

Prof. Bibhudendu Panda Head of The
Department, MCA
Rourkela Institute of Management Studies, Rourkela

CERTIFICATE

This is to certify that **TITHI SARANGI** student of **M.C.A, Rourkela Institute of Management Studies, Rourkela, Odisha** of Session 2023-2025 has completed the project successfully.

I wish all success in his life.

(Prof. Bibhudendu Panda)

DECLARATION

I am **TITHI SARANGI** , hereby declare that the project report Entitled "**STOCK MANAGEMENT SYSTEM**" is of my work. The above work I submitted to "**Biju Patnaik University of Technology, Rourkela"** for the award of **"Master in Computer Applications**" Degree.

To the best of my knowledge, this work has not been submitted or published anywhere for the award of any degree.

TITHI SARANGI

ACKNOWLGEMENT

I am deeply indebted to **Rourkela Institute of Management Studies, Chhend, Rourkela,** for providing me an opportunity to undertake a project work entitled **"STOCK MANAGEMENT SYSTEM".**
I am grateful to my project guide **Prof. Bibhudendu Panda** without his guidance it would not have been possible on my part to complete the project.
I acknowledge the help and co-operation received from all my team members in making this project.
I consider myself fortunate that I have successfully completed this project; I acknowledge my sincere gratitude to all those works and ideas that had helped me in writing this project.

TITHI SARANGI
University Roll No: 2305260026
MCA (2023-2025)
Rourkela Institute of Management Studies,
Rourkela.

Abstract

The Stock Management System is a software application developed to streamline and automate the process of managing inventory within an organization. The system is designed to efficiently handle stock-related activities such as tracking product availability, monitoring stock levels, managing purchase and sales records, generating reports, and ensuring real-time updates. This project aims to reduce the manual efforts involved in inventory management, minimize errors, and improve the overall accuracy and efficiency of stock control operations.

The system enables users to add, update, and delete stock items, categorize products, maintain supplier and customer records, and monitor transactions. It supports various reports such as current stock status, reorder levels, sales reports, and purchase summaries, aiding in informed decision-making. The application also provides alerts for low-stock items, helping to prevent stock-outs and overstocking. This system can be deployed in retail shops, warehouses, manufacturing units, and other business domains where inventory control is critical.

Overall, the Stock Management System enhances operational productivity, reduces overhead costs, and ensures timely availability of products, thereby contributing to better resource planning and customer satisfaction.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

Background

Grocery stores used to do their inventory manually before the new dawn of technology to mankind, large stores found it hard to maintain their operations efficiently and effectively. Sales and stock management systems were developed to carry out the daily activities whether in grocery stores or companies. With a proper and timely determination of the system, the strategy allows for freeing a significant number of stocks increasing resource efficiency. Excel spreadsheets are used for data entry manually which can lead to great inaccuracies in the information processed. Significance Using Excel sheets for maintaining sales and stock in a store may result in lot of challenges. Some of them are getting errors while entering the data manually, as the data increases scalability decreases and difficult to interact with other systems like point -of-sales system. Due to all these reasons, it is really difficult to decide about sales and stocks based on real -time details. Hence we need to have an integrated system for all kinds of store in order to monitor sales and stock.

Purpose

The key goals of this initiative are as follows:

1. Automate the sales and stock management processes, reducing manual errors and increasing efficiency.

2. Provide real-time visibility into sales and stock data, enabling informed decision-making.

3. Enhance the customer experience through improved checkout processes and customer management.

4. Streamline the inventory management process, including stock levels, reordering, and reporting.

5. Provide detailed sales and stock analytics, including graphs and charts, to help the retail store track its performance.

Motivation

In the world we belong to right, now every business tries to achieve a balance between what is required and what is desired, with reducing expenses as the primary objective. Goods and materials that are used in any business are controlled by sales and stock management tools. It examines the amount of supplies that is accessible and exactly where it is kept so that it is readily available to utilize whenever needed. For predicting the demand in future, the system deals with forecasting the demand, controlling assets items and supplies inventory with a cost, estimation, valuation of commodities and validation of products.

CHAPTER TWO

LITERATURE REVIEW

Challenges

There are numerous sales and stock systems used in almost every business enterprise for managing their inventory and sales, although inaccurate records OD stock is universal among the systems. The proposed system will develop a method for delivering real-time insights into the activities carried along the grocery store. Existing systems are using the same type of database technology for data visualization, the study of presented abstract information in certain schematic forms. The data visualization goal is to relay information graphically using shapes, arts, or charts with percentages

Database is a group of connected files that are structured to be accessible to end users simultaneously. It gathers and organizes data so that it can be displayed in a single location. They are created and carried by a software known as Database Management System, which deals with the way of information maintained, structured and accessed. Structured Query Language is implemented by other applications for processing data in a relational database management systems.MY SQL and PHP are used in this system for remote server services.

CHAPTER THREE

SYSTEM REQUIREMENTS

Hardware Requirements

- Display: 1920x1080 resolution Monitor.
- Processor: Intel Core i5 or higher.
- Network: Internet connectivity or Wi-Fi.
- Storage: At least 256GB SSD.
- RAM: 8GB or higher.

Software Requirements

- Visual Studio Code.
- Git
- My SQL
- XAMPP

CHAPTER FOUR

TOOLS AND TECHNOLOGIES

Graphical User Interface will be the preferred front-end interface to information display, since easy to use even for inexperienced users, it is attractive, provide shortcuts, and allows room for multitasking.

HTML

Hyper Text Mark-up Language is used for developing front-end Graphical user interface. It is standard language used for web pages.

CSS

Cascading style sheets are a style sheet language used to show a document or content written in mark-up language.

SPRING BOOT

Spring Boot is a Java framework that makes it easier to create and run Java applications. It simplifies the configuration and setup process, allowing developers to focus more on writing code for their applications.

PHP

The system was created with the LA ravel PHP framework. The software will be developed using PHP one of the most widely used and reliable technologies for developing custom software solutions.

MYSQL

For database MYSQL is used. This technology will ensure that the software is scalable, reliable, and secure.

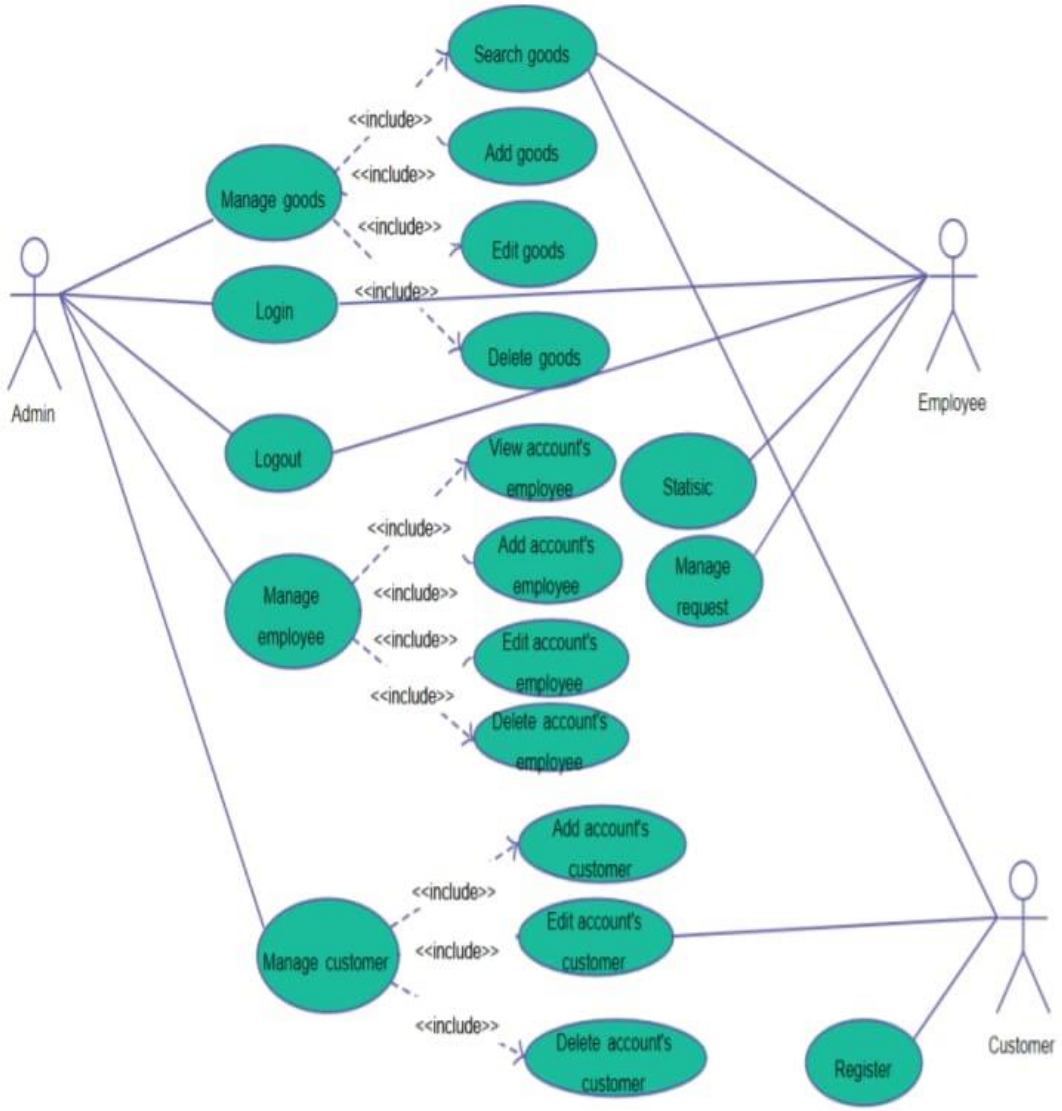
JAVA

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application developers Write Once and Run Anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

