DTS AngularJS Capstone Assignment

Name: Atharva Athanikar 72287317E

Batch: DTS Batch-3

Task 1: Use Components for showing profile photo as shown in question paper

user-profile.component.ts:

import { Component } from '@angular/core';

@Component({

  selector: 'app-user-profile',

  templateUrl: './user-profile.component.html',

  styleUrls: ['./user-profile.component.css']

})

export class UserProfileComponent {

  user = {

    name: 'John Doe',

    email: 'john.doe@gmail.com',

    contact: '+1-234-567-8901',

    age: 35,

    gender: 'Male',

    address: '5th Avenue, New York, United States'

  };

}

user-profile.component.html:

<div class="profile-container">

    <h2>User Profile</h2>

    <img src="assets/profile-picture.webp" alt="Profile Picture" class="profile-picture" />

    <p><strong>Name:</strong> {{ user.name }}</p>

    <p><strong>Email:</strong> {{ user.email }}</p>

    <p><strong>Contact:</strong> {{ user.contact }}</p>

    <p><strong>Age:</strong> {{ user.age }}</p>

    <p><strong>Gender:</strong> {{ user.gender }}</p>

    <p><strong>Address:</strong> {{ user.address }}</p>

  </div>

user-profile.component.css:

.profile-container {

    border: 2px solid #ddd;

    padding: 20px;

    max-width: 400px;

    margin: auto;

    border-radius: 20px;

}

.profile-picture {

    width: 150px;

    height: 150px;

    border-radius: 50%;

    display: block;

    margin: 0 auto 15px;

}

h2 {

    text-align: center;

    text-decoration: underline;

}

app.component.ts:

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'User Profile Application';

}

app.component.html:

<div class="app-container">

  <h1>{{ title }}</h1>

  <br>

  <app-user-profile></app-user-profile>

</div>

app.component.css:

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f1f1f1;

  }

.header {

    background-color: #2c3e50;

    color: #fff;

    text-align: center;

    padding: 10px 0;

    font-size: 24px;

}

.profile-container {

    margin: 20px auto;

    max-width: 800px;

    padding: 15px;

    background-color: #fff;

    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);

    border-radius: 8px;

