

Nama : Inayah Zalza Dinda Rahayu

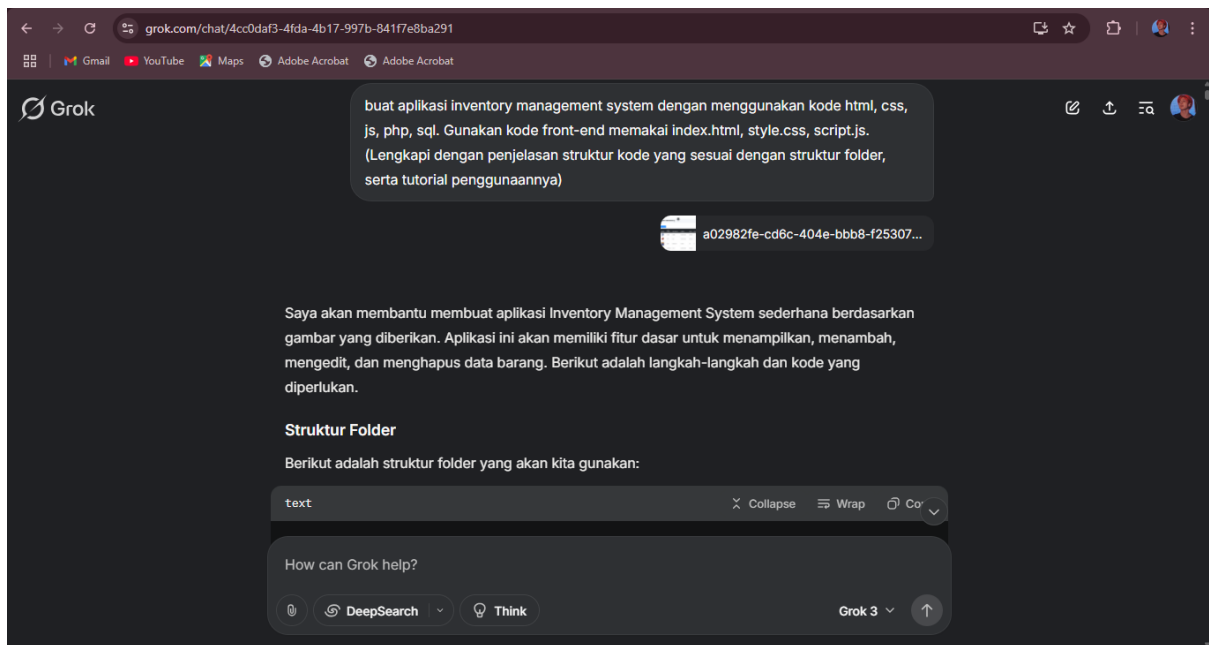
NIM : 12030123140270

Kelas : Pengkodean dan Pemrograman / D

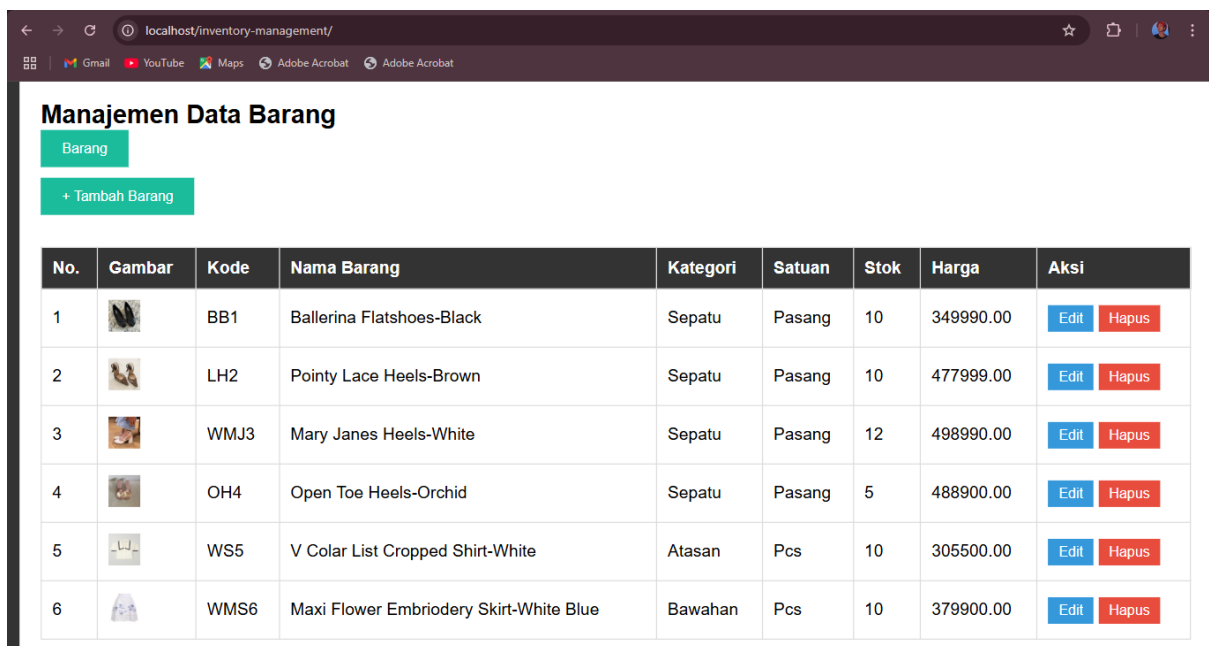
TUGAS 4

APLIKASI INVENTORY MANAGEMENT SYSTEM

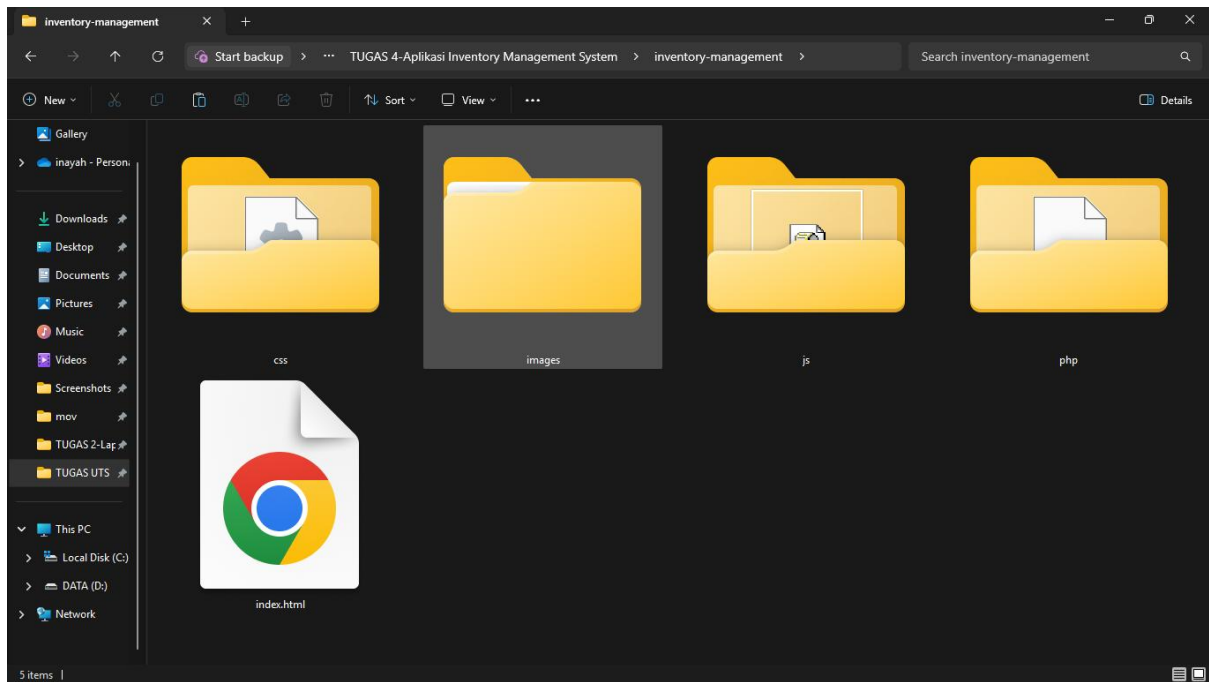
1. PROMPT GROK AI



2. TAMPILAN

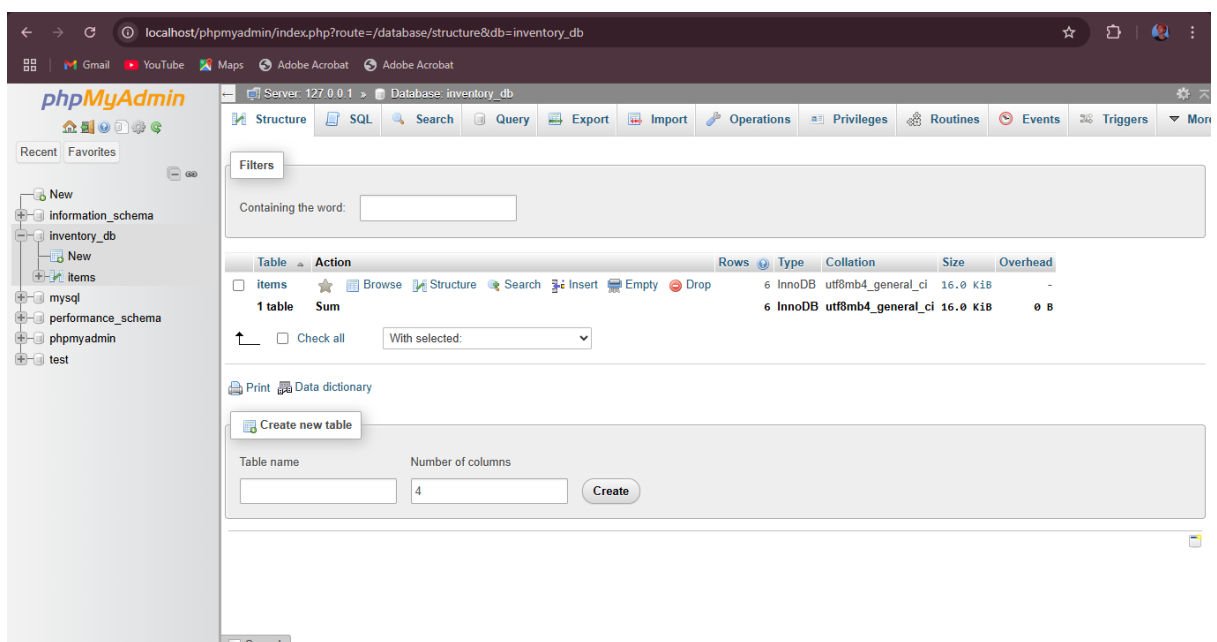


3. STRUKTUR FOLDER

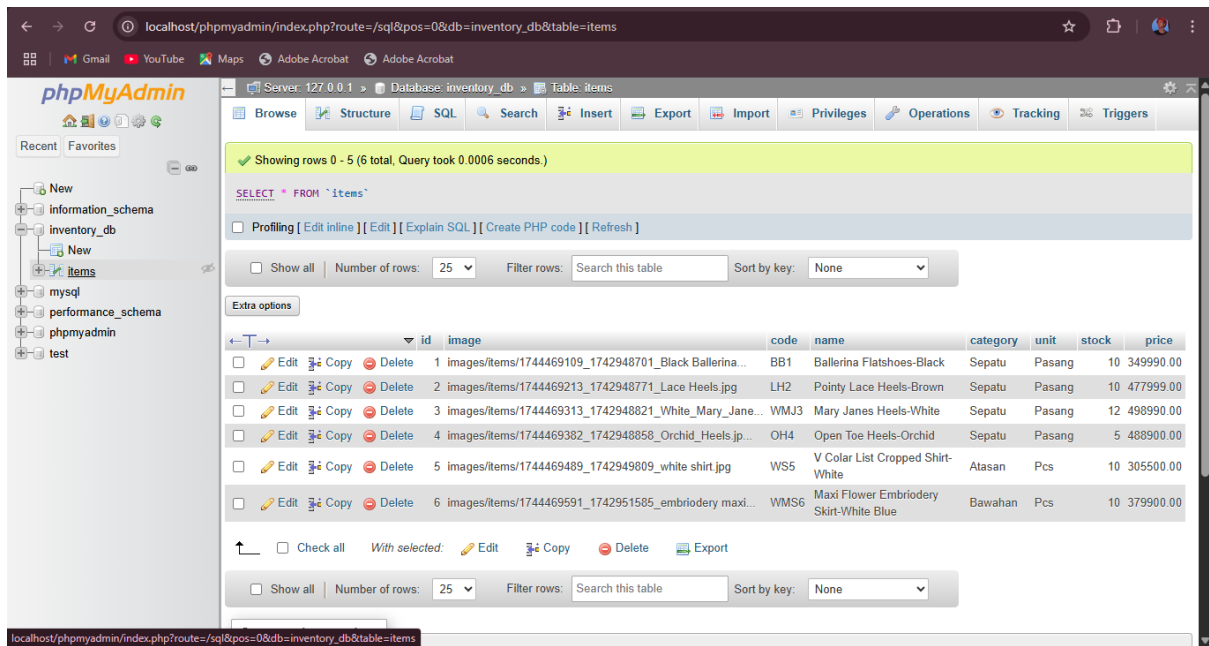


4. LOCALHOST PHPMYADMIN

a. DATABASE



b. TABLE ITEMS



5. KODE INDEX.HTML

```
<!DOCTYPE html>

<html lang="id">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Manajemen Data Barang</title>

  <link rel="stylesheet" href="css/style.css">

</head>

<body>

  <div class="sidebar">