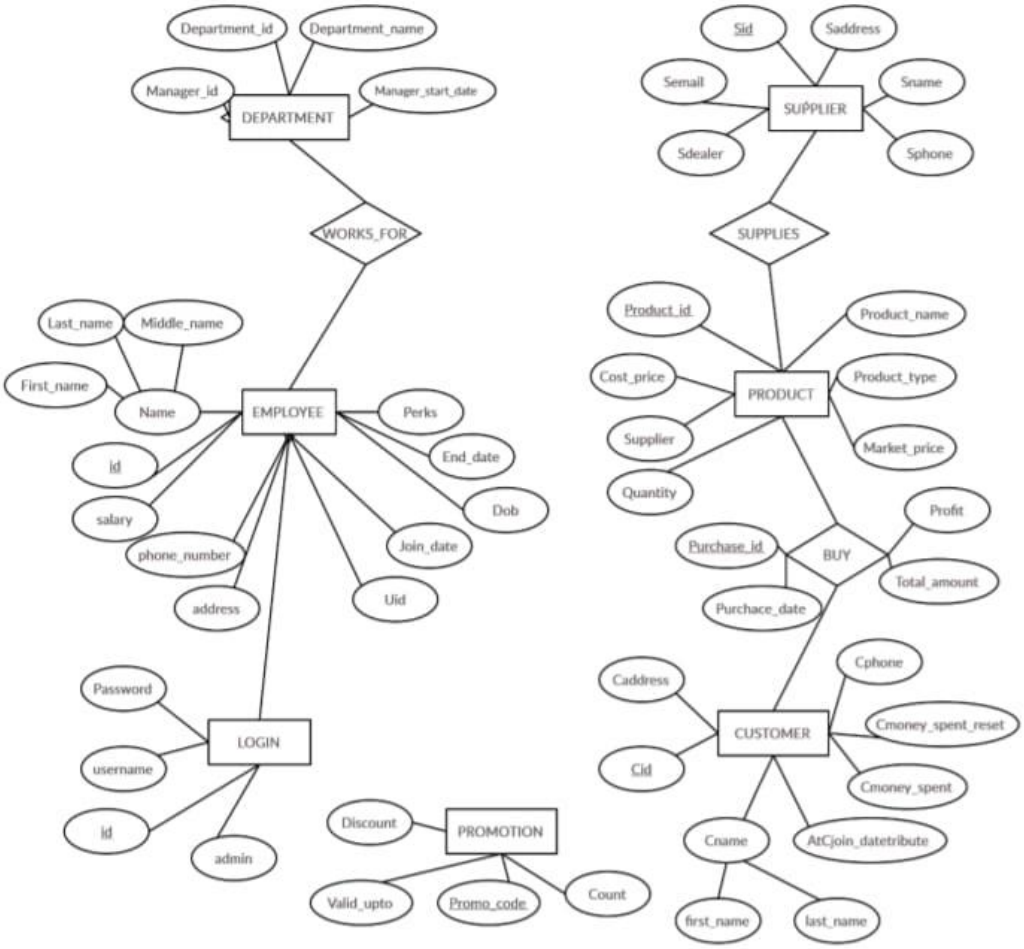
CHAPTER FIVE

SYSTEM DESIGN

Use Case Diagram



ER Diagram


Department_id
Department_name
Manager_id
DEPARTMENT
Manager_start_date
WORKS_FOR
Last_name
Middle_name
First_name
Name
EMPLOYEE
Perks
End_date
Dob
Join_date
Uid
id
salary
phone_number
address
Password
LOGIN
username
id
admin
Discount
PROMOTION
Valid_upto
Promo_code
Count
Sid
Saddress
Semail
SUPPLIER
Sname
Sdealer
Sphone
SUPPLIES
Product_id
Product_name
Cost_price
PRODUCT
Product_type
Supplier
Market_price
Quantity
Profit
Purchase_id
BUY
Total_amount
Purchace_date
Caddress
Cphone
CUSTOMER
Cmoney_spent_reset
Cid
Cmoney_spent
Cname
AtCjoin_datetribute
first_name
last_name

CHAPTER SIX

SYSTEM ANALYSIS

Proposed System

The Primary components of this system are as follows:

1. Point of Sale (POS): A simple-to-use interface that permits sales operations like management of clients, handling payments, and the scanning of barcodes efficiently and effectively.

2. Inventory Management: This application can help us in determining the current stock and handles the stock, reports and reordering.

3. Customer Management: This system will maintain and arrange data regarding the customers, which includes their contact details, purchase history and reward points.

4. Sales and Profit Analytics: This platform can display graphical and visual representations of both revenue and sales, which helps in assisting the store to make prudent decisions.

5. Reports: This application will create statistics regarding transactions, materials, consumers, and other aspects that can assist the business to maintain records of the manner in which it is accomplished.

Web Application

The main purpose of this application is to supervise the inventory management process of the business. When all the operations are automated, they can be carried out accurately and the business will acquire the edge over competitors.

The following details are added to the business requirements discussed in the Scope section:

1. Aids in the search for a specific product and its remaining supply.
2. Information regarding product sales and purchases.
3. Brief information on the organization's current news status.
4. Display inventory as of the date entered.
5. It aids in determining the total presented inventory in the organization.
6. To determine the amount and specifics of sales distributed on a certain day.
7. Inventory transactions are properly managed.
8. Each transaction has a unique entry date, as well as a quantity and rate.
9. Only the administrator has access to the page.

User Prerequisites

The user type determines the user requirements.

Admin

1. Ability to build new stores with dates.
2. Ability to alter the entry according to the entry.
3. Ability to add, change, and delete stock entries.

Inventory Control

1. Ability to verify stock availability.
2. Capable of checking the balance payment.
3. Ability to view the remaining sales inventory.

CHAPTER SEVEN

IMPLEMENTATION

Primary Implementation Admin:

The administrator is responsible for handling the system. This involves setting up and dealing with the user accounts, operating the database and maintaining how efficiently the system performs.

Store Manager:

The store manager will be responsible for running the store daily. He is also responsible for maintaining the data on sales and inventory, making reports. Implementing decisions on sales and inventory information.

Inventory Manager:

The inventory manager will be responsible for dealing with stock data, which involves incorporating new goods, keeping stock levels up to date and keeping record of expenses.

Reports User:

The reports user will be responsible for creating reports, examining sales and stock data, and coming up with alternatives based on what they discover through the data.

Each customer will be given individual login information and access. Thus confidential data will remain encrypted and every individual will be allowed to access the details they require to carry out the job. The PHP and MYSQL based inventory and sales management system will make it simpler and precise count of sales and stock data.

Exploring Database

Implementation of the PHP and MySQL-based Sales and Stock Management System for a small retail has the following steps:

• **Hosting:** A cloud-based hosting platform should be selected. This hosting site should have enough storage space, bandwidth to deal with large volumes of data is required.

• **Database Design**: For storing data regarding transactions and supplies. It is necessary to develop well organized database. The database must contain fields for transactions, products and inventory and the fields need to be interconnected in a proper way.

• **User Interface Design**: The entire system should have a user -friendly dashboard which allows customers to enter and display both sales and inventory data. The interface should be customizable and function properly on a number of platforms like laptops and mobile phones.

• **PHP Programming:** PHP scripts need to be developed to provide the system all the features like entering information, searching the data and generating reports. The code should be developed to be secure as well as expanding and must deal with error management and store them appropriately.

• Integration with Point-of-Sale (POS) Systems: PHP and My-SQL systems should be interconnected in order to get real-time data.

• **Testing and Deployment**: The system needs to be tested properly in order to ensure that it functions as planned also ensuring that the date precise and secured. When the testing is completed the system can be uploaded on the hosting site and made accessible to all the clients.

• **Maintenance and Upgrades**: Regular maintenance and upgrades must be performed on the system to ensure that it continues to function effectively and provide the necessary level of service. This may involve fixing bugs, adding new features, and upgrading the underlying software and hardware components.

CHAPTER EIGHT

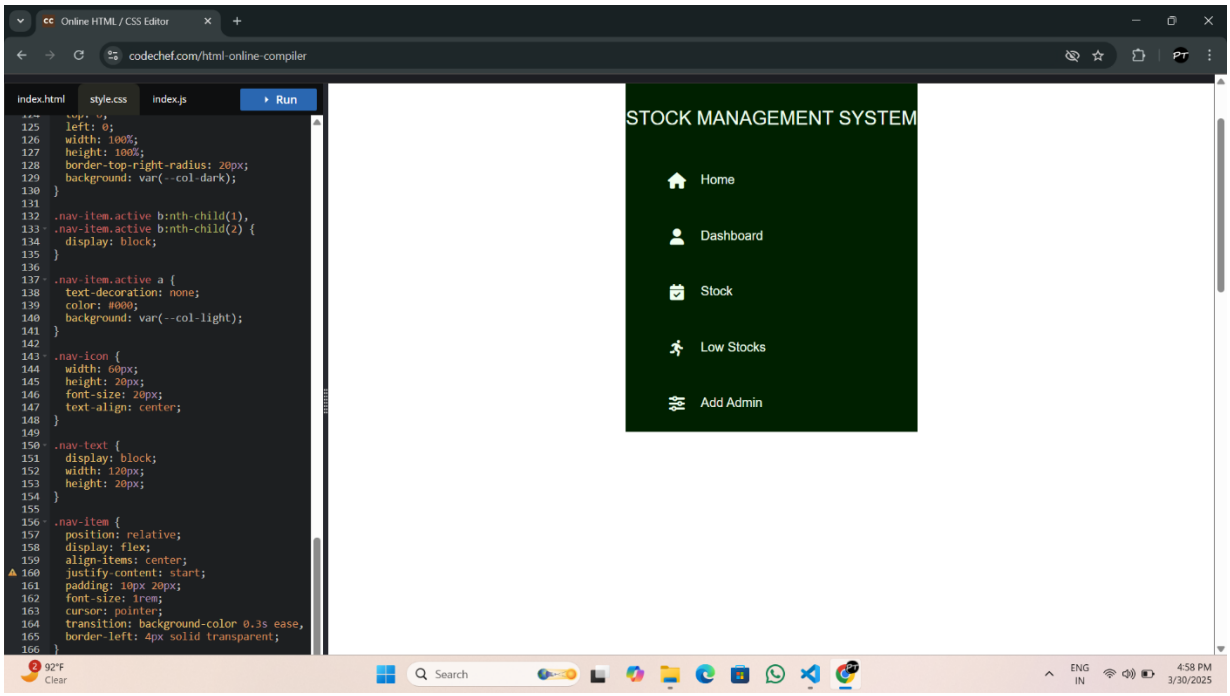
CODING SECTION:

- **Navbar.html:**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Stock Management System</title>
</head>

<body>
  <nav class="main-menu" th:fragment="nav">
    <link rel="stylesheet" href="css/navbar.css" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" />
    <h1>STOCK MANAGEMENT SYSTEM</h1>
    <img class="logo"
      src="https://ibex.et/uploads/service/stock-03_1722504513.jpg"
      alt="" / width="300px" height="200px">
    <ul>
      <li class="nav-item">
        <a data-th-href="@{/}">
          <i class="fa fa-house nav-icon"></i>
          <span class="nav-text">Home</span>
        </a>
      </li>
```

```
 <li class="nav-item">
  <a href="javascript:void(0);" onclick="handleDashboardClick()">
   <i class="fa fa-user nav-icon"></i>
   <span class="nav-text">Dashboard</span>
  </a>
 </li>

 <li class="nav-item">
  <a data-th-href="@{/viewstock}">
   <i class="fa fa-calendar-check nav-icon"></i>
   <span class="nav-text">Stock</span>
  </a>
 </li>

 <li class="nav-item">
  <a data-th-href="@{/low-stock}">
   <i class="fa fa-person-running nav-icon"></i>
   <span class="nav-text">Low Stocks</span>
  </a>
 </li>

 <li class="nav-item">
  <a data-th-href="@{/addadmin}">
   <i class="fa fa-sliders nav-icon"></i>
   <span class="nav-text">Add Admin</span>
  </a>
 </li>
</ul>
```

```
  <script src="js/navbar.js"></script>
 </nav>
</body>


</html>
```



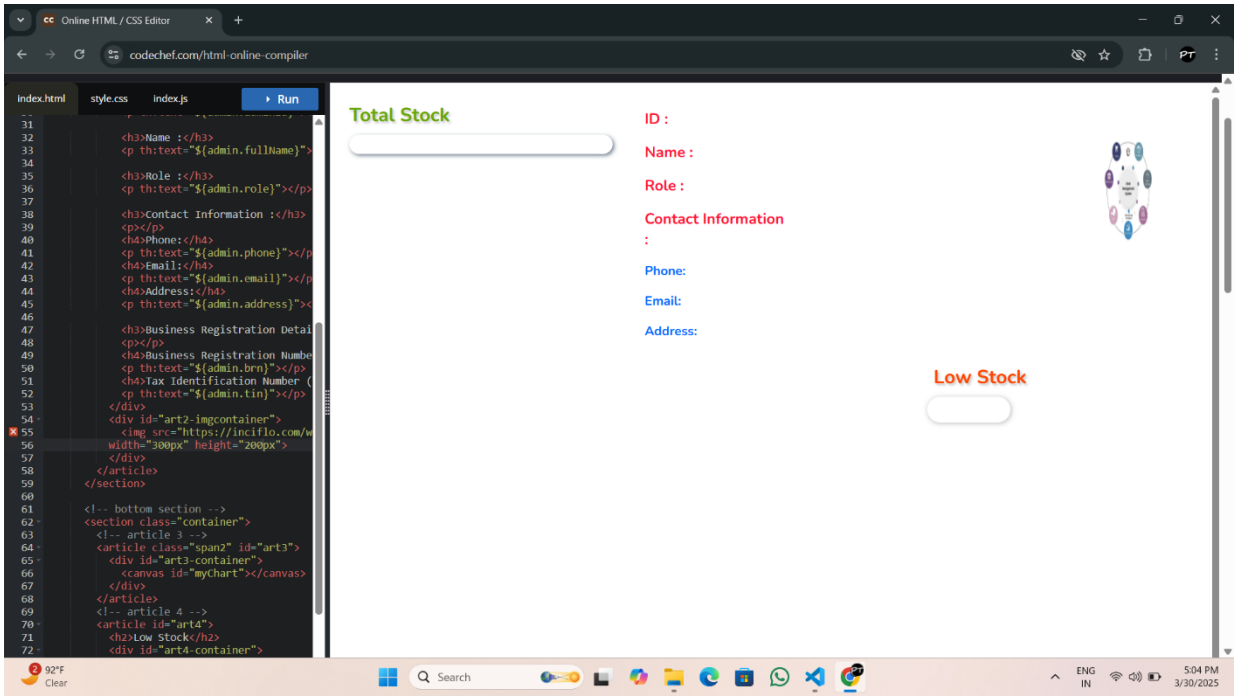
Dash.html:

```
<!DOCTYPE html>
<html lang="en">


<head>
 <meta charset="UTF-8" />
 <title>STOCK Management SYSTEM</title>


 <link rel="stylesheet" href="css/dash.css" />
</head>
```

```
<body>
  <main>
    <nav th:replace="Navbar/navbar.html::nav"></nav>
    <!--  -->
    <section id="main-content">
      <section class="container">
        <article id="art1">
          <h2>Total Stock</h2>
          <div id="art1-container">
            <div class="tot-stock" data-th-each="stock : ${stock}">
              <p data-th-text="${stock.itemName}"></p>
              <span data-th-text="|Qty: ${stock.quantity}|"></span>
            </div>
          </div>
        </article>
        <!-- article 2 -->
        <article class="span2" id="art2">
          <div id="personal-detail">
            <h3>ID :</h3>
            <p th:text="${admin.adminId}"></p>

            <h3>Name :</h3>
            <p th:text="${admin.fullName}"></p>

            <h3>Role :</h3>
            <p th:text="${admin.role}"></p>

            <h3>Contact Information :</h3>
            <p></p>
```

```
          <h4>Phone:</h4>
          <p th:text="${admin.phone}"></p>
          <h4>Email:</h4>
          <p th:text="${admin.email}"></p>
          <h4>Address:</h4>
          <p th:text="${admin.address}"></p>

          <h3>Business Registration Details :</h3>
          <p></p>
          <h4>Business Registration Number:</h4>
          <p th:text="${admin.brn}"></p>
          <h4>Tax Identification Number (TIN):</h4>
          <p th:text="${admin.tin}"></p>
        </div>
        <div id="art2-imgcontainer">
          <img src="https://inciflo.com/wp-content/uploads/2024/06/inciflo-blogs-1.jpg"
alt="Invencio Image" /
          width="300px" height="200px">
        </div>
      </article>
    </section>

    <!-- bottom section -->
    <section class="container">
      <!-- article 3 -->
      <article class="span2" id="art3">
        <div id="art3-container">
          <canvas id="myChart"></canvas>
        </div>
      </article>
```

```
      <!-- article 4 -->
      <article id="art4">
        <h2>Low Stock</h2>
        <div id="art4-container">
          <div class="low-stock" data-th-each="lowStock : ${lowStock}">
            <p data-th-text="${lowStock.itemName}"></p>
            <span data-th-text="|${lowStock.quantity}|"></span>
          </div>
        </div>
      </article>
    </section>
  </section>
 </main>

 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
 <script src="js/chart.js"></script>
</body>

<script src="js/dash.js"></script>

</html>
```

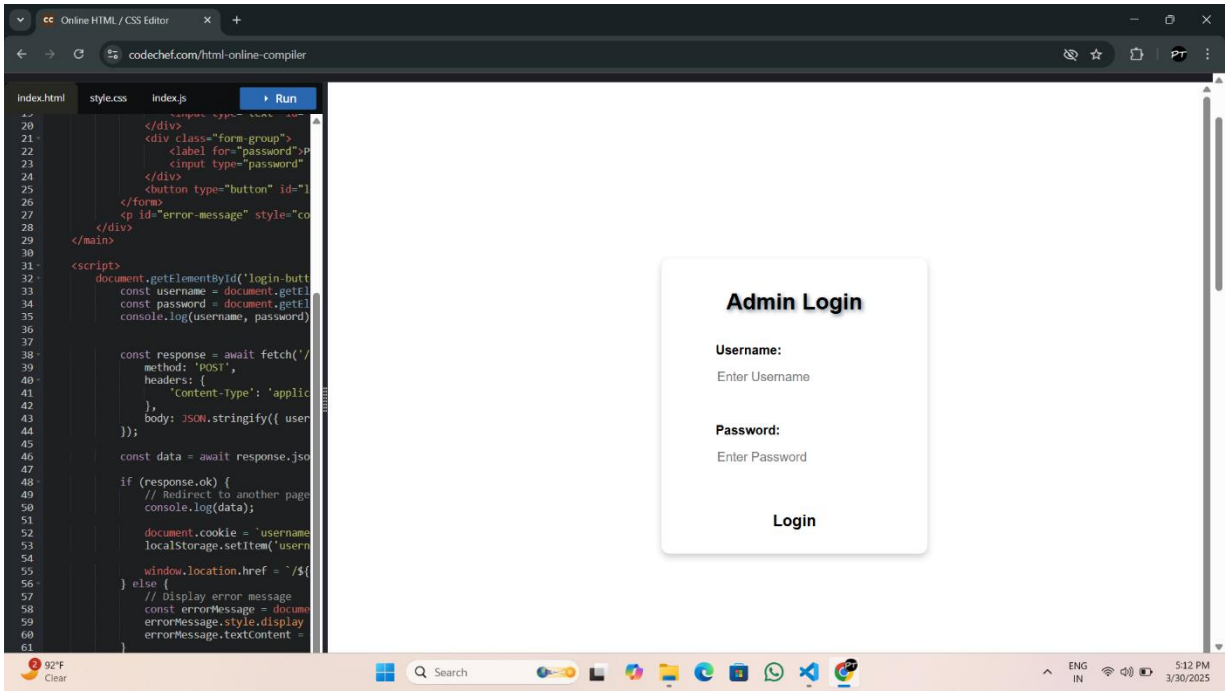


LOGIN.HTML:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Admin Login</title>
    <link rel="stylesheet" href="css/adminLogin.css" />
</head>

<body>
    <main>
        <nav th:replace="Navbar/navbar.html::nav"></nav>
        <div class="container">
```

```
        <h1>Admin Login</h1>
        <form id="admin-login-form">
          <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" placeholder="Enter
Username" required />
          </div>
          <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" placeholder="Enter
Password" required />
          </div>
          <button type="button" id="login-button" class="submit-btn
span2">Login</button>
        </form>
        <p id="error-message" style="color: red; display: none;"></p>
      </div>
    </main>

    <script>
      document.getElementById('login-button').addEventListener('click', async () => {
        const username = document.getElementById('username').value;
        const password = document.getElementById('password').value;
        console.log(username, password);


        const response = await fetch('/login', {
          method: 'POST',
          headers: {
            'Content-Type': 'application/json',
```

```
            },
            body: JSON.stringify({ username, password }),
        });

        const data = await response.json();

        if (response.ok) {
            // Redirect to another page on successful login
            console.log(data);

            document.cookie = `username=${data.userId}`;
            localStorage.setItem('username', data.userId);

            window.location.href = `/${data.userId}`;
        } else {
            // Display error message
            const errorMessage = document.getElementById('error-message');
            errorMessage.style.display = 'block';
            errorMessage.textContent = data.message || 'Invalid username or password';
        }
      });
    </script>
</body>

</html>
```



Admin.html:

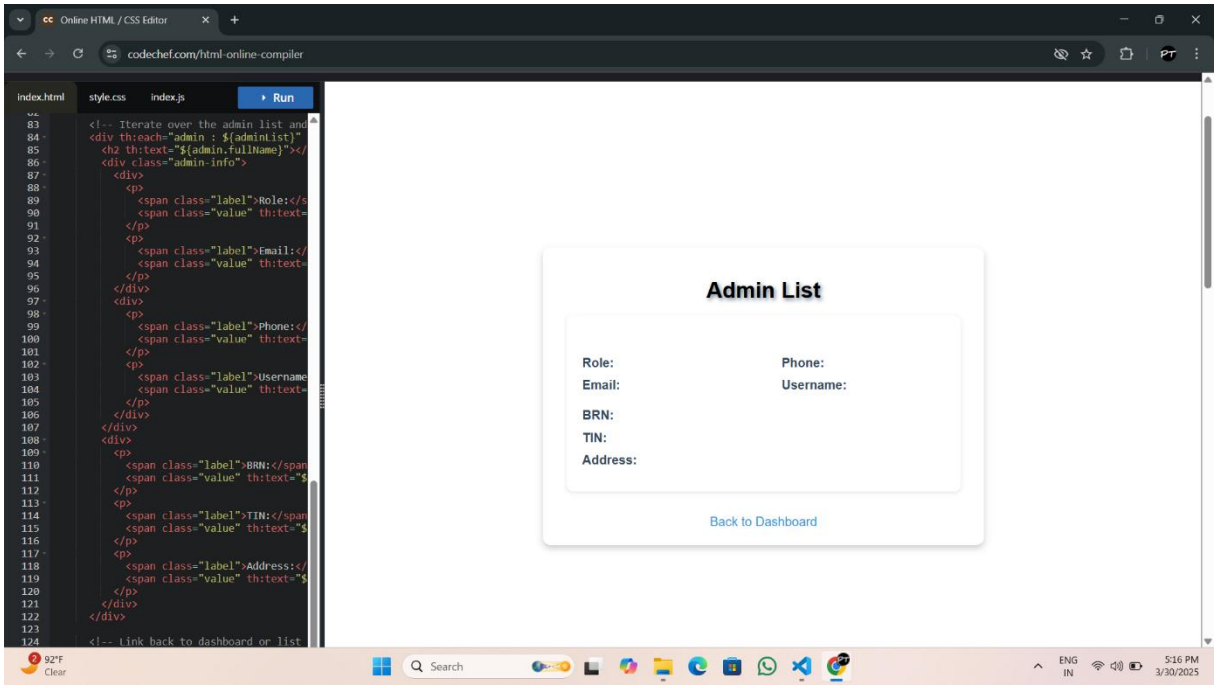
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Admin List</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #f4f7f6;
        margin: 0;
        padding: 0;
      }

      .container {
        max-width: 900px;
```

```
  margin: 0 auto;
  padding: 20px;
}

h1 {
  text-align: center;
  color: #333;
}

.admin-card {
  background-color: #ffffff;
  padding: 20px;
  margin-bottom: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.admin-card h2 {
  margin-bottom: 15px;
  color: #2c3e50;
  font-size: 1.6rem;
}

.admin-card p {
  font-size: 1rem;
  margin: 10px 0;
  color: #7f8c8d;
}
```

```
.label {
  font-weight: bold;
  color: #34495e;
}

.value {
  color: #2ecc71;
}

.admin-card .admin-info {
  display: flex;
  justify-content: space-between;
}

.admin-card .admin-info div {
  width: 45%;
}

.back-link {
  display: block;
  text-align: center;
  margin-top: 30px;
  font-size: 1rem;
  color: #3498db;
  text-decoration: none;
}