}

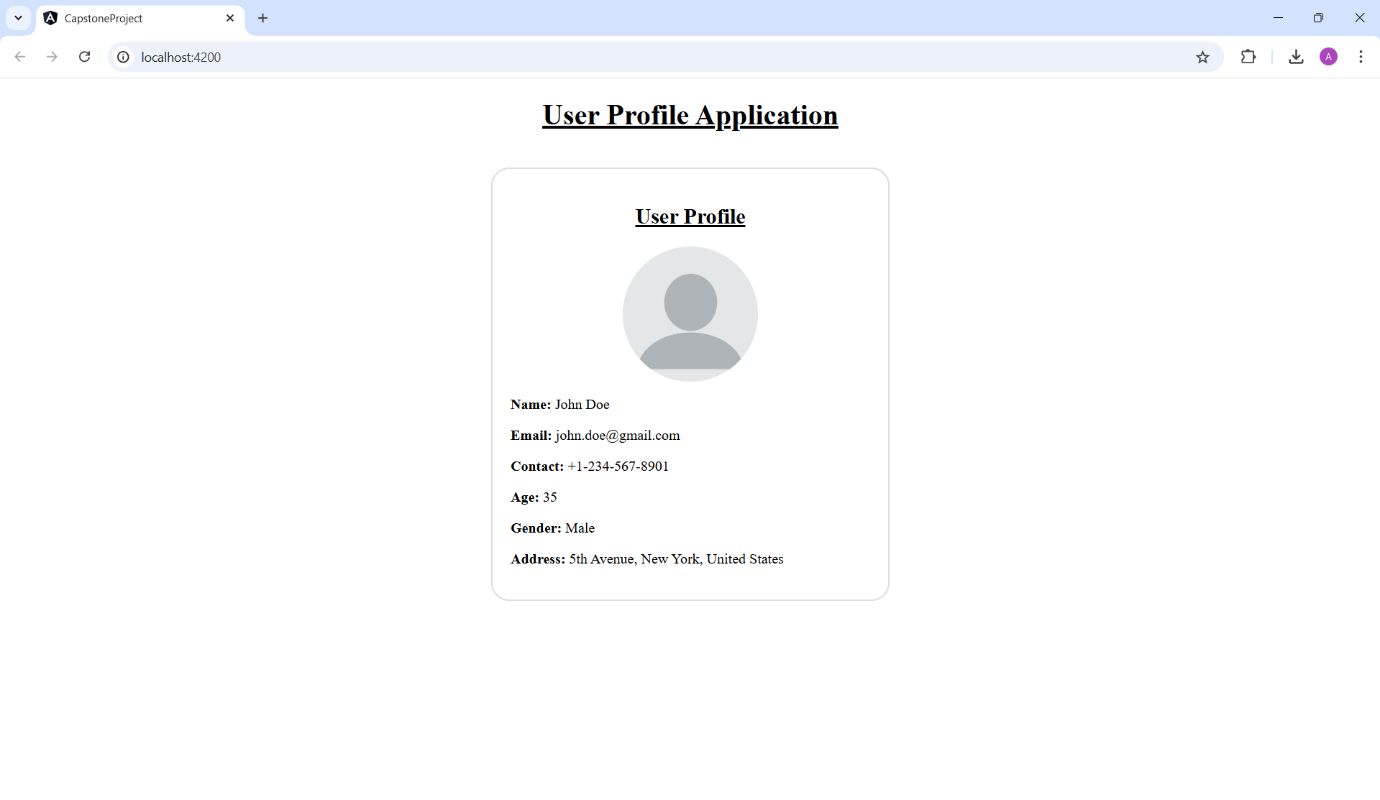
h1 {

    text-align: center;

    text-decoration: underline;

}

OUTPUT:



Task 2: Observe the output given below and implement the same functionality using Angular concepts

Note: Use templates and its elements to implement this

Note:

• If age is greater than 21, display "You are an Adult"

• If age between 13 and 21, display "You are a teen ager"

• In all other cases, display "You are a kid"

age-message.component.ts:

import { Component } from '@angular/core';

@Component({

  selector: 'app-age-message',

  templateUrl: './age-message.component.html',

  styleUrls: ['./age-message.component.css']

})

export class AgeMessageComponent {

  age: number = 0;

  message: string = '';

  showMessage() {

    if (this.age > 21) {

      this.message = 'You are an Adult';

    } else if (this.age >= 13 && this.age <= 21) {

      this.message = 'You are a Teenager';

    } else {

      this.message = 'You are a Kid';

    }

  }

}

age-message.component.html:

<div class="age-container">

    <h2>When was I born?</h2>

    <label for="age">Enter your Age:</label>

    <input type="number" id="age" [(ngModel)]="age" />

    <button (click)="showMessage()">Show Birth Year</button>

    <p><strong>{{ message }}</strong></p>

</div>

age-message.component.css:

.age-container {

    text-align: center;

    margin: 20px auto;

}

button {

    margin-top: 10px;

}

app.component.ts:

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'When was I born?';

  age: number = 0;

  message: string = '';

  showMessage() {

    if (this.age > 21) {

      this.message = 'You are an Adult';

    } else if (this.age >= 13 && this.age <= 21) {

      this.message = 'You are a Teenager';

    } else {

      this.message = 'You are a Kid';

    }

  }

}

app.component.html:

<div class="header">

  <h1>{{ title }}</h1>

</div>

<div class="age-container">

  <label for="age">Enter your Age:</label>

  <input type="number" id="age" [(ngModel)]="age" />

  <button (click)="showMessage()">Show Birth Year</button>

  <p \*ngIf="message"><strong>{{ message }}</strong></p>

</div>

app.component.css:

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f4f4f4;

}

.header {

    background-color: #2c3e50;

    color: #fff;

    text-align: center;

    padding: 10px 0;

    font-size: 24px;