    <h2>Pergudangan</h2>

    <div class="user">

      <img id="userPhoto" src="" alt="User">

      <p>Administrator</p>

    </div>

    <ul>

      <li>Master</li>

      <li>Kategori</li>

      <li>Satuan</li>
```

```
<li class="active">Barang</li>
</ul>
</div>
<div class="main-content">
  <h1>Manajemen Data Barang</h1>
  <div class="tab">
    <button class="tablinks active">Barang</button>
  </div>
  <button class="btn-add" onclick="openModal('add')">+ Tambah Barang</button>
  <div class="table-container"> <!-- Tambahkan div ini untuk membungkus tabel -->
    <table id="itemTable">
      <thead>
        <tr>
          <th>No.</th>
          <th>Gambar</th>
          <th>Kode</th>
          <th>Nama Barang</th>
          <th>Kategori</th>
          <th>Satuan</th>
          <th>Stok</th>
          <th>Harga</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody id="itemTableBody"></tbody>
    </table>
  </div>
</div>

<!-- Modal untuk Tambah/Edit Barang -->
<div id="itemModal" class="modal">
```

```

<div class="modal-content">
  <span class="close" onclick="closeModal()">×</span>
  <h2 id="modalTitle">Tambah Barang</h2>
  <form id="itemForm" enctype="multipart/form-data">
    <input type="hidden" id="itemId" name="id">
    <label for="image">Gambar:</label>
    <input type="file" id="image" name="image" accept="image/*"><br>
    <label for="code">Kode:</label>
    <input type="text" id="code" name="code" required><br>
    <label for="name">Nama Barang:</label>
    <input type="text" id="name" name="name" required><br>
    <label for="category">Kategori:</label>
    <input type="text" id="category" name="category" required><br>
    <label for="unit">Satuan:</label>
    <input type="text" id="unit" name="unit" required><br>
    <label for="stock">Stok:</label>
    <input type="number" id="stock" name="stock" required><br>
    <label for="price">Harga:</label>
    <input type="number" id="price" name="price" step="0.01" required><br>
    <button type="submit">Simpan</button>
  </form>
</div>
</div>

<script src="js/script.js"></script>
</body>
</html>