.back-link:hover {
  text-decoration: underline;
```

```
    }
  </style>
</head>

<body>
  <div class="container">
    <h1>Admin List</h1>

    <!-- Iterate over the admin list and display each admin in a content description format -->
    <div th:each="admin : ${adminList}" class="admin-card">
      <h2 th:text="${admin.fullName}"></h2>
      <div class="admin-info">
        <div>
          <p>
            <span class="label">Role:</span>
            <span class="value" th:text="${admin.role}"></span>
          </p>
          <p>
            <span class="label">Email:</span>
            <span class="value" th:text="${admin.email}"></span>
          </p>
        </div>
        <div>
          <p>
            <span class="label">Phone:</span>
            <span class="value" th:text="${admin.phone}"></span>
          </p>
          <p>
            <span class="label">Username:</span>
```

```
            <span class="value" th:text="${admin.username}"></span>
          </p>
        </div>
      </div>
      <div>
        <p>
          <span class="label">BRN:</span>
          <span class="value" th:text="${admin.brn}"></span>
        </p>
        <p>
          <span class="label">TIN:</span>
          <span class="value" th:text="${admin.tin}"></span>
        </p>
        <p>
          <span class="label">Address:</span>
          <span class="value" th:text="${admin.address}"></span>
        </p>
      </div>
    </div>

    <!-- Link back to dashboard or list -->
    <a href="/" class="back-link">Back to Dashboard</a>
  </div>
 </body>
</html>
```



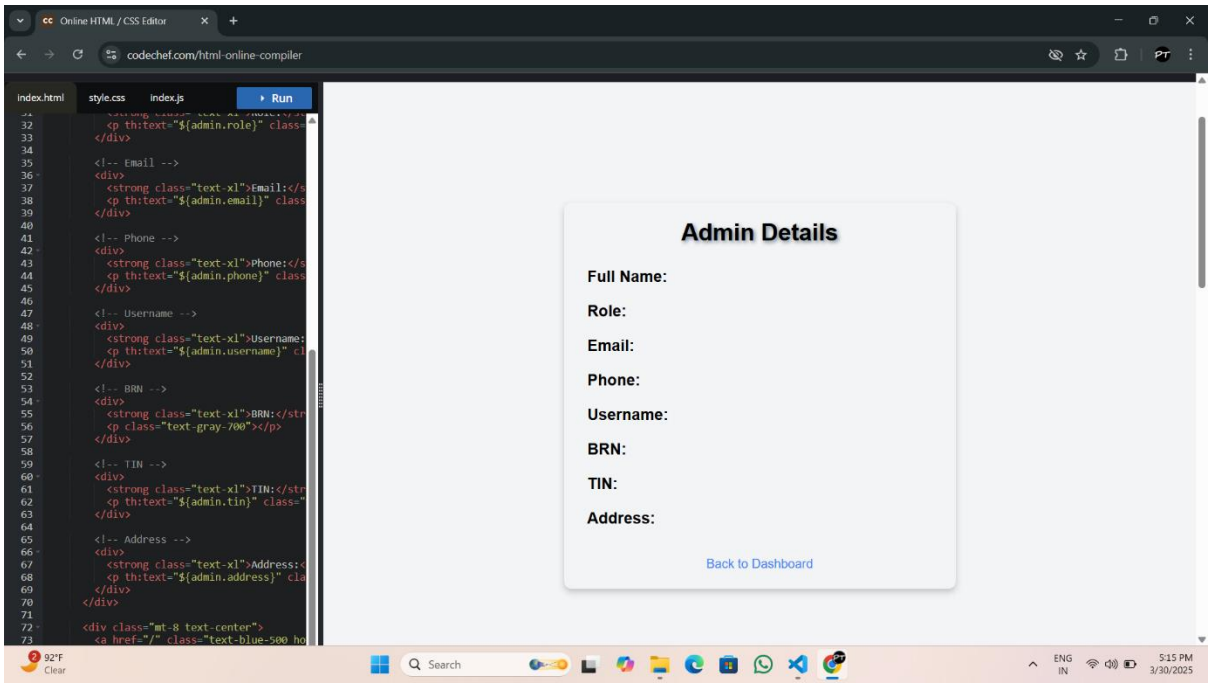
Admin-by-id.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Admin Details</title>
  <link
   rel="stylesheet"
   href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
  />
  <!-- Optional Icon Library -->
  <link
   rel="stylesheet"
   href="https://cdn.jsdelivr.net/npm/tailwindcss@2.0.0/dist/tailwind.min.css"
```

```
 />
 <!-- Tailwind CSS -->
</head>
<body class="bg-gray-100">
 <div class="container mx-auto mt-10 p-6 bg-white shadow-md rounded-lg">
  <h1 class="text-3xl font-bold text-center mb-6">Admin Details</h1>

  <div class="space-y-4">
   <!-- Full Name -->
   <div>
    <strong class="text-xl">Full Name:</strong>
    <p th:text="${admin.fullName}" class="text-gray-700"></p>
   </div>

   <!-- Role -->
   <div>
    <strong class="text-xl">Role:</strong>
    <p th:text="${admin.role}" class="text-gray-700"></p>
   </div>

   <!-- Email -->
   <div>
    <strong class="text-xl">Email:</strong>
    <p th:text="${admin.email}" class="text-gray-700"></p>
   </div>

   <!-- Phone -->
   <div>
    <strong class="text-xl">Phone:</strong>
```

```
  <p th:text="${admin.phone}" class="text-gray-700"></p>
 </div>

 <!-- Username -->
 <div>
  <strong class="text-xl">Username:</strong>
  <p th:text="${admin.username}" class="text-gray-700"></p>
 </div>

 <!-- BRN -->
 <div>
  <strong class="text-xl">BRN:</strong>
  <p class="text-gray-700"></p>
 </div>

 <!-- TIN -->
 <div>
  <strong class="text-xl">TIN:</strong>
  <p th:text="${admin.tin}" class="text-gray-700"></p>
 </div>

 <!-- Address -->
 <div>
  <strong class="text-xl">Address:</strong>
  <p th:text="${admin.address}" class="text-gray-700"></p>
 </div>
</div>

<div class="mt-8 text-center">
```

```
      <a href="/" class="text-blue-500 hover:underline">Back to Dashboard</a>
    </div>
  </div>
 </body>
</html>
```

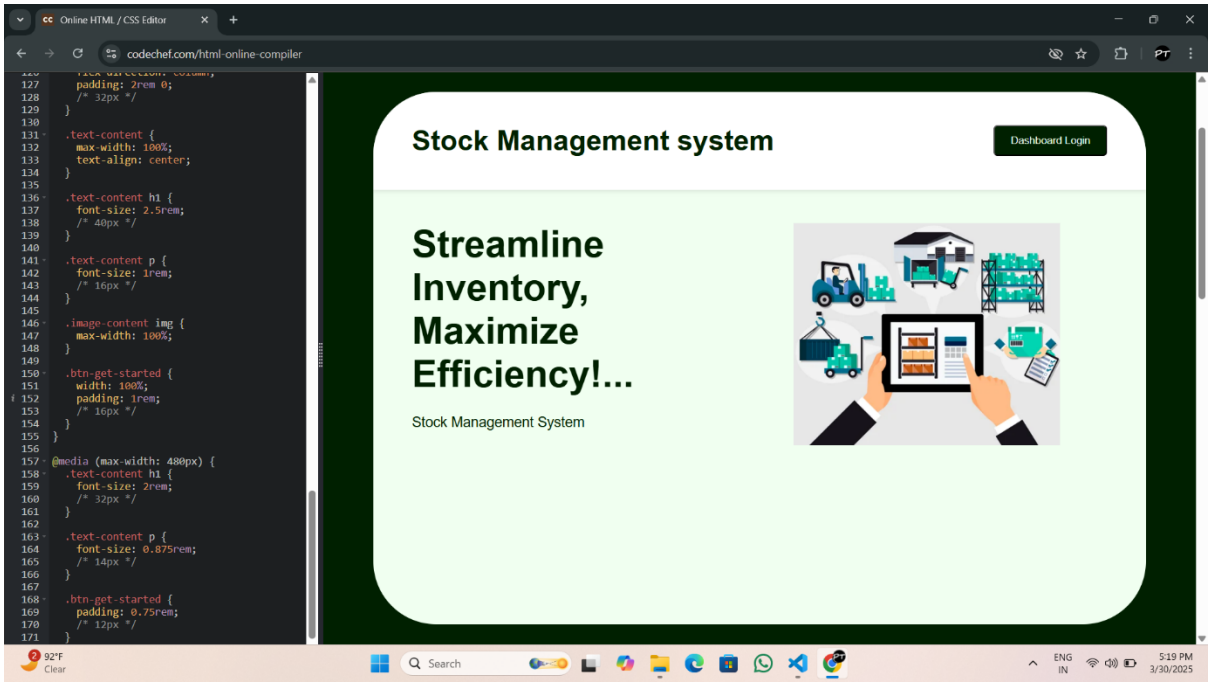


Index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Stock Management system</title>
  <link rel="stylesheet" href="css/style.css" />
</head>
```

```
<body>
  <section>
    <header class="header">
      <div class="container">
        <div class="logo">
          <h2>Stock Management system</h2>
        </div>
        <div class="nav-bar">
          <nav>
            <button onclick="window.location.href='/login'" class="btn-get-started">Dashboard Login</button>
          </nav>
        </div>
    </header>

    <main class="hero-section">
      <div class="container">
        <div class="text-content">
          <h1>Streamline Inventory, Maximize Efficiency!...</h1>
          <p>Stock Management System</p>
        </div>
        <div class="image-content">
          <img src="https://5.imimg.com/data5/SELLER/Default/2022/8/RX/UM/CL/63287060/stock-management-system-software-500x500.png" alt="Database Illustration" /
          width="400px" height="300px">
        </div>
      </div>
    </main>
```

```
  </section>
  <script src="js/script.js"></script>
</body>


</html>
```



Add-admin.html:

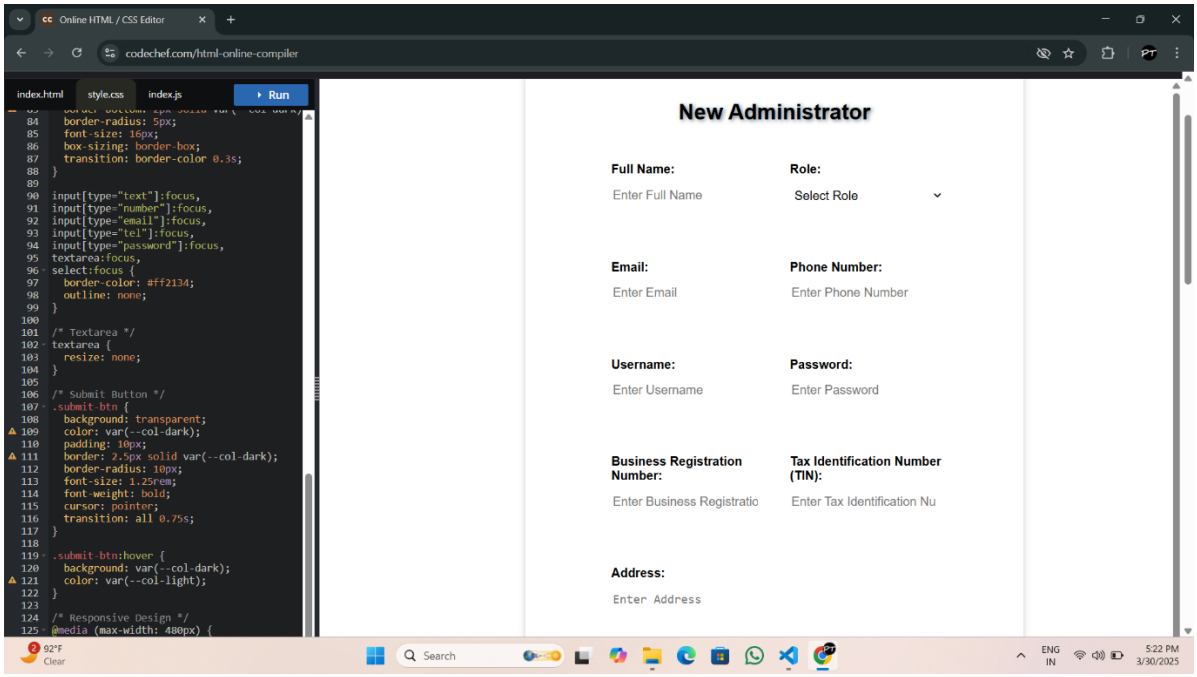
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Inventory Management - Add Stock</title>
    <link rel="stylesheet" href="css/addAdmin.css" />
  </head>
```

