ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY, KOTTAYAM

PROJECT REPORT ON

INVENTORY MANAGEMENT SYSTEM

In partial fulfillment of the requirements for the

Award of the degree of

B Voc SOFTWARE DEVELOPMENT

By

ANSU VARGHESE

III B Voc Software Development

Register No: VB22SWD008

Under the guidance of

Ms. JANEENA SHAJU
Asst. Professor

Department of Computer Applications

2022-2025

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY, KOTTAYAM

PROJECT REPORT ON

INVENTORY MANAGEMENT SYSTEM

In partial fulfillment of the requirements for the

Award of the degree of

B Voc SOFTWARE DEVELOPMENT

By

ANSU VARGHESE

III B Voc Software Development

Register No: VB22SWD008

Under the guidance of

Ms. JANEENA SHAJU
Asst. Professor

Department of Computer Applications

2022-25

CERTIFICATE

This is to certify that the project report on **INVENTORY MANAGEMENT SYSTEM** is a bonafide record of the work done by **ANSU VARGHESE(VB22SWD008)** during the year 2022-2025 and submitted in partial fulfillment of the requirement for the degree of **B.Voc Software Development** under Mahatma Gandhi University.

Submitted for end semester exam held on ……

Head of Department

Internal examiner **External examiner**

Date:

DECLARATION

I, **ANSU VARGHESE(Register no: VB22SWD008), BVoc Software Development** final year student of **St. Teresa's College (Autonomous), Ernakulam**, hereby declare that the project submitted named **INVENTORY MANAGEMENT SYSTEM** for the **Bachelors of Vocation Degree in Software Development** is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

ACKNOWLEDGEMENT

First and foremost, I would like to thank God almighty for the successful completion of my project. I express my sincere thanks to **Provincial Superior and Manager Rev. Dr. Sr. Vinitha CSST, Rev. Sr. Emeline CSST and Principal Dr. Alphonsa Vijaya Joseph of St. Teresa's college** (**AUTONOMOUS**) for giving me an opportunity to undertake this project. I express my sincere gratitude towards the **Head of the department Ms. SHEEBA EMMANUEL** and, I would like to extend my heartfelt appreciation to **Ms. JANEENA SHAJU** (**Asst. Prof)**, my project guide for her constant support which helped in the successful completion of my project. I'm grateful to all the faculties of the Department of Computer Applications for their unwavering support and guidance throughout this journey. Finally, I extend my sincere thanks to my parents and friends and all those who directly or indirectly contributed to the realization of this project.

ANSU VARGHESE

SYNOPSIS

The Inventory Management System (IMS) is developed to efficiently manage and track the various assets and resources required for the smooth functioning of college operations. With the growing demands of managing large inventories across multiple departments, the IMS aims to simplify the process by automating inventory tracking, improving resource allocation, and reducing discrepancies. By integrating real-time data updates, automated alerts, and detailed reporting features, the system will provide administrators and staff with accurate, up-to-date information, enabling informed decision-making and effective resource management. This report provides an overview of the system's design, functionality, and the anticipated benefits it offers in terms of enhancing inventory control, reducing operational inefficiencies, and supporting the overall academic and administrative processes within the college.

CONTENTS

1. INTRODUCTION

1. INTRODUCTION

The **Inventory Management System (IMS)** aims to automate and streamline the process of monitoring and managing the different assets and resources within departments of St. Teresa's College. With this system, the college seeks to enhance efficiency in operations, better allocation resources, and obtain correct, real-time inventory usage data.

The Inventory Management System (IMS) aims to automate and streamline the process of monitoring and managing the different assets and resources within departments of St. Teresa's College. With this system, the college seeks to enhance efficiency in operations, better allocate resources, and obtain correct, real-time inventory usage data.

With an increasing requirement to track and document pieces of furniture like desks, benches, chairs, computer equipment, and lab instruments, the IMS aims to substitute existing manual methods of record-keeping with a more scaleable and trusted digital alternative. The system is designed to provide functionalities like automatic stock updates, replenishment alert facilities, and full reporting functionality.

This project report describes the objectives, system design, functionalities, and expected benefits of the IMS. It also indicates how the system enhances academic and administrative processes and promotes transparency and informed decision-making.

1.1 ABOUT PROJECT

The IMS resolves the inventory management challenges by launching an easy-to-use platform that makes asset tracking easier. The features are:

Real-Time Tracking: Provide real-time availability of the inventory items.
Department-Wise Allocation: Permit departments to view and update the status of their respective assets.
Data Security: Secure inventory data against unauthorized access through safe login procedures.
Automated Reporting: Create timely reports to aid in decision making.

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

System Analysis is an in-depth examination of the operations carried out by the system and their interaction between the modules of the system. This phase comprises the examination of the parent system and determination of the system objectives. The key purpose of this phase is collecting required information and applying the structured tool for analysis. This also comprises designing the system. For this project, the requirements are examined in-depth and information are gathered and recorded.

2.2 EXISTING SYSTEM

The present inventory control system is much dependent on manual means, i.e., on paper and spreadsheet. This system takes a long time and involves possibilities of mistakes.

LIMITATIONS OF EXISTING SYSTEM:

1. Present system does not update inventory quantities in real time
2. Relying on manual books causes chances of incorrect data input
3. Creating detailed budget, audit, or procurement reports takes a long time and hard work
4. Difficulty in Dealing with Huge Inventories

2.3 PROPOSED SYSTEM

The Inventory Management System (IMS) optimizes efficiency through the automating of asset tracking, minimizing manual errors, and offering real-time monitoring. It comprises the following notable features:

- **Seamless Asset Management**: A simple add, update, and delete asset process with an easy-to-use interface for accurate record-keeping.
- **Real-Time Inventory Monitoring**: Monitor inventory in real-time, providing real-time visibility into stock levels, asset utilization, and availability.
- **Department-Wise Asset Allocation Reports**: Create in-depth reports allocating assets according to their respective departments, enhancing responsibility and resource planning.
- **Secure Access Controls**: Enforces role-based access where administrators are the only ones who can manage sensitive information and set system options, providing security and avoiding unauthorized modifications.

This system optimizes the inventory activities, facilitates better decision-making, and makes asset management more efficient.

2.4 SYSTEM SPECIFICATION

The system specifications present a detailed outline of the technical and operational details required for effective development and deployment of the Inventory Management System (IMS) in St. Teresa's College. The specifications act as a guideline to ensure the system fulfills functional requirements, is scalable in the future, and integrates with the institution's current infrastructure without any issues.

Some of the key considerations are:

System Functionality: Establishing essential features like asset tracking, real-time inventory monitoring, reporting features, and secure access controls to make certain the IMS efficiently automates inventory management.

Scalability: Making sure the system is able to support future expansion, such as the capacity to manage growing numbers of assets, users, and departments without sacrificing performance.

Infrastructure Compatibility: Evaluating hardware and software needs to support seamless integration with the college's existing IT infrastructure, reducing deployment issues and guaranteeing data protection.