}

.age-container {

    background-color: #fff;

    max-width: 600px;

    margin: 20px auto;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);

    text-align: center;

}

label {

    display: block;

    font-size: 18px;

    margin-bottom: 10px;

}

input[type="number"] {

    padding: 8px;

    width: 80%;

    max-width: 300px;

    margin-bottom: 15px;

    border: 1px solid #ccc;

    border-radius: 4px;

}

button {

    background-color: #3498db;

    color: #fff;

    border: none;

    padding: 10px 20px;

    font-size: 16px;

    border-radius: 4px;

    cursor: pointer;

    transition: background-color 0.3s;

}

button:hover {

    background-color: #2980b9;

}

p {

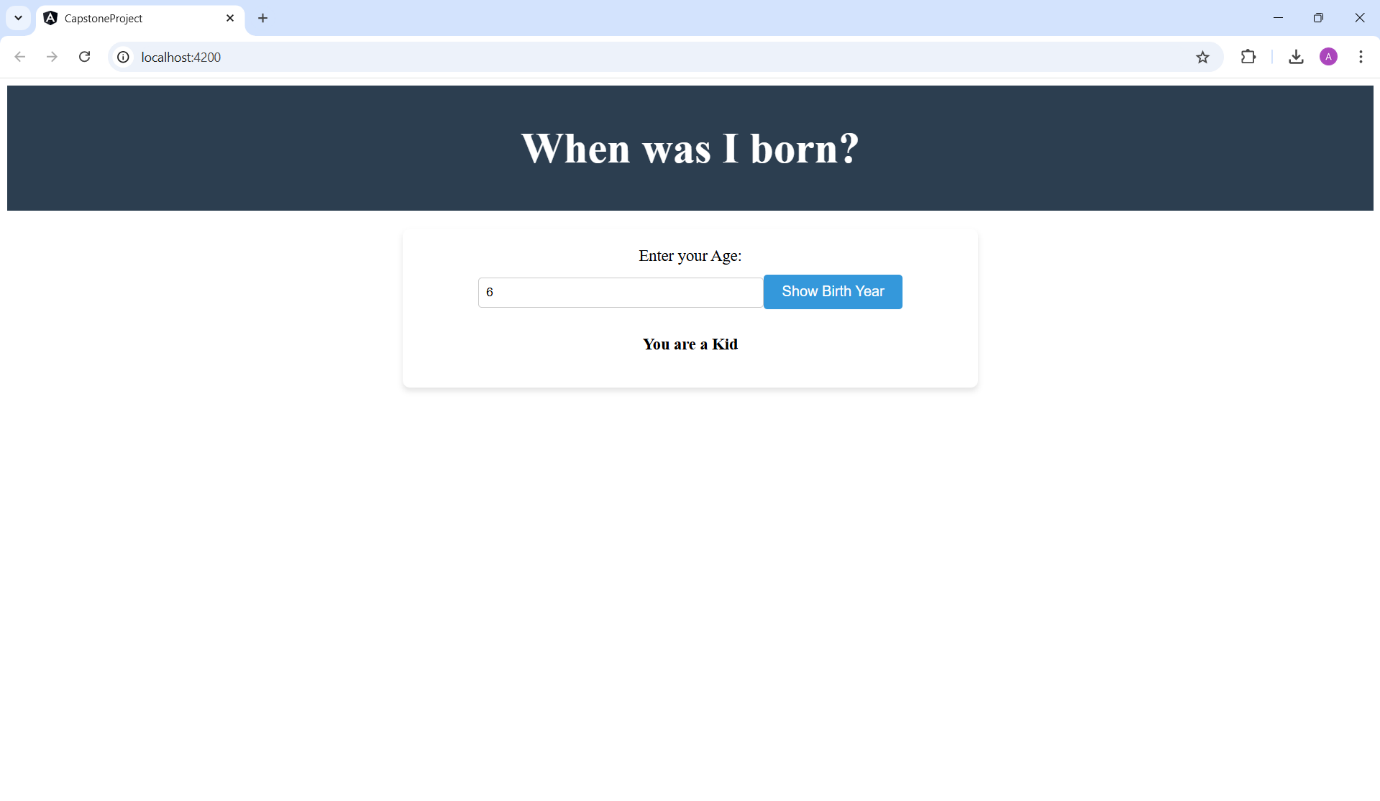
    font-size: 18px;