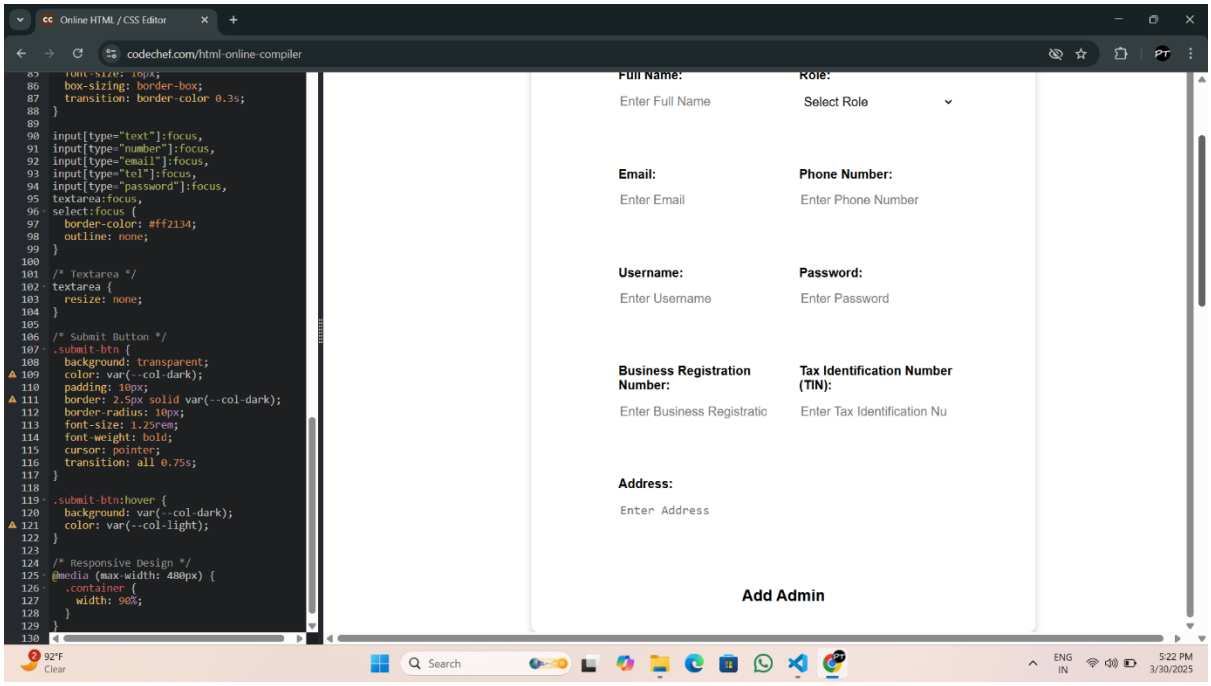
```
<body>
  <main>
    <nav th:replace="Navbar/navbar.html::nav"></nav>
    <div class="container">
      <h1>New Administrator</h1>
      <form id="add-admin-form" method="post" action="/add-admin">
        <div class="form-group">
          <label for="full-name">Full Name:</label>
          <input
            type="text"
            id="full-name"
            name="fullName"
            placeholder="Enter Full Name"
            required
          />
        </div>
        <div class="form-group">
          <label for="role">Role:</label>
          <select id="role" name="role" required>
            <option value="">Select Role</option>
            <option value="Seller">Owner</option>
            <option value="Seller">Supervisor</option>
            <option value="Admin">Admin</option>
            <option value="Seller">Seller</option>
          </select>
        </div>
        <div class="form-group">
          <label for="email">Email:</label>
          <input
```

```
    type="email"
    id="email"
    name="email"
    placeholder="Enter Email"
    required
   />
  </div>
  <div class="form-group">
   <label for="phone">Phone Number:</label>
   <input
    type="tel"
    id="phone"
    name="phone"
    placeholder="Enter Phone Number"
    required
   />
  </div>
  <div class="form-group">
   <label for="username">Username:</label>
   <input
    type="text"
    id="username"
    name="username"
    placeholder="Enter Username"
    required
   />
  </div>
  <div class="form-group">
   <label for="password">Password:</label>
```

```
  <input
    type="password"
    id="password"
    name="password"
    placeholder="Enter Password"
    required
  />
</div>
<div class="form-group">
  <label for="brn">Business Registration Number:</label>
  <input
    type="text"
    id="brn"
    name="brn"
    placeholder="Enter Business Registration Number"
    required
  />
</div>
<div class="form-group">
  <label for="tin">Tax Identification Number (TIN):</label>
  <input
    type="text"
    id="tin"
    name="tin"
    placeholder="Enter Tax Identification Number (TIN)"
    required
  />
</div>
<div class="form-group span2">
```

```
        <label for="address">Address:</label>
        <textarea
          id="address"
          name="address"
          rows="1"
          placeholder="Enter Address"
        ></textarea>
      </div>
      <button type="submit" class="submit-btn span2">Add Admin</button>
    </form>
  </div>
 </main>
</body>
</html>
```





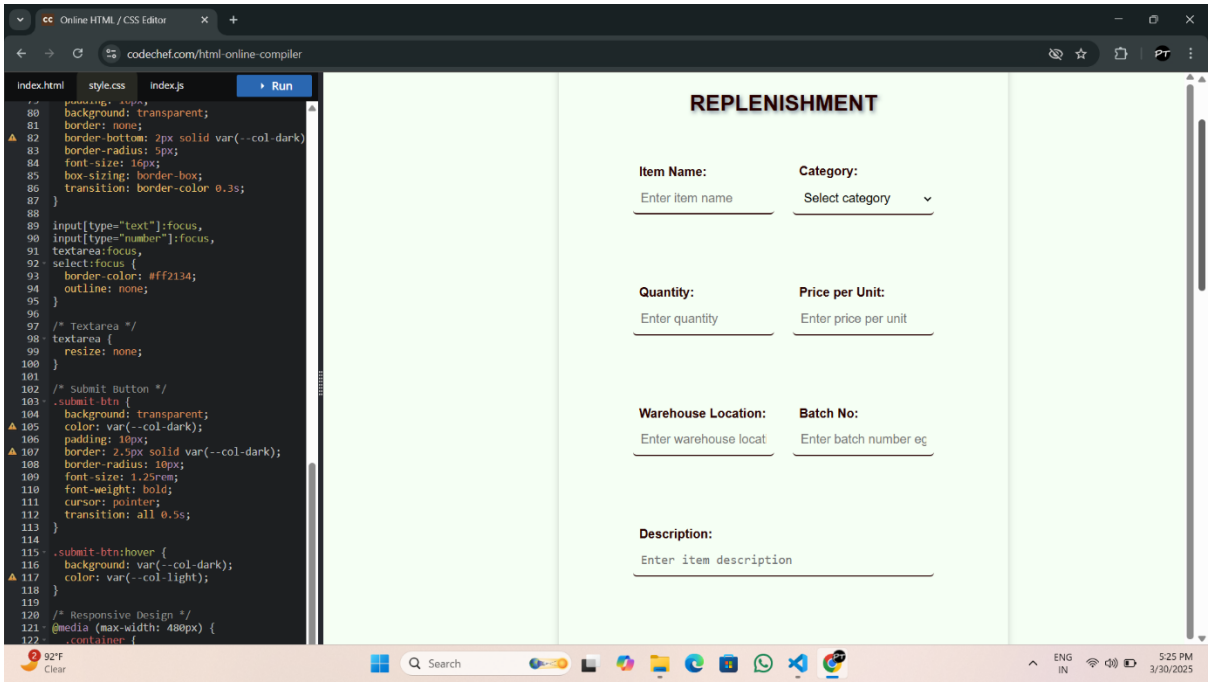
Add-stock.html:

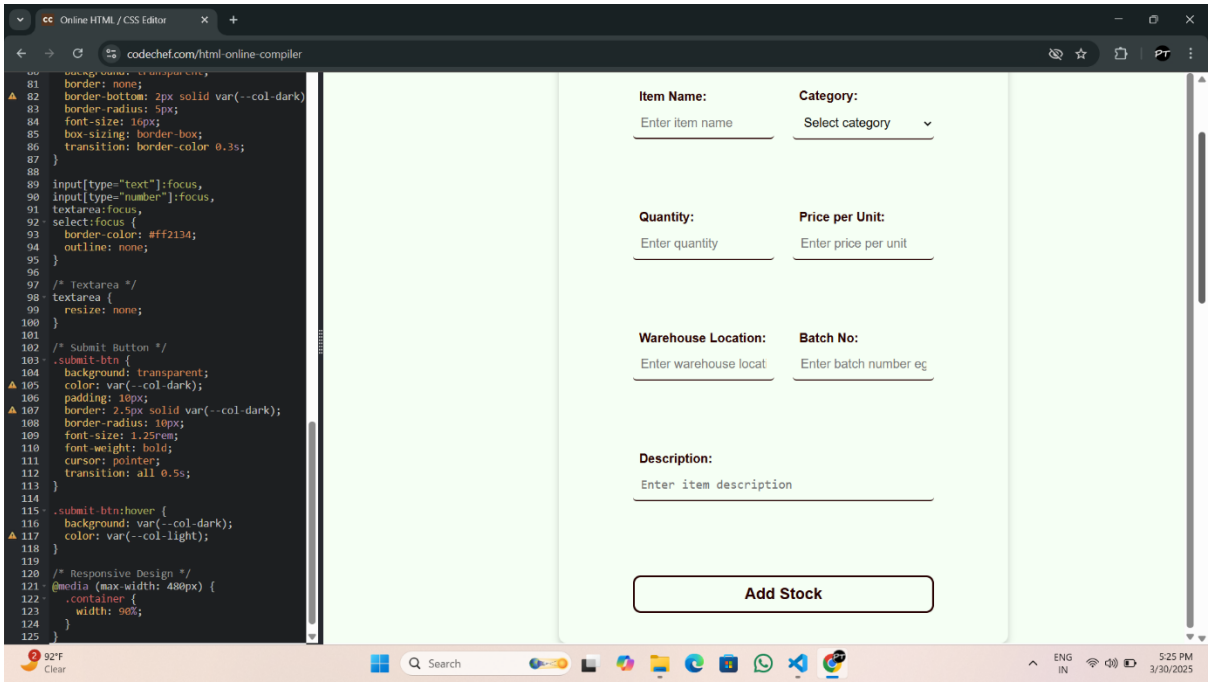
```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Inventory Management - Add Stock</title>
  <link rel="stylesheet" href="css/addStock.css" />
 </head>
 <body>
  <main>
   <nav th:replace="Navbar/navbar.html::nav"></nav>
   <div class="container">
    <h1>REPLENISHMENT</h1>
    <form id="add-stock-form" method="post" action="/add-stock">
     <div class="form-group">
      <label for="item-name">Item Name:</label>
      <input
```

```
    type="text"
    id="item-name"
    name="itemName"
    placeholder="Enter item name"
    required
   />
  </div>
  <div class="form-group">
   <label for="category">Category:</label>
   <select id="category" name="category" required>
    <option value="">Select category</option>
    <option value="electronics">Electronics</option>
    <option value="apparel">Apparel</option>
    <option value="home-appliances">Home Appliances</option>
    <option value="groceries">Groceries</option>
   </select>
  </div>
  <div class="form-group">
   <label for="quantity">Quantity:</label>
   <input
    type="number"
    id="quantity"
    name="quantity"
    placeholder="Enter quantity"
    required
   />
  </div>
  <div class="form-group">
   <label for="price">Price per Unit:</label>
```

```
  <input
    type="number"
    id="price"
    name="price"
    placeholder="Enter price per unit"
    required
  />
</div>
<div class="form-group">
  <label for="warehouse">Warehouse Location:</label>
  <input
    type="text"
    id="warehouse"
    name="warehouse"
    placeholder="Enter warehouse location"
    required
  />
</div>
<div class="form-group">
  <label for="batch no">Batch No:</label>
  <input
    type="text"
    id="batch no"
    name="batchNo"
    placeholder="Enter batch number eg-BT36"
    required
  />
</div>
<div class="form-group span2">
```

```
        <label for="description">Description:</label>
        <textarea
          id="description"
          name="description"
          rows="1"
          placeholder="Enter item description"
        ></textarea>
      </div>
      <button type="submit" class="submit-btn span2">Add Stock</button>
    </form>
  </div>
 </main>
</body>
</html>
```