User Accessibility & Security: Enforcing role-based access control, enabling authorized staff to update inventory while limiting sensitive information to administrators.

These specifications offer a formalized framework to direct the development, deployment, and maintenance of the IMS, guaranteeing its efficiency, dependability, and long-term viability for St. Teresa's College.

2.5 OPERATING SYSTEM

Windows 11 is the latest major version of Microsoft's Windows NT operating system, released on October 5, 2021. It superseded Windows 10 (2015) and is available at no cost to any qualifying Windows 10 device with the new Windows 11 requirements. Windows 11 boasts a new user interface with a revamped Start Menu and Taskbar, enhanced touch controls, security features, and built-in widgets for easy access to information. It also optimized virtual desktops, gaming performance, and multitasking capabilities. It has several disadvantages too like incompatible hardware, reduced compatibility with legacy software, and fewer choices of customization.

2.6 LANGUAGE OR SOFTWARE PACKAGE

The Inventory Management System (IMS) is developed with a combination of HTML, CSS, and JavaScript to offer an operational and friendly interface. HTML (Hypertext Markup Language) is utilized as the core language, laying out the web pages and determining the structure of the content in the application. CSS (Cascading Style Sheets) is used to control the visual appearance, offering consistent styling and an appealing interface. JavaScript offers the dynamic features and interactivity, such as real-time updates, form validation, and responsive elements to enhance the overall user experience

2.7 HARDWARE & SOFTWARE SPECIFICATIONS

- **Operating System:**

  Client-Side: Microsoft Windows 10 or above / macOS / Linux

  Server-Side: Windows Server 2019 / Linux (Ubuntu 20.04 or CentOS)

- Database Management System (DBMS): MySQL 8.0

- **Programming Languages and Frameworks:**

  Frontend: HTML5, CSS3, JavaScript (with frameworks like React.js)

  Backend: Python, Django

- **Development Tools:**

  Integrated Development Environment (IDE): Visual Studio Code

  Local Development Environment: localhost

- **Browser Compatibility:**

  Google Chrome (latest version)

  Mozilla Firefox (latest version)

3. SYSTEM DESIGN

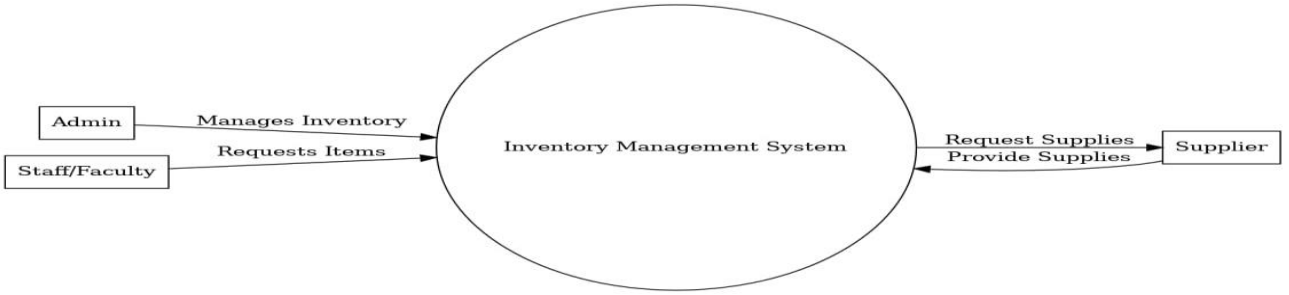
3.1 INTRODUCTION

System design is an interactive process whereby requirements are communicated to a "blueprint" for building the software first; the blueprint captures a global representation of software that's design is captured at high-level abstraction. A level traceable directly to individual data, functional and behavioral requirements. Design interaction taking place subsequent refinement resulted in design representation at a far lower level of abstraction. System design is an innovative art of conceiving and creating input, databases, offline files, method, and procedures, to process data to obtain significant full output that meet the organization objectives. During the design stage consideration to the human element, i.e., inputs to the users will have on the system. Some of the primary factors that need to be observed using the design of the system are:

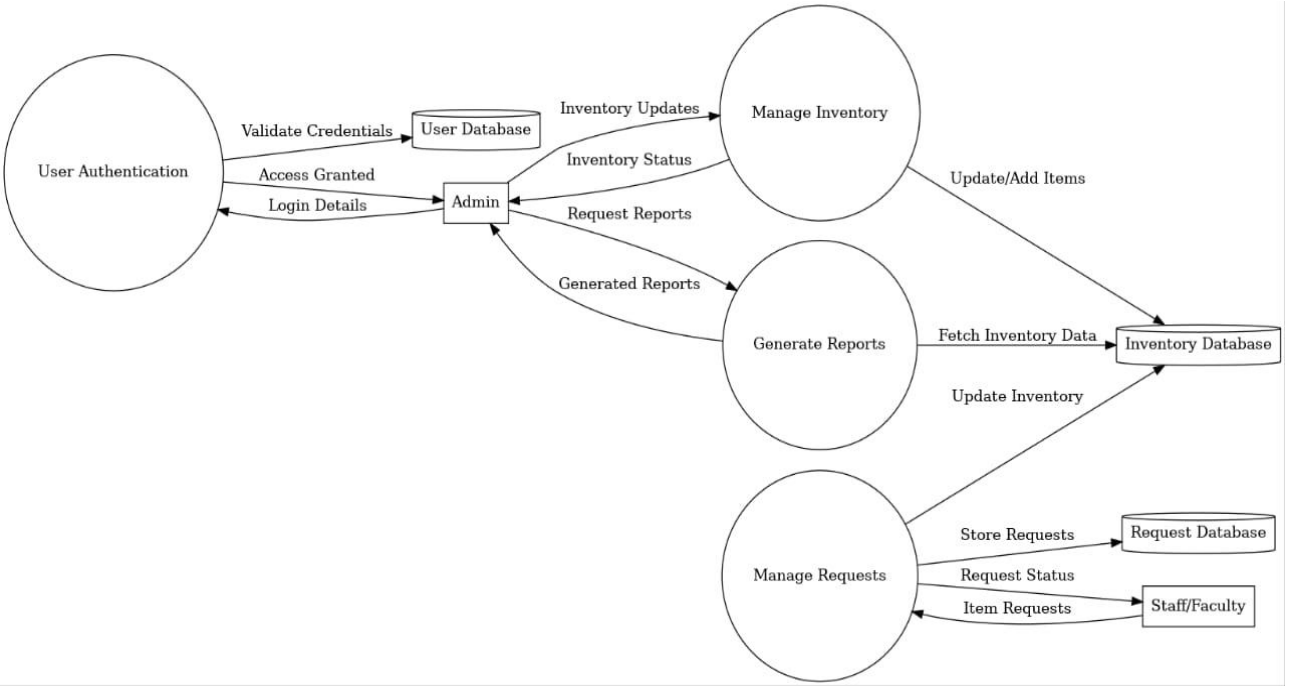
•Practicality: System should be able to be used for a long time and should have simplicity.
•Efficiency: Utilize available resources better. Efficiency includes accuracy, punctuality, and complete system output.
•Cost: Minimum cost and improved results.
•Security: Provide physical security for data.