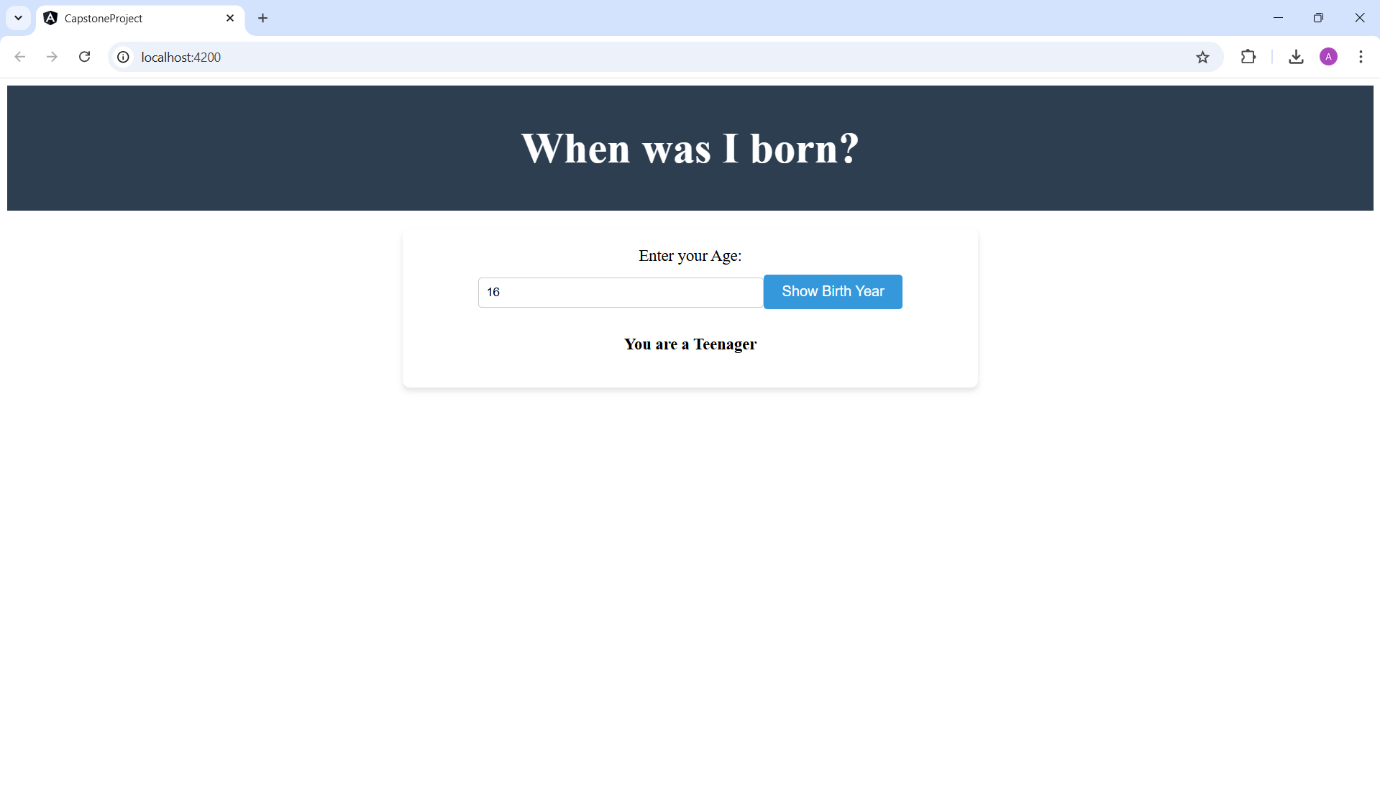
    margin-top: 15px;