Product.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>GSAP Product Price Range Slider</title>
  <link rel="stylesheet" href="css/product.css" />
</head>

<body>
  <main>
    <div th:replace="Navbar/navbar.html::nav"></div>
    <section>
```

```
<header class="header">
  <div class="logo">STOCK MANAGEMENT SYSTEM</div>
  <div class="search-bar">
    <form action="/viewstock-search" method="post">
      <input type="text" placeholder="Search your item" name="searchTerm" />
      <button type="submit">Search</button>
    </form>
  </div>
</header>
<!--  -->
<section id="main-body">
  <div class="product-list">
    <div data-th-each="stock : ${response}" class="product">
      <div class="img" data-th-if="${stock.category == 'electronics'}">
        <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcRbO2NXhNpOJYPftTAH0haXoYla0gos255EQw&s"
alt="Product Image" /
        WIDTH="300PX" HEIGHT="200PX">
      </div>
      <div class="img" data-th-if="${stock.category == 'apparel'}">
        <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcRmdWFG80FeHB8de8itrpdTr7N7Fv0jJAq0qg&s"
alt="Product Image" /
        WIDTH="300PX" HEIGHT="200PX">
      </div>
      <div class="img" data-th-if="${stock.category == 'home-appliances'}">
        <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQM0uGz7yhhzViACxb8YhJKycGFYFR4bvy_TQ&s"
alt="Product Image" /
        width="300px" height="200px">
      </div>
```

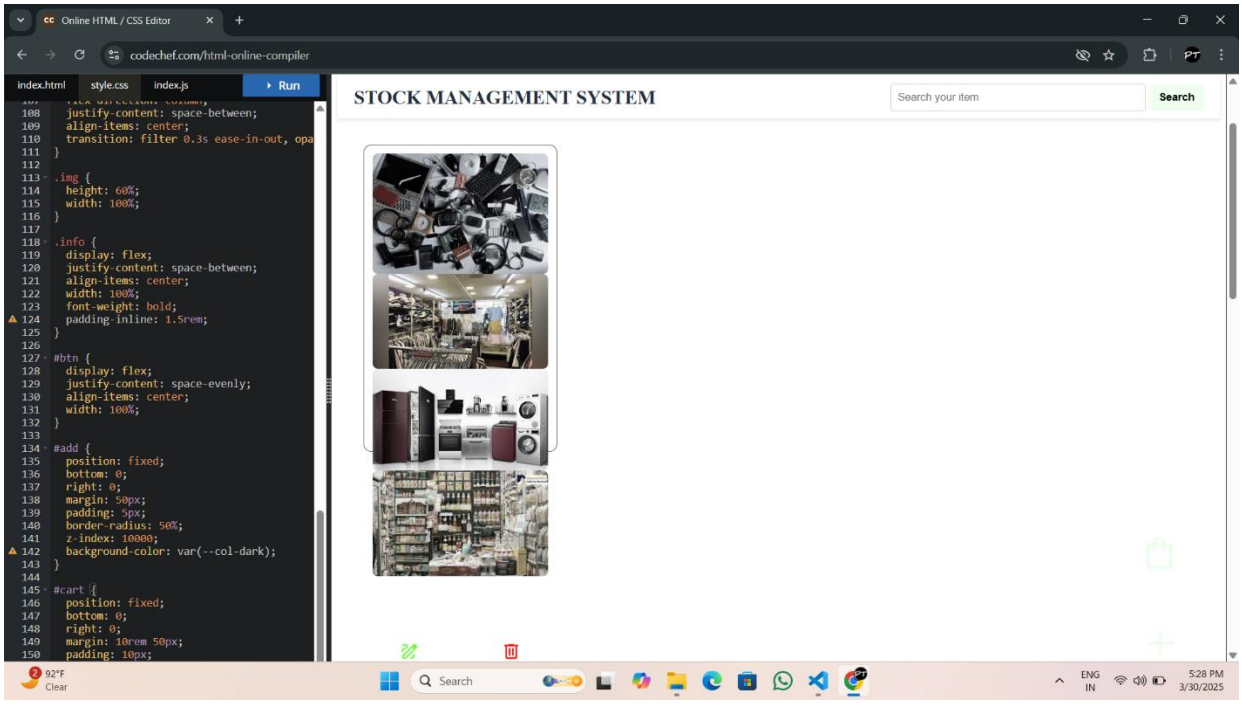
```
<div class="img" data-th-if="${stock.category == 'groceries'}">
  <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSmzOeH-pHO_iZ7QTVh7pq37VD-7GtN5WSJLw&s" alt="Product Image" /
  width="300px" height="200px">
</div>
<div class="info">
  <h3 data-th-text="${stock.itemName}"></h3>
  <p data-th-text="|₹ ${stock.price}|"></p>
</div>
<div class="info">
  <h4 data-th-text="${stock.warehouse}"></h4>
  <p data-th-text="|Qty: ${stock.quantity}|"></p>
</div>
<div id="btn">
  <a data-th-href="|/update-stock?stockId=${stock.stockId}|" class="btn">
    <svg xmlns="http://www.w3.org/2000/svg" height="25px" viewBox="0 -960 960 960" width="25px"
      fill="#75FB4C">
      <path
        d="M160-120v-170l527-526q12-12 27-18t30-6q16 0 30.5 6t25.5 18l56 56q12 11 18 25.5t6 30.5q0 15-6 30t-18 27L330-120H160Zm80-80h56l393-392-28-29-29-28-392 393v56Zm560-503-57-57 57 57Zm-139 82-29-28 57 57-28-29ZM560-120q74 0 137-37t63-103q0-36-19-62t-51-45l-59 59q23 10 36 22t13 26q0 23-36.5 41.5T560-200q-17 0-28.5 11.5T520-160q0 17 11.5 28.5T560-120ZM183-426l60-60q-20-8-31.5-16.5T200-520q0-12 18-24t76-37q88-38 117-69t29-70q0-55-44-87.5T280-840q-45 0-80.5 16T145-785q-11 13-9 29t15 26q13 11 29 9t27-13q14-14 31-20t42-6q41 0 60.5 12t19.5 28q0 14-17.5 25.5T262-654q-80 35-111 63.5T120-520q0 32 17 54.5t46 39.5Z" />
    </svg>
  </a>
  <button type="button" data-th-onclick="addToCart('[[${stock.stockId}]]')">
    <svg xmlns="http://www.w3.org/2000/svg" height="24px" viewBox="0 -960 960 960" width="24px"
      fill="#ffffff">
```

```
            <path
              d="m480-560-56-56 63-64H320v-80h167l-64-64 57-56 160 160-160 160ZM280-80q-33 0-56.5-23.5T200-160q0-33 23.5-56.5T280-240q33 0 56.5 23.5T360-160q0 33-23.5 56.5T280-80Zm400 0q-33 0-56.5-23.5T600-160q0-33 23.5-56.5T680-240q33 0 56.5 23.5T760-160q0 33-23.5 56.5T680-80ZM40-800v-80h131l170 360h280l156-280h91L692-482q-11 20-29.5 31T622-440H324l-44 80h480v80H280q-45 0-68.5-39t-1.5-79l54-98-144-304H40Z" />
            </svg>
          </button>
          <script>
            function addToCart(stockId) {
              console.log(stockId);
              const cleanedStockId = stockId.replace(/"/g, '');
              console.log(cleanedStockId);
              const quantity = prompt("Enter quantity:");

              if (quantity && !isNaN(quantity)) {
                fetch('http://localhost:8081/add-to-cart', {
                  method: 'POST',
                  headers: {
                    'Content-Type': 'application/x-www-form-urlencoded',
                  },
                  body: new URLSearchParams({
                    stockId: cleanedStockId,
                    quantity: quantity,
                  }),
                })
                  .then(response => {
                    if (response.ok) {
                      return response.json();
                    } else {
```

```
          return response.text().then(err => {
            throw new Error(err);
          });
        }
      })
      .then((stockDetails) => {
        const cart = JSON.parse(localStorage.getItem("cart")) || [];
        const existingItemIndex = cart.findIndex((item) => item.stockId === stockId);

        if (existingItemIndex !== -1) {
          cart[existingItemIndex].quantity += parseInt(quantity);
        } else {
          cart.push({
            ...stockDetails,
            quantity: parseInt(quantity),
            discount: stockDetails.discount || 0, // Ensure a default discount value
          });
        }

        localStorage.setItem("cart", JSON.stringify(cart));
        alert("Item added to cart successfully!");
      })
      .catch((error) => {
        console.error("Error adding item to cart:", error);
      });
    }
  }

</script>
```

```
          <a data-th-href="|/delete-stock?stockId=${stock.stockId}|" class="btn">
            <svg xmlns="http://www.w3.org/2000/svg" height="25px" viewBox="0 -960 960
960" width="25px"
              fill="rgb(255, 18, 18)">
              <path
                d="M280-120q-33 0-56.5-23.5T200-200v-520h-40v-80h200v-
40h240v40h200v80h-40v520q0 33-23.5 56.5T680-120H280Zm400-600H280v520h400v-
520ZM360-280h80v-360h-80v360Zm160 0h80v-360h-80v360ZM280-720v520-520Z" />
            </svg>
          </a>
        </div>
      </div>
    </div>
  </section>
  <aside>
    <section>
      <a data-th-href="|/addstock|">
        <button id="add" type="submit">
          <svg xmlns="http://www.w3.org/2000/svg" height="55px" viewBox="0 -960 960
960" width="55px"
            fill="#efffee">
            <path d="M450-450H200v-60h250v-250h60v250h250v60H510v250h-60v-250Z" />
          </svg>
        </button>
      </a>
    </section>
    <section>
      <a data-th-href="|/cart|">
        <button id="cart" type="submit">
```