```

6. KODE CSS/STYLE.CSS

```

body {
  margin: 0;

```

```
font-family: Arial, sans-serif;

display: flex;

height: 100vh; /* Pastikan body mengambil tinggi penuh viewport */

overflow: hidden; /* Cegah scroll pada body */
}

.sidebar {

width: 200px;

background-color: #333;

color: white;

padding: 20px;

height: 100vh;
}

.sidebar h2 {

margin: 0;

font-size: 20px;
}

.sidebar .user {

text-align: center;

margin: 20px 0;
}

.sidebar .user img {

border-radius: 50%;

width: 50px;

height: 50px;
}

.sidebar ul {
```

```
list-style: none;
padding: 0;
}

.sidebar ul li {
padding: 10px;
cursor: pointer;
}

.sidebar ul li.active {
background-color: #1abc9c;
}

.main-content {
flex: 1;
padding: 20px;
}

h1 {
margin: 0;
font-size: 24px;
}

.tab .tablinks {
padding: 10px 20px;
background-color: #f1f1f1;
border: none;
cursor: pointer;
}

.tab .tablinks.active {
```

```
background-color: #1abc9c;
color: white;
}

.btn-add {
background-color: #1abc9c;
color: white;
padding: 10px 20px;
border: none;
cursor: pointer;
margin: 10px 0;
}

.table-container {
max-height: 400px; /* Batasi tinggi tabel, sesuaikan sesuai kebutuhan */
overflow-y: auto; /* Aktifkan scroll vertikal pada tabel */
}

table {
width: 100%;
border-collapse: collapse;
margin-top: 20px;
}

table, th, td {
border: 1px solid #ddd;
}

th, td {
padding: 10px;
text-align: left;
```



```
}

th {
  background-color: #333;
  color: white;
}

td img {
  width: 30px;
  height: 30px;
}

.btn-edit, .btn-delete {
  padding: 5px 10px;
  border: none;
  cursor: pointer;
}

.btn-edit {
  background-color: #3498db;
  color: white;
}

.btn-delete {
  background-color: #e74c3c;
  color: white;
}

.modal {
  display: none;
  position: fixed;
```

```
top: 0;
left: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.5);
overflow: auto; /* Pastikan modal bisa di-scroll jika kontennya panjang */
}

.modal-content {
background-color: white;
margin: 15% auto;
padding: 20px;
width: 50%;
border-radius: 5px;
}

.close {
float: right;
font-size: 24px;
cursor: pointer;
}

form label {
display: block;
margin: 10px 0 5px;
}

form input {
width: 100%;
padding: 8px;
margin-bottom: 10px;
```

```
}

form button {
  background-color: #1abc9c;
  color: white;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

7. KODE JS/SCRIPT.JS

```
document.addEventListener('DOMContentLoaded', () => {
  fetchItems();

  document.getElementById('itemForm').addEventListener('submit', (e) => {
    e.preventDefault();
    const formData = new FormData(e.target);
    const id = document.getElementById('itemId').value;

    if (id) {
      // Edit item
      fetch('php/edit_item.php', {
        method: 'POST',
        body: formData
      })
        .then(response => response.json())
        .then(data => {
          if (data.success) {
            fetchItems();
            closeModal();
          } else {
```

```

        alert('Gagal mengedit barang');
    }
});
} else {
    // Add item
    fetch('php/add_item.php', {
        method: 'POST',
        body: formData
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            fetchItems();
            closeModal();
        } else {
            alert('Gagal menambah barang');
        }
    });
}
});

function fetchItems() {
    fetch('php/config.php')
    .then(response => response.json())
    .then(data => {
        const tbody = document.getElementById('itemTableBody');
        tbody.innerHTML = "";
        data.forEach((item, index) => {
            const row = `
                <tr>

```