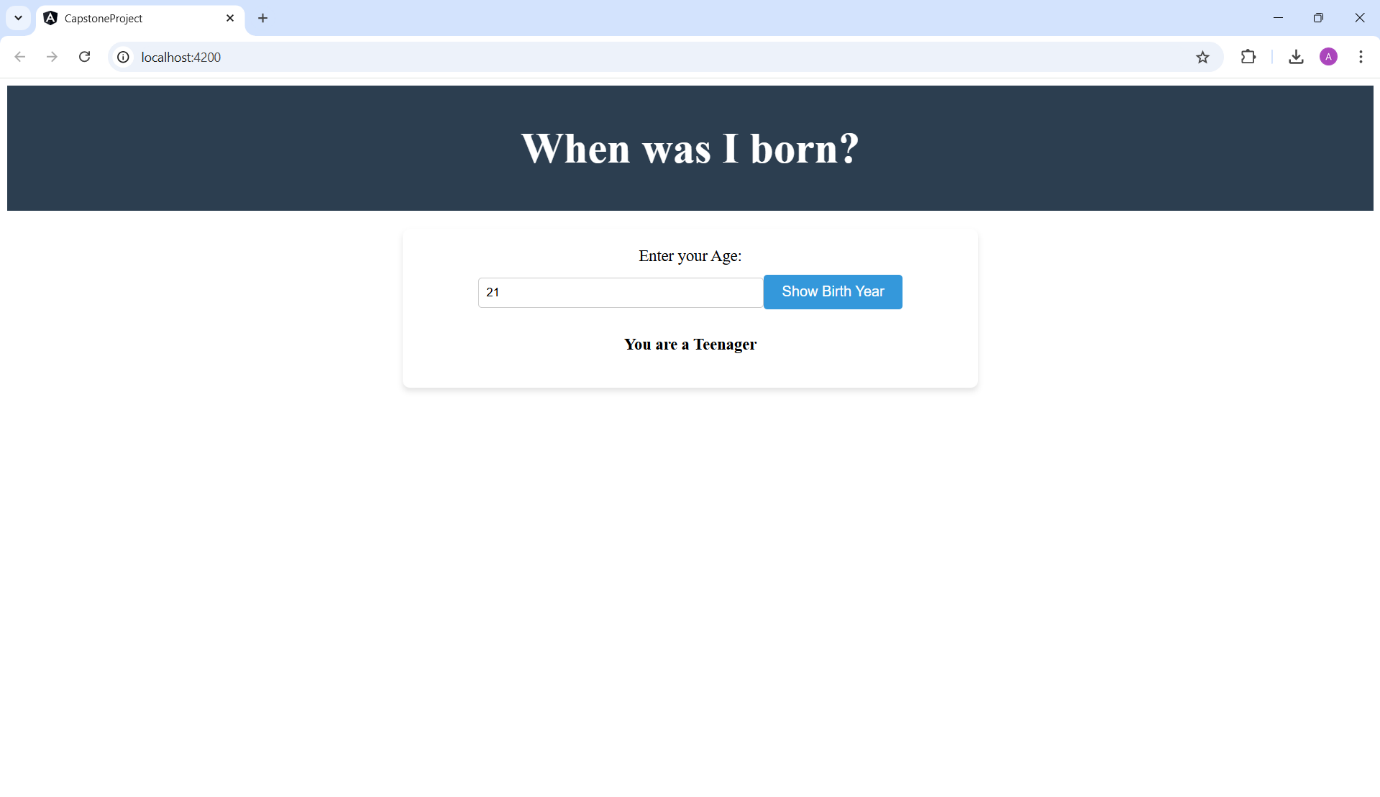
    font-weight: bold;