```
          <svg xmlns="http://www.w3.org/2000/svg" height="48px" viewBox="0 -960 960
960" width="48px"
            fill="#efffee">
            <path
              d="M240-80q-33 0-56.5-23.5T160-160v-480q0-33 23.5-56.5T240-720h80q0-66
47-113t113-47q66 0 113 47t47 113h80q33 0 56.5 23.5T800-640v480q0 33-23.5 56.5T720-
80H240Zm0-80h480v-480h-80v80q0 17-11.5 28.5T600-520q-17 0-28.5-11.5T560-560v-
80H400v80q0 17-11.5 28.5T360-520q-17 0-28.5-11.5T320-560v-80h-80v480Zm160-
560h160q0-33-23.5-56.5T480-800q-33 0-56.5 23.5T400-720ZM240-160v-480 480Z" />
          </svg>
        </button>
      </a>
    </section>
  </aside>
 </section>
</main>

<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.9.1/gsap.min.js"></script>
<script src="js/product.js"></script>
</body>

</html>
```

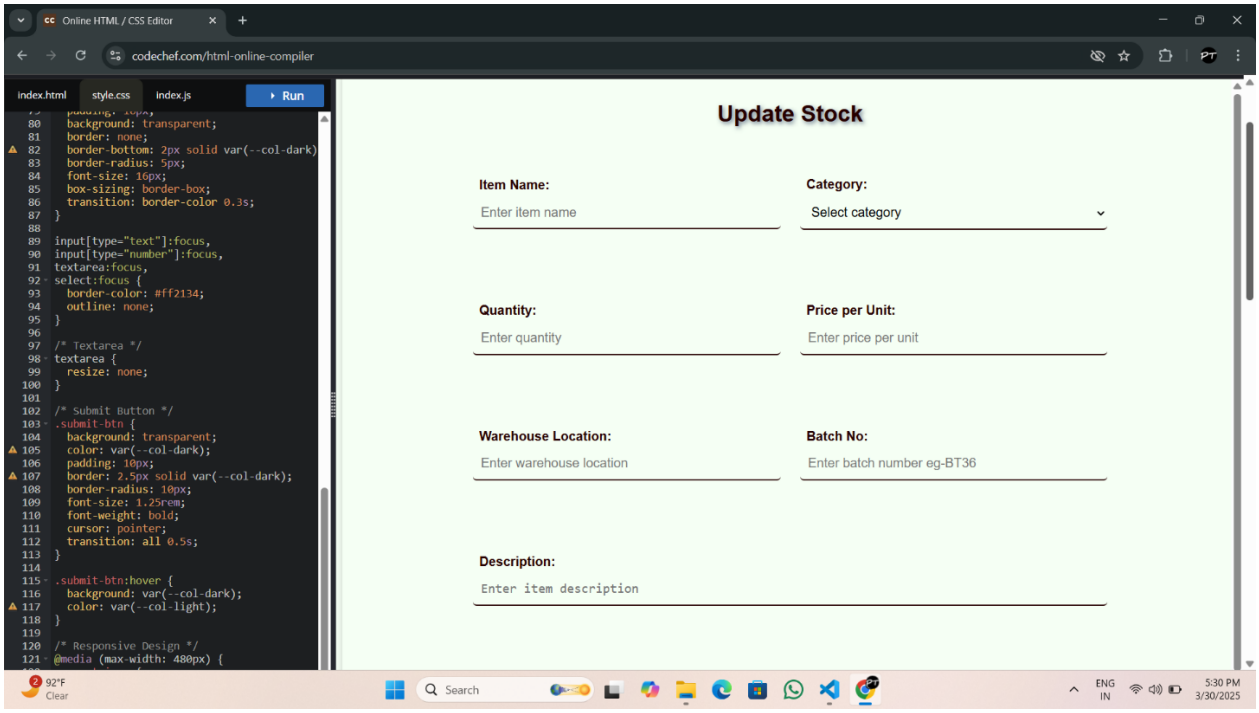


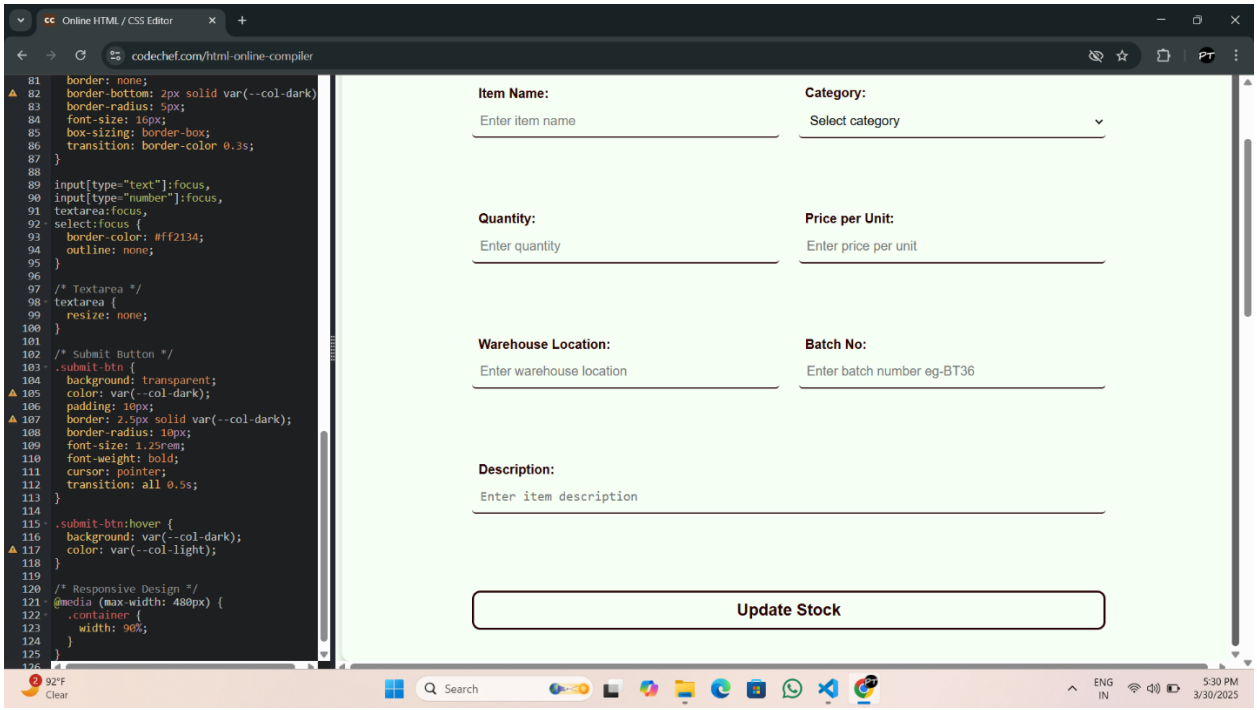
Update-product.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Inventory Management - Add Stock</title>
    <link rel="stylesheet" href="css/addStock.css" />
</head>
```

```
<body>
  <div class="container" data-th-object="${stock}">
    <h1>Update Stock</h1>
    <form id="add-stock-form" method="post" action="/update-stock">
      <!-- <input type="hidden" name="_method" value="put" /> -->
      <input type="hidden" name="stockId" data-th-field="*{stockId}" />
      <div class="form-group">
        <label for="item-name">Item Name:</label>
        <input type="text" id="item-name" name="itemName" placeholder="Enter item
name" required
          data-th-field="*{itemName}" />
      </div>
      <div class="form-group">
        <label for="category">Category:</label>
        <select id="category" name="category" required data-th-field="*{category}">
          <option value="">Select category</option>
          <option value="electronics">Electronics</option>
          <option value="apparel">Apparel</option>
          <option value="home-appliances">Home Appliances</option>
          <option value="groceries">Groceries</option>
        </select>
      </div>
      <div class="form-group">
        <label for="quantity">Quantity:</label>
        <input type="number" id="quantity" name="quantity" placeholder="Enter
quantity" required
          data-th-field="*{quantity}" />
      </div>
      <div class="form-group">
```

```
            <label for="price">Price per Unit:</label>
            <input type="number" id="price" name="price" placeholder="Enter price per unit" required
                data-th-field="*{price}" />
        </div>
        <div class="form-group">
            <label for="warehouse">Warehouse Location:</label>
            <input type="text" id="warehouse" name="warehouse" placeholder="Enter warehouse location" required
                data-th-field="*{warehouse}" />
        </div>
        <div class="form-group">
            <label for="batch no">Batch No:</label>
            <input type="text" id="batch no" name="batchNo" placeholder="Enter batch number eg-BT36" required
                data-th-field="*{batchNo}" />
        </div>
        <div class="form-group span2">
            <label for="description">Description:</label>
            <textarea id="description" name="description" rows="1" placeholder="Enter item description"
                data-th-field="*{description}"></textarea>
        </div>
        <button type="submit" class="submit-btn span2">Update Stock</button>
    </form>
  </div>
</body>

</html>
```


Online HTML / CSS Editor
codechef.com/html-online-compiler
index.html
style.css
index.js
Run
Update Stock
Item Name:
Enter item name
Category:
Select category
Quantity:
Enter quantity
Price per Unit:
Enter price per unit
Warehouse Location:
Enter warehouse location
Batch No:
Enter batch number eg-BT36
Description:
Enter item description


Online HTML / CSS Editor
codechef.com/html-online-compiler
Item Name:
Enter item name
Category:
Select category
Quantity:
Enter quantity
Price per Unit:
Enter price per unit
Warehouse Location:
Enter warehouse location
Batch No:
Enter batch number eg-BT36
Description:
Enter item description
Update Stock

Cart.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Cart Page - Invencio</title>
  <link rel="stylesheet" href="css/cart.css" />
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.28/jspdf.plugin.autotable.min.js"></script>


</head>

<body>
  <!-- Header Section -->
  <header class="header">
    <div class="logo">STOCK MANAGEMENT SYSTEM</div>
    <div class="search-bar">
      <input type="text" placeholder="Enter Customer Name" />
      <input type="date" placeholder="Enter Date" />
    </div>
  </header>

  <!-- Main Content -->
  <main class="main-content">
    <div class="cart-container">
      <!-- Cart Items -->
```

```
    <div class="cart-items">
      <h2>Shopping Cart</h2>
      <div id="cart-items-container">
        <!-- Cart items will be dynamically added here -->
      </div>
    </div>

    <!-- Cart Summary -->
    <div class="cart-summary">
      <h2>Order Summary</h2>
      <div class="summary-details">
        <p>Subtotal (<span id="cart-item-count">0</span> items):</p>
        <p class="subtotal" id="cart-subtotal">₹0</p>
      </div>
      <button class="checkout-btn" onclick="checkout()">Proceed to Checkout</button>
    </div>
  </div>
</main>

<script>
  // Function to load cart items from local storage
  function loadCart() {
    const cart = JSON.parse(localStorage.getItem("cart")) || [];
    const cartItemsContainer = document.getElementById("cart-items-container");
    const cartItemCount = document.getElementById("cart-item-count");
    const cartSubtotal = document.getElementById("cart-subtotal");

    // Clear current content
    cartItemsContainer.innerHTML = "";
```

```
let subtotal = 0;
let itemCount = 0;

// Populate cart items
cart.forEach((item, index) => {
 if (!item.discount) item.discount = 0; // Ensure discount property exists

 let productImage = "";
 switch (item.category) {
  case 'electronics':
   productImage = "images/electronic.png";
   break;
  case 'apparel':
   productImage = "images/apparel.png";
   break;
  case 'home-appliances':
   productImage = "images/home-appliances.png";
   break;
  case 'groceries':
   productImage = "images/groceries.png";
   break;
  default:
   productImage = "images/default.png"; // default image if no category matches
 }

 const itemElement = document.createElement("div");
 itemElement.className = "cart-item";
 itemElement.innerHTML = `
```