3.2 DATA FLOW DIAGRAM

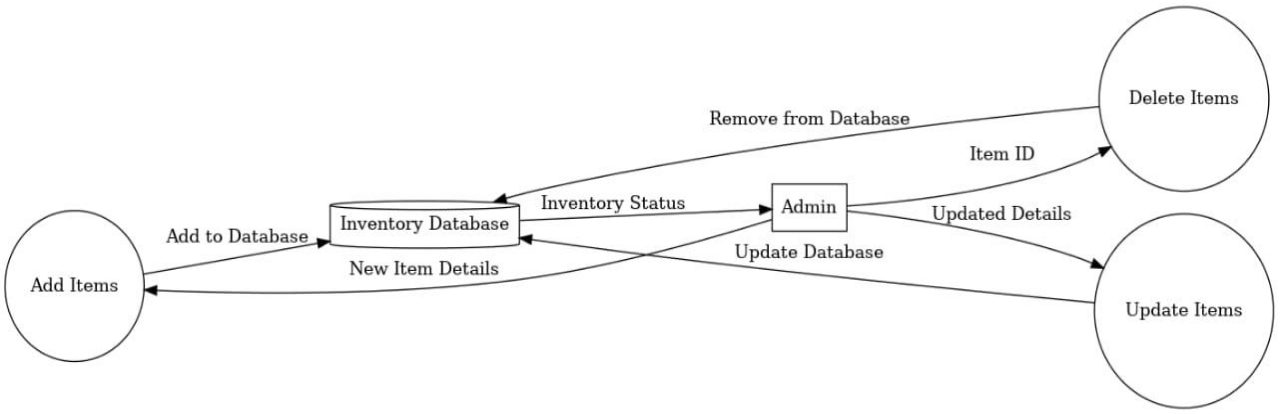
Context level diagram



Level 1 diagram



Level 2 diagram



3.3 DATABASE DESIGN

Database design, A most important part of the system design phase. In a database environment, data available are used by several users instead of each program managing its own data, authorized users share data across application with the database software managing the data as an entity. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive, and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system.

TABLE DESIGN

Table: Inventory

Column Name	Data Type	Description
item_id	INT (PK)	Unique identifier for each inventory item
item_name	VARCHAR(255)	Name of the item
category	VARCHAR(100)	Category of the item (e.g., Electronics, Furniture)
quantity	INT	Current stock quantity
unit_price	DECIMAL(10,2)	Price per unit of the item
department_id	INT (FK)	References the department using the item

Table: Departments

Column Name	Data Type	Description
department_id	INT (PK)	Unique identifier for each department
department_name	VARCHAR(255)	Name of the department

Table: Users

Column Name	Data Type	Description
user_id	INT (PK)	Unique identifier for each user
username	VARCHAR(100)	User's login name
password	VARCHAR(255)	Hashed password
role	ENUM('Admin', 'Staff', 'Faculty')	Defines user role
department_id	INT (FK)	References the department the user belongs to

Table: Order Details

Column Name	Data Type	Description
order_detail_id	INT (PK)	Unique identifier for order details
order_id	INT (FK)	References the purchase order
item_id	INT (FK)	References the item ordered
quantity	INT	Quantity of the item ordered
unit_price	DECIMAL(10,2)	Price per unit at the time of order

4.SYSTEM DEVELOPMENT

4.1 INTRODUCTION

Modular programming is a software design paradigm that involves spitting the functionality of a program into independent, interchangeable units, each containing everything required to run but one portion of the desired functionality. Conceptually, modules are a separation of concerns, and enhance maintainability by imposing logical boundaries between parts.

4.2 PROCESS DESCRIPTION

1. ADMIN Module:

- **Login**
    - Login with admin credentials for system access.
- **Manage Staff**
    - Add, update, delete staff users.
    - Assign roles and permissions to staff.
- **Manage Inventory**
    - Add, update, and delete stock items.
    - Set reorder levels and stock alerts.
- **Generate Reports**
    - Generate inventory, and stock movement reports graphically.
- **Manage Users**
    - View and manage all user roles and permissions.
- **Backup and Maintenance**
    - Perform system backups and updates.

2. STAFF Module:

- **Login**
    - Login with staff credentials.
- **View Inventory**
    - View inventory stock details and item descriptions.
- **Update Stock**
    - Update stock quantities (add/remove items).
- **Stock Receipt**
    - Record receipt of stock items from suppliers.
- **Generate Basic Reports**
    - Generate reports on inventory status, low stock, etc. graphically.

3. FACULTY / DEPARTMENT HEAD MODULE:

This module is for faculty or department heads who may need access to manage inventory for their specific departments (e.g., lab supplies, library books).

- **Login**
    - Faculty login with department-specific access.
- **View Departmental Inventory**
    - View inventory specific to their department (e.g., library, computer lab, sports equipment).

- **Request Stock**
    - Request new supplies based on departmental needs.
- **Track Usage**
    - Track how much of the inventory is being used by the department graphically.

4. REPORTS AND ANALYTICS Module

This module is designed to provide various reports to different users (Admin, Faculty, Finance, etc.) to monitor the inventory system's efficiency and effectiveness.

- **Login**
    - Login with appropriate credentials to access specific reports.
- **Generate Reports**
    - Inventory status, stock levels, and usage reports.
    - Departmental reports for faculty or administrative purposes.
- **Data Visualization**
    - Visualize data using graphs and charts for stock trends, purchase frequency, etc.

5. SYSTEM TESTING & IMPLIMENTATION

5.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represent the ultimate review of the specification, design, and coding. System testing makes a logical assumption that all parts of the system is correct; the goal will be successfully achieved.

The implementation process enables users to assume control of the system for practical application and assessment. Maintenance involves modifications to the current system, enhancement introduces additional features, and development entails the replacement of the existing system. The implementation phase is characterized by the transition from a newly designed system to an operational state.

5.2 SYSTEM IMPLEMENTATION

The implementation phase is characterized by the transition from a newly designed system to an operational state.It is the key stage in achieving a successful new system. Implementation is the stage if the project, where the theoretical design is turned into a working system. At this stage the main workload, the greatest up heal and the major impact on existing practices shift to user department. If the implementation stage is not planned and controllers carefully, it can cause chaos.

This stage of implementation constitutes a distinct project within the overall system development. It involves careful planning, investigation of the current system and its constraints on the implementation, design methods to achieve the changeover procedures, and evaluation of change over methods.

The implementation plan comprises the following steps:

- Testing the developed system with the sample data.
- Detection and correction of errors.
- Making necessary changes in the system.
- Training and involvement of user personnel.
- Installation of software utilities.