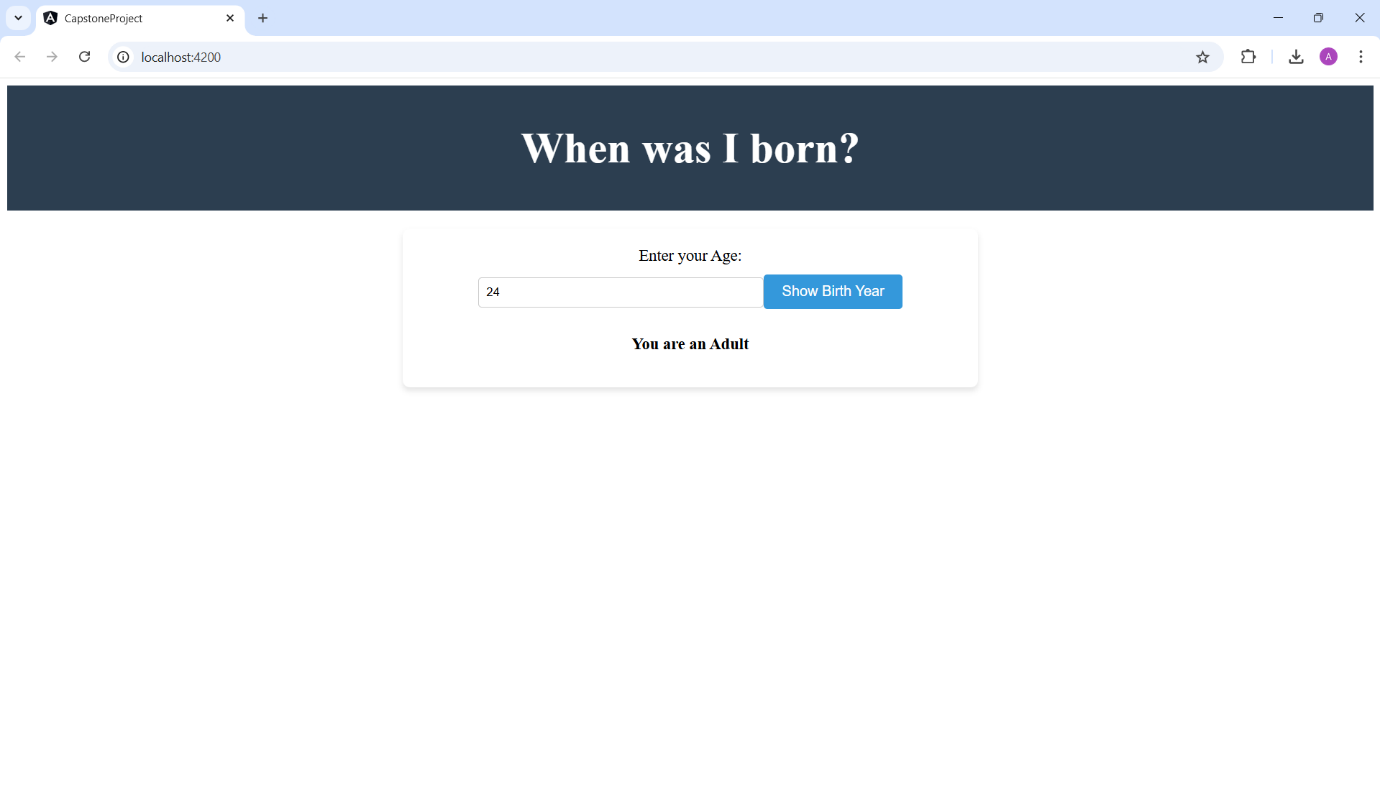
}

OUTPUT:









Task 3: **Directives – 3 (5 Marks)**  
**Problem Statement:**  
Observe the output given below and implement the same functionality using Angular.  
Use the below array data to achieve the desired output:

[

{ orderId: 2001, productName: "iphone 11 pro", quantity: 1, billAmount: 1850, transaction: 'completed' },

{ orderId: 2002, productName: "iphone 10", quantity: 1, billAmount: 990, transaction: 'pending' },

{ orderId: 2003, productName: "Samsung Galaxy S10", quantity: 2, billAmount: 1980, transaction: 'completed' },