```
        <img src="${productImage}" alt="Product Image" />
        <div class="item-details">
          <h3>${item.stockName}</h3>
          <p class="stock">In Stock</p>
          <div class="item-actions">
            <button onclick="removeCartItem(${index})">Remove</button>
          </div>
        </div>
        <div class="item-quantity">
          <select onchange="updateQuantity(${index}, this.value)">
            ${[...Array(1500).keys()].map(qty => qty + 1).concat(item.quantity > 5 ? item.quantity : []).map(
        qty => `<option value="${qty}" ${qty === item.quantity ? "selected" : ""}>${qty}</option>`
      ).join("")}
          </select>
          <input type="number" min="0" max="100" value="${item.discount}"
                oninput="updateDiscount(${index}, this.value)" placeholder="Discount %" />
                <p>₹${(item.price)}</p>
                <p>₹${(item.price * item.quantity * (1 - item.discount / 100)).toFixed(2)}</p>
        </div>
      `;

      cartItemsContainer.appendChild(itemElement);

      // Calculate subtotal
      subtotal += item.price * item.quantity * (1 - item.discount / 100);
      itemCount += item.quantity;
    });
```

```
    // Update cart summary
    cartItemCount.textContent = itemCount;
    cartSubtotal.textContent = `₹${subtotal.toFixed(2)}`;
  }

  // Function to remove a cart item
  function removeCartItem(index) {
    const cart = JSON.parse(localStorage.getItem("cart")) || [];
    cart.splice(index, 1);
    localStorage.setItem("cart", JSON.stringify(cart));
    loadCart();
  }

  // Function to update item quantity
  function updateQuantity(index, quantity) {
    const cart = JSON.parse(localStorage.getItem("cart")) || [];
    cart[index].quantity = parseInt(quantity);
    localStorage.setItem("cart", JSON.stringify(cart));
    loadCart();
  }

  // Function to update discount for an item
  function updateDiscount(index, discount) {
    const cart = JSON.parse(localStorage.getItem("cart")) || [];
    cart[index].discount = Math.min(100, Math.max(0, parseFloat(discount))); // Ensure valid
discount
    localStorage.setItem("cart", JSON.stringify(cart));
    loadCart();
  }
```

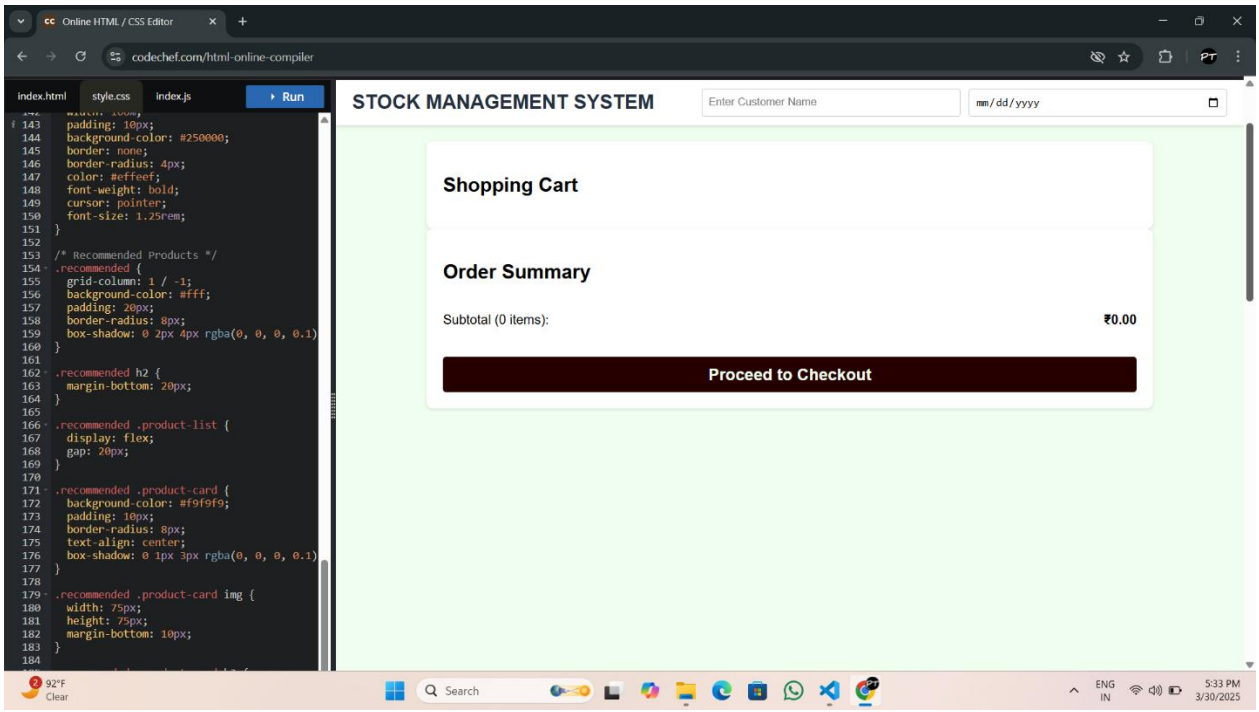
```
  // Function to generate a unique invoice ID
  function generateInvoiceId() {
    const now = new Date();

    const year = now.getFullYear();
    const month = String(now.getMonth() + 1).padStart(2, '0'); // Months are 0-indexed, so add 1
    const day = String(now.getDate()).padStart(2, '0');
    const hours = String(now.getHours()).padStart(2, '0');
    const minutes = String(now.getMinutes()).padStart(2, '0');
    const seconds = String(now.getSeconds()).padStart(2, '0');
    const milliseconds = String(now.getMilliseconds()).padStart(3, '0');

    // Create a unique invoice ID in the format: YYYYMMDDHHMMSSSSS
    const invoiceId = `${year}${month}${day}${hours}${minutes}${seconds}${milliseconds}`;

    return invoiceId;
  }

  // Function to handle checkout
  async function checkout() {
    const { jsPDF } = window.jspdf;
    const pdf = new jsPDF();

    const cart = JSON.parse(localStorage.getItem("cart")) || [];
    const customerName = document.querySelector(".search-bar input[type='text']").value.trim();
    const orderDate = document.querySelector(".search-bar input[type='date']").value;

    if (!customerName || !orderDate) {
```

```
  alert("Please enter customer name and date.");
  return;
}

if (cart.length === 0) {
  alert("Your cart is empty!");
  return;
}

// Generate the invoice ID
const invoiceId = generateInvoiceId();

// Header
pdf.setFontSize(16);
pdf.text(`Invoice ID: ${invoiceId}`, 14, 20);  // Displaying Invoice ID
pdf.text(`Customer Name - ${customerName}`, 14, 30);
pdf.setFontSize(12);
pdf.text(`Date: ${orderDate}`, 14, 40);

// Table headers
const headers = ["Item No", "Name", "Price", "Qty", "Discount", "Total"];

// Table data
let subtotal = 0;
const data = cart.map((item, index) => {
  const discountedPrice = item.price * item.quantity * (1 - item.discount / 100);
  subtotal += discountedPrice;
  return [
    index + 1,
```

```
    item.stockName,
    `Rs.${parseFloat(item.price).toFixed(2)}`,
    item.quantity,
    `${item.discount}%`,
    `Rs.${discountedPrice.toFixed(2)}`
  ];
});

// Add table using autoTable
pdf.autoTable({
  startY: 50,
  head: [headers],
  body: data,
  theme: "grid",
  styles: { halign: "center" },
  headStyles: { fillColor: [0, 0, 0] },
});

// Total Amount
pdf.setFontSize(14);
pdf.text(`Total Amount: Rs.${subtotal.toFixed(2)}`, 14, pdf.lastAutoTable.finalY + 10);

// Generate PDF filename
const fileName = `${customerName}_${invoiceId}.pdf`;

// Save PDF
pdf.save(fileName);

// After saving, update stock and clear cart
```

```
    try {
      const response = await fetch("http://localhost:8081/update-stock-quantity", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify(cart),
      });

      if (response.ok) {
        alert("Checkout completed successfully!");
        localStorage.removeItem("cart");
        loadCart();
        window.location.href = "http://localhost:8081/viewstock";
      } else {
        const errorMessage = await response.text();
        alert(`Failed to update stock: ${errorMessage}`);
      }
    } catch (error) {
      console.error("Error during checkout:", error);
      alert("An unexpected error occurred. Please try again later.");
    }
  }


  // Load cart items when the page loads
  window.onload = loadCart;
 </script>

</body>
```

</html>


Online HTML / CSS Editor
codechef.com/html-online-compiler
index.html
style.css
index.js
Run
STOCK MANAGEMENT SYSTEM
Enter Customer Name
mm/dd/yyyy
Shopping Cart
Order Summary
Subtotal (0 items):
₹0.00
Proceed to Checkout
Search
ENG
IN
5:33 PM
3/30/2025

CHAPTER NINE

TESTING

Introduction

The goal of software testing is to evaluate software program capabilities or attributes to ensure that it meets the application standards. Testing does not guarantee quality, and its objective is not to detect bugs. Testing might consist of verification and validation as well as dependability estimation. The fundamental goal of testing is to find errors in application. The most crucial function of testing is to provide information. Check ensure that the program is functioning properly while creating, updating, and deleting product entries.

Unit Testing

Individual software components are tested in this manner. It is usually done by the programmer rather than the testers. This demands detailed information and expertise of the internal program design and code. We performed numerous testing tasks during unit testing, such as the reflection of the unit data on the database and its interface. Several sorts of bugs linked with the component were discovered and resolved. To test our software, we use a variety of functional keys. Our software unit testing is concerned with stock units, opening stock units, and product unit validation.

CHAPTER TEN

CONCLUSION

The implementation of a PHP and MySQL-based sales and stock management system for a small retail store can provide significant benefits over the traditional method of using Excel spreadsheets. By having real-time data and capacity to work with POS systems, recommended system (this project) will give us more accurate record of sales and stock easily. Increasing demands of the many stores can be satisfied by this system as it is versatile and reliable. This system can be turned into a well-defined process by building a database, developing a user interface, developing PHP code, merging with POS system, regular testing, and proper deployments and carrying out regular upgrades and maintenance. Replacement of automated and interconnected solution by manual processes the retail store can make superior decisions based on real time. Thus making the store more productive and successful.

Project Limitation

In this project there are certain limitations. We were unable to meet all of our goals due to a lack of understanding in specific disciplines and a lack of time when the project began. We hope these constraints are significant. Some of the project's constraints are as follows:

1. This application is not appropriate for organizations with a high volume.
2. Products and various warehouse levels.
3. This software application can only generate rudimentary reports.
4. It only has a single admin panel.
5. It is not ideal for huge organizations.

CHAPTER ELEVEN

FUTURE ENHANCEMENTS

Because we began this project with little knowledge of the inventory management system, we learned about enhancement capacity while creating it. Some of the areas we can expand for betterment and effectiveness are given below:

1. Design of an interactive user interface.
2. Manage Inventory by stores
3. Oracle is used as the database.
4. An online payment mechanism can be included.
5. The system should be adaptable to any situation.
6. For product returns sales and purchase system should be added.
7. Lost and broken items.

REFERENCES

Websites & Online Resources

- IBM Cloud Docs – Stock Management Solutions: https://www.ibm.com/cloud
- Oracle Inventory Management: https://www.oracle.com/scm
- Microsoft Azure Supply Chain Solutions: https://azure.microsoft.com/en-us/solutions/supply-chain/

Open-Source & Technical References

- GitHub Open-Source Stock Management Projects: https://github.com/topics/inventory-management
- Python for Inventory Management: https://realpython.com
- MySQL Documentation for Database Management: https://dev.mysql.com/doc/