5.3 DEBUGGING

5.3.1 BLACK BOX TESTING: Black-box testing is a type of software testing in which the tester is not connected with the internal knowledge or implementation details of the software, but rather validating the functionality based on the provided specifications or requirements.

5.3.2 WHITE BOX TESTING: White box testing techniques analyze the internal structures the used data structures, internal design, code structure, and the working of the software rather than just the functionality as in black-box testing. It is alternatively referred to as glass box testing, clear box testing, or structural testing. This method is employed to evaluate the internal logic, flow, and structure of the software. The tester develops test cases to analyze the code paths and logic flows, ensuring that they align with the defined requirements.

5.3.2 SYSTEM SECURITY: As technology advances, application environment become more complex and application development security becomes more challenging. Applications, systems, and networks are constantly under various security attacks such as malicious code or denial of service. Some of the challenges from the application development security point of view includes Viruses, Trojan horses, Logic bombs, Worms, and Agent. As an addition this application is stored encrypted in the database so the user gets more reliable to its robustness. Security testing is crucial for software that handles sensitive information, as it helps to safeguard against unauthorized access by cybercriminals.

5.3.4 SCOPE FOR FUTURE ENHANCEMENT:

In future, many enhancements can be made some of them include:

- **Integration with Current Systems**: Integrate easily with the college's primary website, accounting programs, and other administrative systems for centralized data management.
- **Advanced Reporting and Analytics**: Install sophisticated analytics to create detailed reports, monitor usage patterns, and balance inventory replenishment.
- **Mobile Accessibility**: Create a mobile version or a progressive web app (PWA) to make it easily accessible to employees.
- **IoT Integration**: Use IoT devices (e.g., RFID, barcode scanners) for real-time inventory tracking and automated updates.

6. CONCLUSION

CONCLUSION

The Inventory Management System designed for our college automates inventory processes by offering an easy-to-use interface for effective tracking and management. Designed with Django, the system minimizes manual errors, maximizes resource utilization, and optimizes overall operational efficiency, meeting the college's current inventory management requirements.

The project also lays the foundation for additional future developments like integration with additional systems, richer analytics, and IoT-based real-time monitoring. Through ensuring more accountability and transparency, the system proves the capabilities of technology-solutioned implementation in streamlining institutional management functions.

7. APPENDIX

5.4 INPUT AND OUTPUT SCREEN:

ADMIN LOGIN PAGE:

Please log in to your account

Email

Enter username or email

Password

Enter Password

Sign in

ADMIN HOME PAGE:

COLLEGE INVENTORY MANAGEMENT

Logout

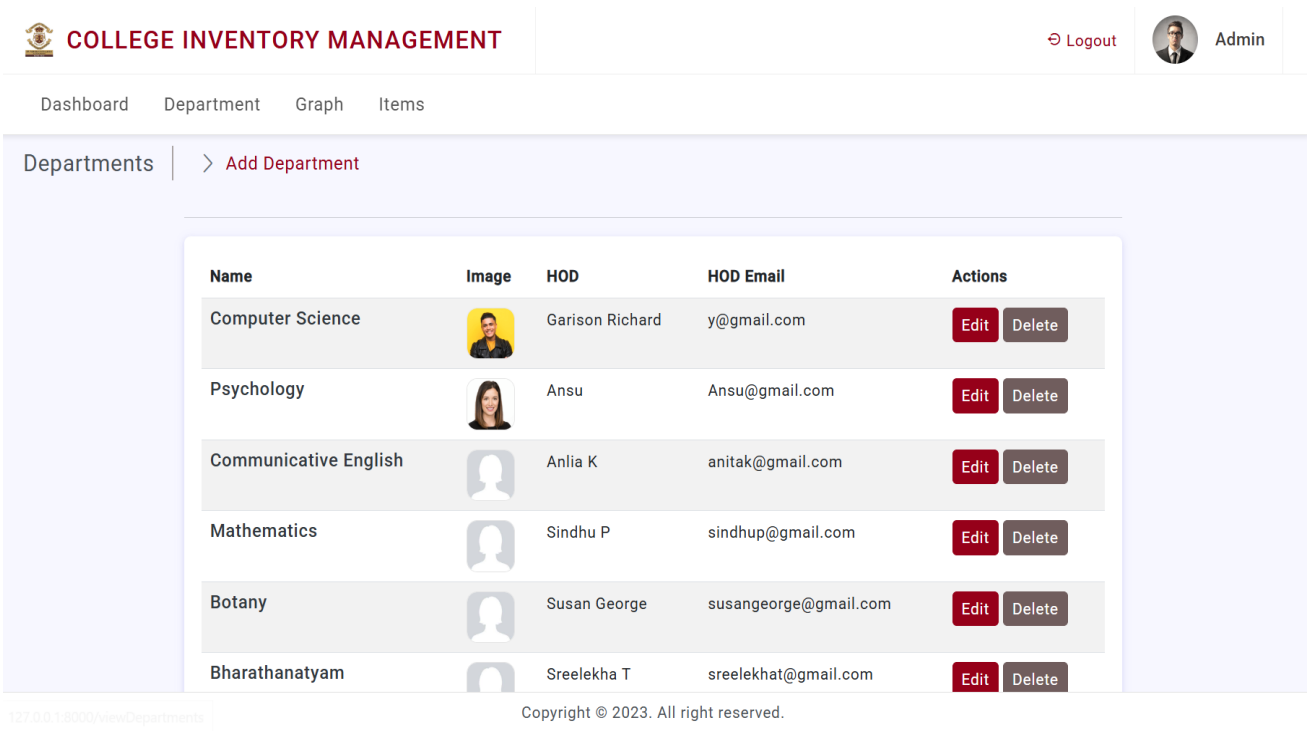
Admin

Dashboard Department Graph Items

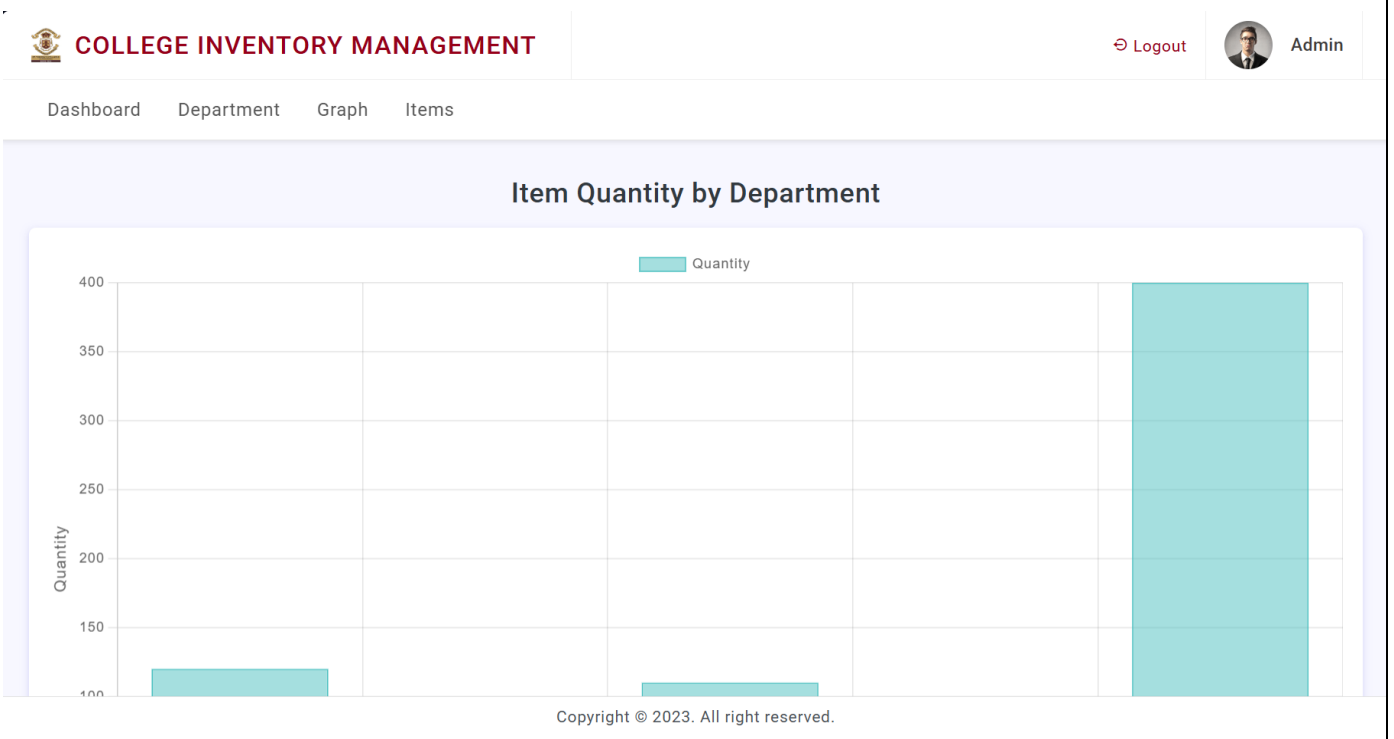
Items | Add Item

Department	Item	Image	Quantity	Price	Actions
Computer Science	Chair		120nos	200 per unit	Edit Delete
Computer Science	Bench		30nos	600 per unit	Edit Delete
Psychology	Chair		110nos	200 per unit	Edit Delete
Psychology	Table		50nos	2500 per unit	Edit Delete

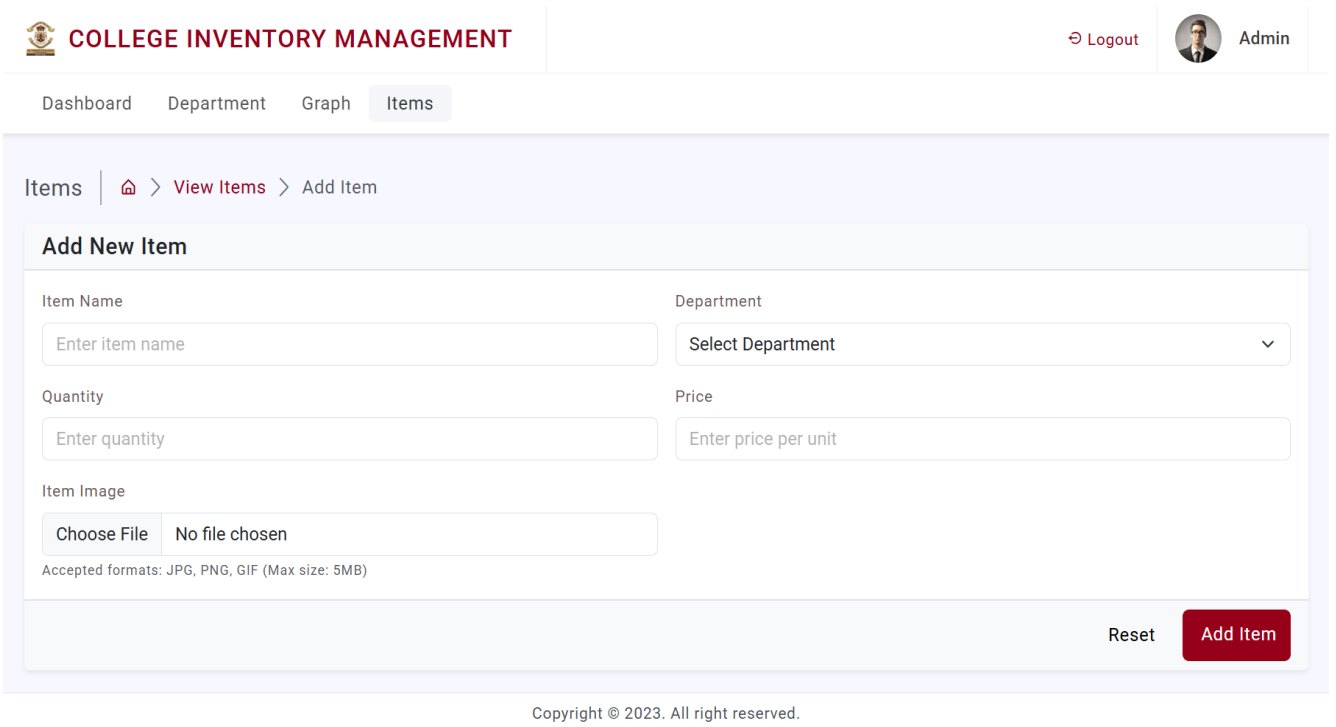
ADMIN DEPARTMENT PAGE:



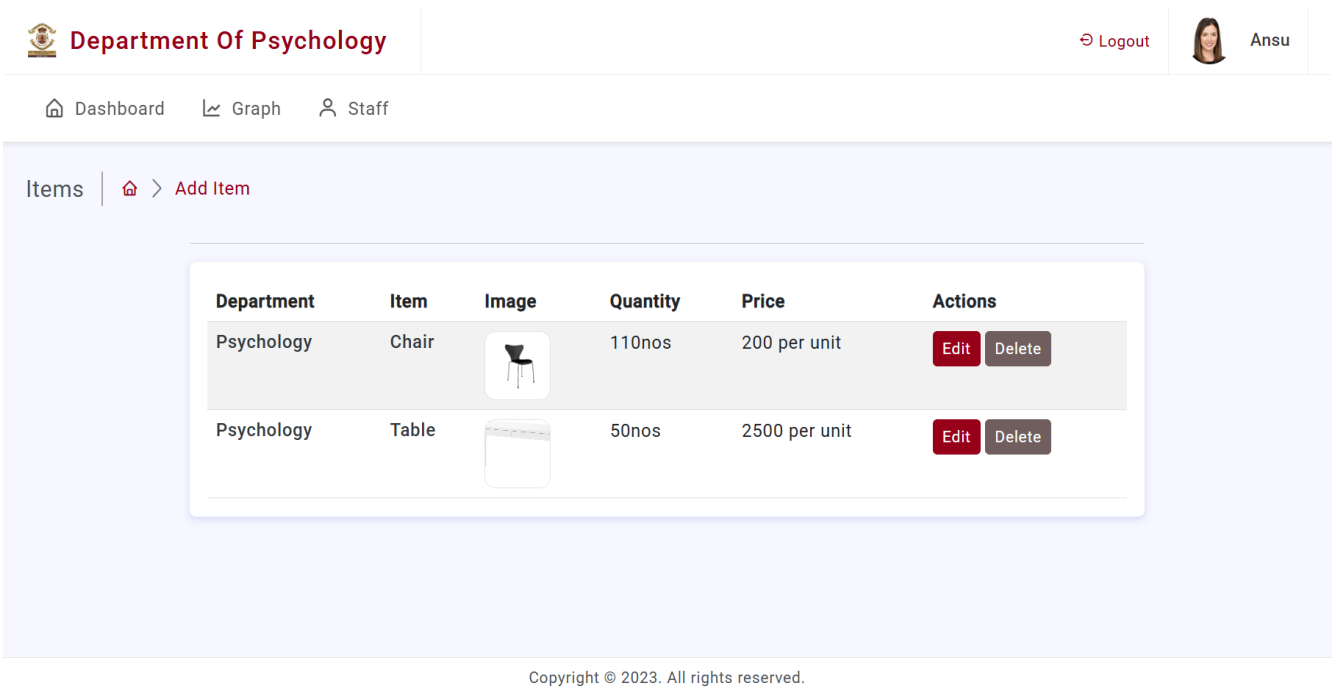
ADMIN GRAPH PAGE:



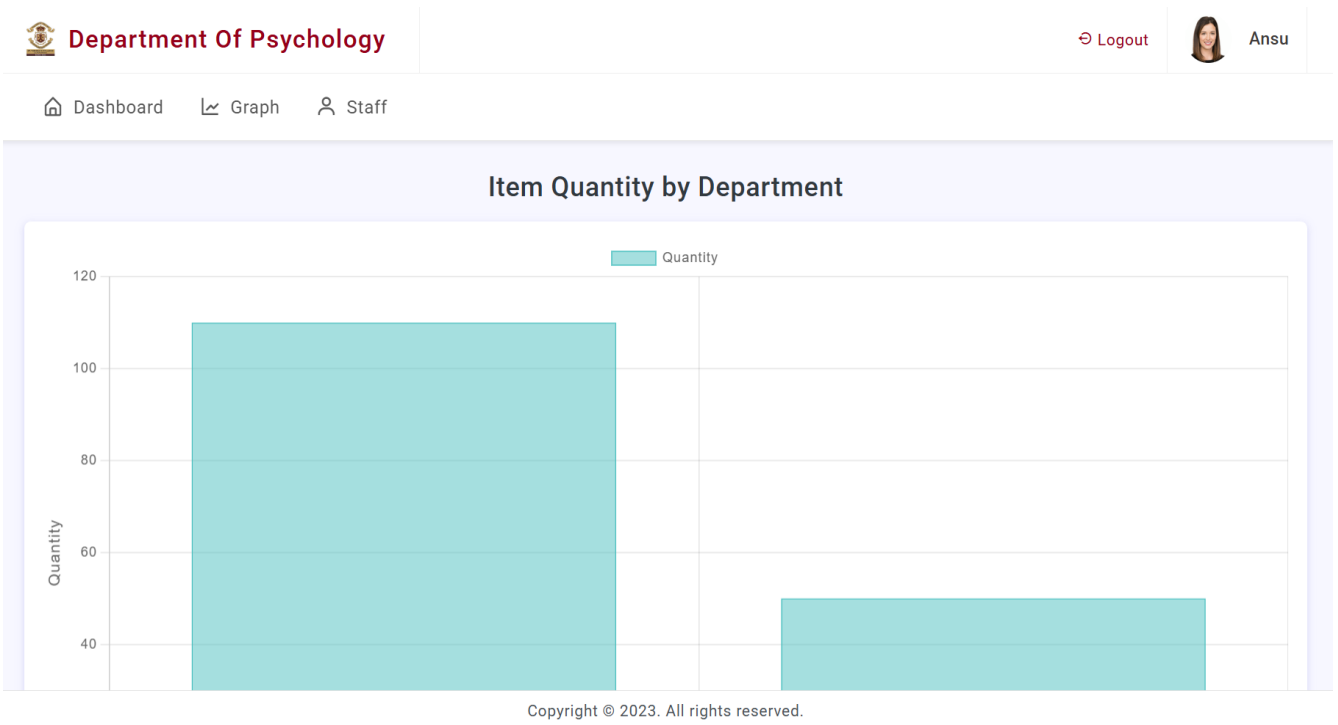
ADMIN ITEMS PAGE:



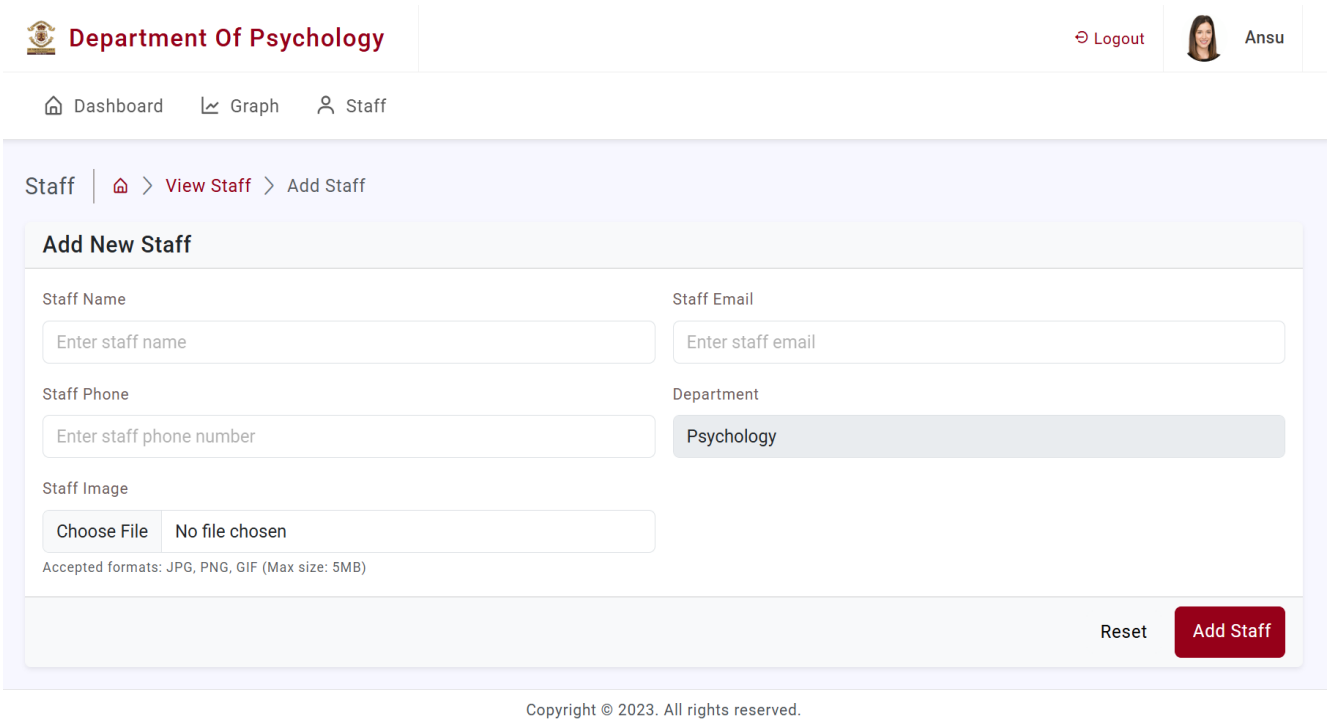
DEPARTMENT USER LANDING PAGE:



DEPARTMENT USER GRAPH PAGE:



DEPARTMENT USER STAFF PAGE:



SAMPLE CODE:

LOGIN:

```
{% for message in messages %}
  <script>
    alert('{{message}}')
  </script>
{% endfor %}
{% load static %}
<!doctype html>
<html lang="en">

<!-- Mirrored from codervent.com/rocker/demo/horizontal/auth-cover-signin.html by HTTrack Website Copier/3.x [XR&CO'2014], Mon, 06 Jan 2025 11:01:00 GMT -->
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--favicon-->
    <link rel="icon" href="{% static 'assets/images/favicon-32x32.png' %}" type="image/png" />
    <!--plugins-->
    <link href="{% static 'assets/plugins/simplebar/css/simplebar.css' %}" rel="stylesheet" />
    <link href="{% static 'assets/plugins/perfect-scrollbar/css/perfect-scrollbar.css' %}" rel="stylesheet" />
    <link href="{% static 'assets/plugins/metismenu/css/metisMenu.min.css' %}" rel="stylesheet" />
    <!-- loader-->
    <link href="{% static 'assets/css/pace.min.css' %}" rel="stylesheet" />
    <script src="{% static 'assets/js/pace.min.js' %}"></script>
    <!-- Bootstrap CSS -->
    <link href="{% static 'assets/css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static 'assets/css/bootstrap-extended.css' %}" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500&amp;display=swap" rel="stylesheet">
    <link href="{% static 'assets/css/app.css' %}" rel="stylesheet">
    <link href="{% static 'assets/css/icons.css' %}" rel="stylesheet">
    <title>STC Inventory</title>
</head>
```

```
<body class="">
   <!--wrapper-->
   <div class="wrapper">
      <div class="section-authentication-cover">
         <div class="">
            <div class="row g-0">

               <div class="col-12 col-xl-7 col-xxl-8 auth-cover-left align-items-center
justify-content-center d-none d-xl-flex">

                  <div class="card shadow-none bg-transparent shadow-none rounded-0
mb-0">
                     <div class="card-body">
                        <img src="{% static  'assets/images/login-images/login-cover.svg'
%}" class="img-fluid auth-img-cover-login" width="650" alt=""/>
                     </div>
                  </div>

               </div>

               <div class="col-12 col-xl-5 col-xxl-4 auth-cover-right align-items-center
justify-content-center">
                  <div class="card rounded-0 m-3 shadow-none bg-transparent mb-0">
                     <div class="card-body p-sm-5">
                        <div class="">
                           <div class="mb-3 text-center">
                              <img src="{% static 'assets/images/logo-icon.png' %}' %}"
width="60" alt="">
                           </div>
                           <div class="text-center mb-4">
                              <h5 class=""></h5>
                              <p class="mb-0">Please log in to your account</p>
                           </div>
                           <div class="form-body">
                              <form class="row g-3" method="post">
                                 {% csrf_token %}
                                 <div class="col-12">
                                    <label for="inputEmailAddress" class="form-la-
bel">Email</label>
                                    <input type="text" name="email" class="form-control"
id="inputEmailAddress" placeholder="Enter username or email">
                                 </div>
                                 <div class="col-12">
```

```
                                <label for="inputChoosePassword" class="form-la-
bel">Password</label>
                                <div class="input-group" id="show_hide_password">
                                    <input type="password" class="form-control border-
end-0" id="inputChoosePassword" name="password" placeholder="Enter Password">
<a href="javascript:;" class="input-group-text bg-transparent"><i class="bx bx-
hide"></i></a>
                                </div>
                            </div>

                            <div class="col-12">
                                <div class="d-grid">
                                    <button type="submit" class="btn btn-primary">Sign
in</button>
                                </div>
                            </div>
                            <div class="col-12">
                                <div class="text-center ">
                                    {% comment %} <p class="mb-0">Don't have an ac-
count yet? <a href="/register">Sign up here</a>
                                    </p> {% endcomment %}
                                </div>
                            </div>
                        </form>
                    </div>
                    {% comment %} <div class="login-separater text-center mb-
5"> <span>OR SIGN IN WITH</span>
                        <hr>
                    </div>
                    <div class="list-inline contacts-social text-center">
                        <a href="javascript:;" class="list-inline-item bg-facebook
text-white border-0 rounded-3"><i class="bx bxl-facebook"></i></a>
                        <a href="javascript:;" class="list-inline-item bg-twitter text-
white border-0 rounded-3"><i class="bx bxl-twitter"></i></a>
                        <a href="javascript:;" class="list-inline-item bg-google text-
white border-0 rounded-3"><i class="bx bxl-google"></i></a>
                        <a href="javascript:;" class="list-inline-item bg-linkedin text-
white border-0 rounded-3"><i class="bx bxl-linkedin"></i></a>
                    </div> {% endcomment %}

                </div>
            </div>
        </div>
```

```
                </div>

            </div>
            <!--end row-->
        </div>
    </div>
</div>
<!--end wrapper-->
<!-- Bootstrap JS -->
<script src="{% static 'assets/js/bootstrap.bundle.min.js' %}"></script>
<!--plugins-->
<script src="{% static 'assets/js/jquery.min.js' %}"></script>
<script src="{% static 'assets/plugins/simplebar/js/simplebar.min.js' %}"></script>
<script src="{% static 'assets/plugins/metismenu/js/metisMenu.min.js'
%}"></script>
<script src="{% static 'assets/plugins/perfect-scrollbar/js/perfect-scrollbar.js'
%}"></script>
<!--Password show & hide js -->
<script>
    $(document).ready(function () {
        $("#show_hide_password a").on('click', function (event) {
            event.preventDefault();
            if ($('#show_hide_password input').attr("type") == "text") {
                $('#show_hide_password input').attr('type', 'password');
                $('#show_hide_password i').addClass("bx-hide");
                $('#show_hide_password i').removeClass("bx-show");
            } else if ($('#show_hide_password input').attr("type") == "password") {
                $('#show_hide_password input').attr('type', 'text');
                $('#show_hide_password i').removeClass("bx-hide");
                $('#show_hide_password i').addClass("bx-show");
            }
        });
    });
</script>
<!--app JS-->
<script src="{% static 'assets/js/app.js' %}"></script>
</body>

<!-- Mirrored from codervent.com/rocker/demo/horizontal/auth-cover-signin.html by
HTTrack Website Copier/3.x [XR&CO'2014], Mon, 06 Jan 2025 11:01:00 GMT -->
</html>
```