```

        <td>${ index + 1 }</td>
        <td></td>
        <td>${ item.code}</td>
        <td>${ item.name}</td>
        <td>${ item.category}</td>
        <td>${ item.unit}</td>
        <td>${ item.stock}</td>
        <td>${ item.price}</td>
        <td>
            <button class="btn-edit" onclick="openModal('edit',
${ item.id})">Edit</button>
            <button class="btn-delete"
onclick="deleteItem(${ item.id})">Hapus</button>
        </td>
    </tr>
    `;
    tbody.innerHTML += row;
    });
    });
}

```

```

function openModal(mode, id = null) {
    const modal = document.getElementById('itemModal');
    const form = document.getElementById('itemForm');
    const modalTitle = document.getElementById('modalTitle');

    form.reset();
    document.getElementById('itemId').value = "";

    if (mode === 'edit' && id) {
        modalTitle.textContent = 'Edit Barang';
    }
}

```

```
fetch(`php/config.php?id=${id}`)
    .then(response => response.json())
    .then(data => {
        document.getElementById('itemId').value = data.id;
        document.getElementById('code').value = data.code;
        document.getElementById('name').value = data.name;
        document.getElementById('category').value = data.category;
        document.getElementById('unit').value = data.unit;
        document.getElementById('stock').value = data.stock;
        document.getElementById('price').value = data.price;
    });
} else {
    modalTitle.textContent = 'Tambah Barang';
}

modal.style.display = 'block';
}

function closeModal() {
    document.getElementById('itemModal').style.display = 'none';
}

function deleteItem(id) {
    if (confirm('Apakah Anda yakin ingin menghapus barang ini?')) {
        fetch(`php/delete_item.php?id=${id}`, {
            method: 'DELETE'
        })
        .then(response => response.json())
        .then(data => {
            if (data.success) {
                fetchItems();
            }
        })
    }
}
```

```

        } else {
            alert('Gagal menghapus barang');
        }
    });
}
}

document.addEventListener('DOMContentLoaded', () => {
    fetchItems();
    fetchUserPhoto();
});

function fetchUserPhoto() {
    fetch('php/get_user.php') // File PHP baru untuk mengambil data user
        .then(response => response.json())
        .then(data => {
            document.getElementById('userPhoto').src = data.photo ||
            'https://via.placeholder.com/50'; // Fallback jika foto tidak ada
        })
        .catch(error => {
            console.error('Error fetching user photo:', error);
            document.getElementById('userPhoto').src = 'https://via.placeholder.com/50';
        });
}
}

```

8. KODE PHP

a. ADD_ITEM.PHP

```

<?php
header('Content-Type: application/json');

$servername = "localhost";
$username = "root";
$password = "";

```

```
$dbname = "inventory_db";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die(json_encode(['success' => false, 'message' => 'Connection failed: ' . $conn->connect_error]));
}

$code = $_POST['code'];
$name = $_POST['name'];
$category = $_POST['category'];
$unit = $_POST['unit'];
$stock = $_POST['stock'];
$price = $_POST['price'];

// Handle image upload
$image = "";
if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
    $target_dir = "../images/items/";
    // Pastikan folder ada
    if (!file_exists($target_dir)) {
        mkdir($target_dir, 0777, true);
    }
    // Gunakan nama file yang unik untuk menghindari overwrite
    $image_name = time() . '_' . basename($_FILES['image']['name']);
    $image = $target_dir . $image_name;
    if (move_uploaded_file($_FILES['image']['tmp_name'], $image)) {
        // Path relatif untuk disimpan di database
        $image = "images/items/" . $image_name;
    } else {
        echo json_encode(['success' => false, 'message' => 'Failed to upload image']);
    }
}
```



```

        exit;
    }
}

$sql = "INSERT INTO items (image, code, name, category, unit, stock, price) VALUES
('$image', '$code', '$name', '$category', '$unit', $stock, $price)";
if ($conn->query($sql) === TRUE) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'message' => 'Error: ' . $conn->error]);
}