PARTICIPANT REGISTRATION:

```
{% for message in messages %}
  <script>
    alert('{{message}}')
  </script>
{% endfor %}
{% load static %}
<!DOCTYPE html>
<html lang="en">
  <!-- Mirrored from codervent.com/rocker/demo/horizontal/auth-cover-signin.html by
HTTrack Website Copier/3.x [XR&CO'2014], Mon, 06 Jan 2025 11:01:00 GMT -->
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <!-- favicon -->
    <link rel="icon" href="{% static 'assets/images/favicon-32x32.png' %}" type="im-
age/png" />
    <!-- plugins -->
    <link href="{% static 'assets/plugins/simplebar/css/simplebar.css' %}"
rel="stylesheet" />
    <link href="{% static 'assets/plugins/perfect-scrollbar/css/perfect-scrollbar.css' %}"
rel="stylesheet" />
    <link href="{% static 'assets/plugins/metismenu/css/metisMenu.min.css' %}"
rel="stylesheet" />
    <!-- loader -->
    <link href="{% static 'assets/css/pace.min.css' %}" rel="stylesheet" />
    <script src="{% static 'assets/js/pace.min.js' %}"></script>
    <!-- Bootstrap CSS -->
    <link href="{% static 'assets/css/bootstrap.min.css' %}" rel="stylesheet" />
    <link href="{% static 'assets/css/bootstrap-extended.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/css2?family=Ro-
boto:wght@400;500&amp;display=swap" rel="stylesheet" />
    <link href="{% static 'assets/css/app.css' %}" rel="stylesheet" />
    <link href="{% static 'assets/css/icons.css' %}" rel="stylesheet" />
    <title>STC Inventory</title>
  </head>
  <body class="">
    <!-- wrapper -->
    <div class="wrapper">
      <div class="d-flex align-items-center justify-content-center my-5">
        <div class="container-fluid">
```

```
        <div class="row row-cols-1 row-cols-lg-2 row-cols-xl-3">
          <div class="col mx-auto">
            <div class="card mb-0">
              <div class="card-body">
                <div class="p-4">
                  <div class="mb-3 text-center">
                    <img src="{% static 'assets/images/logo-icon.png' %}" width="60"
alt="" />
                  </div>
                  <div class="text-center mb-4">
                    {% comment %} <h5 class="">Rocker Admin</h5> {% endcomment
%}
                    <p class="mb-0">Please fill the below details to create your account</p>
                  </div>
                  <div class="form-body">
          <form class="row g-3" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            <div class="col-12">
              <label for="inputName" class="form-label">Name</label>
              <input type="text" class="form-control" id="inputName" name="name"
placeholder="Enter your name" required />
            </div>
            <div class="col-12">
              <label for="inputEmailAddress" class="form-label">Email Address</label>
              <input type="email" class="form-control" id="inputEmailAddress"
name="email" placeholder="Enter your email" required />
            </div>
            <div class="col-12">
              <label for="inputPhone" class="form-label">Phone Number</label>
              <input type="tel" class="form-control" id="inputPhone" name="phone"
placeholder="Enter your phone number" required />
            </div>
            <div class="col-12">
              <label for="inputImage" class="form-label">Upload Image</label>
              <input type="file" class="form-control" id="inputImage" name="image" ac-
cept="image/*" required />
            </div>
            <div class="col-12">
              <label for="inputPlace" class="form-label">Place</label>
              <input type="text" class="form-control" id="inputPlace" name="place"
placeholder="Enter your place" required />
            </div>
            <div class="col-12">
```

```
            <label for="inputPassword" class="form-label">Password</label>
            <input type="password" class="form-control" id="inputPassword"
name="password" placeholder="Enter your password" required />
          </div>
          <div class="col-12">
            <div class="d-grid">
              <button type="submit" class="btn btn-primary">Sign up</button>
            </div>
          </div>
          <div class="col-12">
            <div class="text-center">
              <p class="mb-0">
                Already have an account? <a href="/">Sign in here</a>
              </p>
            </div>
          </div>
        </form>

              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <!-- end row -->
  </div>
 </div>
</div>
<!-- end wrapper -->
<!-- Bootstrap JS -->
<script src="assets/js/bootstrap.bundle.min.js"></script>
<!-- plugins -->
<script src="assets/js/jquery.min.js"></script>
<script src="assets/plugins/simplebar/js/simplebar.min.js"></script>
<script src="assets/plugins/metismenu/js/metisMenu.min.js"></script>
<script src="assets/plugins/perfect-scrollbar/js/perfect-scrollbar.js"></script>
<!-- Password show & hide js -->
<script>
 $(document).ready(function () {
  $('#show_hide_password a').on('click', function (event) {
   event.preventDefault()
   if ($('#show_hide_password input').attr('type') == 'text') {
    $('#show_hide_password input').attr('type', 'password')
```

```
            $('#show_hide_password i').addClass('bx-hide')
            $('#show_hide_password i').removeClass('bx-show')
          } else if ($('#show_hide_password input').attr('type') == 'password') {
            $('#show_hide_password input').attr('type', 'text')
            $('#show_hide_password i').removeClass('bx-hide')
            $('#show_hide_password i').addClass('bx-show')
          }
        })
      })
    </script>
    <!-- app JS -->
    <script src="assets/js/app.js"></script>
  </body>

  <!-- Mirrored from codervent.com/rocker/demo/horizontal/auth-cover-signin.html by HTTrack Website Copier/3.x [XR&CO'2014], Mon, 06 Jan 2025 11:01:00 GMT -->
</html>
```

8.BIBILOGRAPHY

8.1 REFERENCES

- Mozilla Developer Network (MDN). **HTML Reference**. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML
- Mozilla Developer Network (MDN). **CSS Reference**. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS
- Mozilla Developer Network (MDN). **JavaScript Guide**. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript
- React Documentation. **Introduction to React**. Meta. Retrieved from https://react.dev
- Python Software Foundation. **Python 3 Documentation**. Retrieved from https://docs.python.org/3/
- Django Software Foundation. **Django Official Documentation**. Retrieved from https://docs.djangoproject.com/en/stable/
- W3C. **Web Standards & Best Practices**. World Wide Web Consortium. Retrieved from https://www.w3.org/
- DigitalOcean. **Django and Python Tutorials**. Retrieved from https://www.digitalocean.com/community/tutorials *(Helpful for Django deployment & backend concepts)*