$conn->close();
?>

```

b. ADD_USER.PHP

```

<?php
header('Content-Type: application/json');

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "inventory_db";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die(json_encode(['success' => false, 'message' => 'Connection failed: ' . $conn->connect_error]));
}

$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_DEFAULT);

```

```

// Handle image upload
$photo = "";
if (isset($_FILES['photo']) && $_FILES['photo']['error'] == 0) {
    $target_dir = "../images/users/";
    if (!file_exists($target_dir)) {
        mkdir($target_dir, 0777, true);
    }
    $photo = $target_dir . basename($_FILES['photo']['name']);
    move_uploaded_file($_FILES['photo']['tmp_name'], $photo);
}

$sql = "INSERT INTO users (username, password, photo) VALUES ('$username',
'$password', '$photo')";
if ($conn->query($sql) === TRUE) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'message' => 'Error: ' . $conn->error]);
}

$conn->close();
?>

```

c. CONFIG.PHP

```

<?php
header('Content-Type: application/json');

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "inventory_db";

```

```

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die(json_encode(['success' => false, 'message' => 'Connection failed: ' . $conn->connect_error]));
}

// Fetch items
if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "SELECT * FROM items WHERE id = $id";
    $result = $conn->query($sql);
    $item = $result->fetch_assoc();
    echo json_encode($item);
} else {
    $sql = "SELECT * FROM items";
    $result = $conn->query($sql);
    $items = [];
    while ($row = $result->fetch_assoc()) {
        $items[] = $row;
    }
    echo json_encode($items);
}

$conn->close();
?>

```

d. DELETE_ITEM.PHP

```

<?php
header('Content-Type: application/json');

```

```

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "inventory_db";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die(json_encode(['success' => false, 'message' => 'Connection failed: ' . $conn->connect_error]));
}

$id = $_GET['id'];
$sql = "DELETE FROM items WHERE id=$id";

if ($conn->query($sql) === TRUE) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'message' => 'Error: ' . $conn->error]);
}

$conn->close();
?>

```

e. EDIT_ITEM.PHP

```

<?php
header('Content-Type: application/json');

$servername = "localhost";
$username = "root";
$password = "";

```

```
$dbname = "inventory_db";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die(json_encode(['success' => false, 'message' => 'Connection failed: ' . $conn->connect_error]));
}

$id = $_POST['id'];
$code = $_POST['code'];
$name = $_POST['name'];
$category = $_POST['category'];
$unit = $_POST['unit'];
$stock = $_POST['stock'];
$price = $_POST['price'];

// Handle image upload
$image = "";
if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
    $target_dir = "../images/items/";
    if (!file_exists($target_dir)) {
        mkdir($target_dir, 0777, true);
    }
    $image_name = time() . '_' . basename($_FILES['image']['name']);
    $image = $target_dir . $image_name;
    if (move_uploaded_file($_FILES['image']['tmp_name'], $image)) {
        $image = "images/items/" . $image_name;
    } else {
        echo json_encode(['success' => false, 'message' => 'Failed to upload image']);
        exit;
    }
}
```

```

}

if ($image) {
    $sql = "UPDATE items SET image='$image', code='$code', name='$name',
category='$category', unit='$unit', stock=$stock, price=$price WHERE id=$id";
} else {
    $sql = "UPDATE items SET code='$code', name='$name', category='$category',
unit='$unit', stock=$stock, price=$price WHERE id=$id";
}

if ($conn->query($sql) === TRUE) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'message' => 'Error: ' . $conn->error]);
}

$conn->close();
?>

```

f. GET_USER.PHP

```

<?php
header('Content-Type: application/json');

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "inventory_db";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die(json_encode(['success' => false, 'message' => 'Connection failed: ' . $conn->connect_error]));
}

```

```
// Ambil data user (misalnya user dengan ID 1)
$sql = "SELECT photo FROM users WHERE id = 1"; // Ganti ID sesuai user yang login
$result = $conn->query($sql);
$user = $result->fetch_assoc();

echo json_encode($user);

$conn->close();

?>
```