{ orderId: 2004, productName: "Motorolla Edge+", quantity: 1, billAmount: 980, transaction: 'pending' }

]

past-orders.component.ts:

import { Component } from '@angular/core';

@Component({

  selector: 'app-past-orders',

  templateUrl: './past-orders.component.html',

  styleUrls: ['./past-orders.component.css']

})

export class PastOrdersComponent {

  orders = [

    { orderId: 2001, productName: 'iPhone 11 Pro', quantity: 1, billAmount: 1850, transaction: 'Completed' },

    { orderId: 2002, productName: 'iPhone 10', quantity: 1, billAmount: 990, transaction: 'Pending' },

    { orderId: 2003, productName: 'Samsung Galaxy S10', quantity: 2, billAmount: 1980, transaction: 'Completed' },

    { orderId: 2004, productName: 'Motorola Edge+', quantity: 1, billAmount: 980, transaction: 'Pending' }

  ];

}

past-orders.component.html:

<div class="orders-container">

    <h2>Past Orders</h2>

    <table>

      <thead>

        <tr>

          <th>Order ID</th>

          <th>Product Name</th>

          <th>Quantity</th>

          <th>Bill Amount</th>

          <th>Transaction</th>

        </tr>

      </thead>

      <tbody>

        <tr \*ngFor="let order of orders">

          <td>{{ order.orderId }}</td>

          <td>{{ order.productName }}</td>

          <td>{{ order.quantity }}</td>

          <td>{{ order.billAmount }}</td>

          <td>{{ order.transaction }}</td>

        </tr>

      </tbody>

    </table>

</div>

past-orders.component.css:

.orders-container {

    margin: auto;

    max-width: 600px;

}

table {

    width: 100%;

    border-collapse: collapse;

}

th, td {

    border: 1px solid #ddd;

    padding: 8px;

}

th {

    background-color: #f4f4f4;

}

app.component.ts:

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  orders = [

    { orderId: 2001, productName: "iPhone 11 Pro", quantity: 1, billAmount: 1850, transaction: 'Completed' },

    { orderId: 2002, productName: "iPhone 10", quantity: 1, billAmount: 990, transaction: 'Pending' },

    { orderId: 2003, productName: "Samsung Galaxy S10", quantity: 2, billAmount: 1980, transaction: 'Completed' },

    { orderId: 2004, productName: "Motorolla Edge+", quantity: 1, billAmount: 980, transaction: 'Pending' }

  ];

}

app.component.html:

<div class="header">

  <h1>Past Orders</h1>

</div>

<div class="table-container">

  <table>

    <thead>

      <tr>

        <th>Order ID</th>

        <th>Product Name</th>

        <th>Quantity</th>

        <th>Bill Amount</th>

        <th>Transaction Status</th>

      </tr>

    </thead>

    <tbody>

      <tr \*ngFor="let order of orders">

        <td>{{ order.orderId }}</td>

        <td>{{ order.productName }}</td>

        <td>{{ order.quantity }}</td>

        <td>{{ order.billAmount }}</td>

        <td [class.Completed]="order.transaction === 'Completed'" [class.Pending]="order.transaction === 'Pending'">

          {{ order.transaction }}

        </td>

      </tr>

    </tbody>

  </table>

</div>

app.component.css:

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f4f4f4;

}

.header {

    background-color: #2c3e50;

    color: #fff;

    text-align: center;

    padding: 10px 0;

    font-size: 24px;

}

.table-container {

    max-width: 800px;

    margin: 20px auto;

    background-color: #fff;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);

}

table {

    width: 100%;

    border-collapse: collapse;

}

thead {

    background-color: #2c3e50;

    color: #fff;

}

th, td {

    text-align: left;

    padding: 10px;

    border: 1px solid #ddd;

}

tbody tr:nth-child(even) {

    background-color: #f9f9f9;

}

tbody tr:hover {

    background-color: #f1f1f1;

}

.Completed {

    color: green;

    font-weight: bold;

}

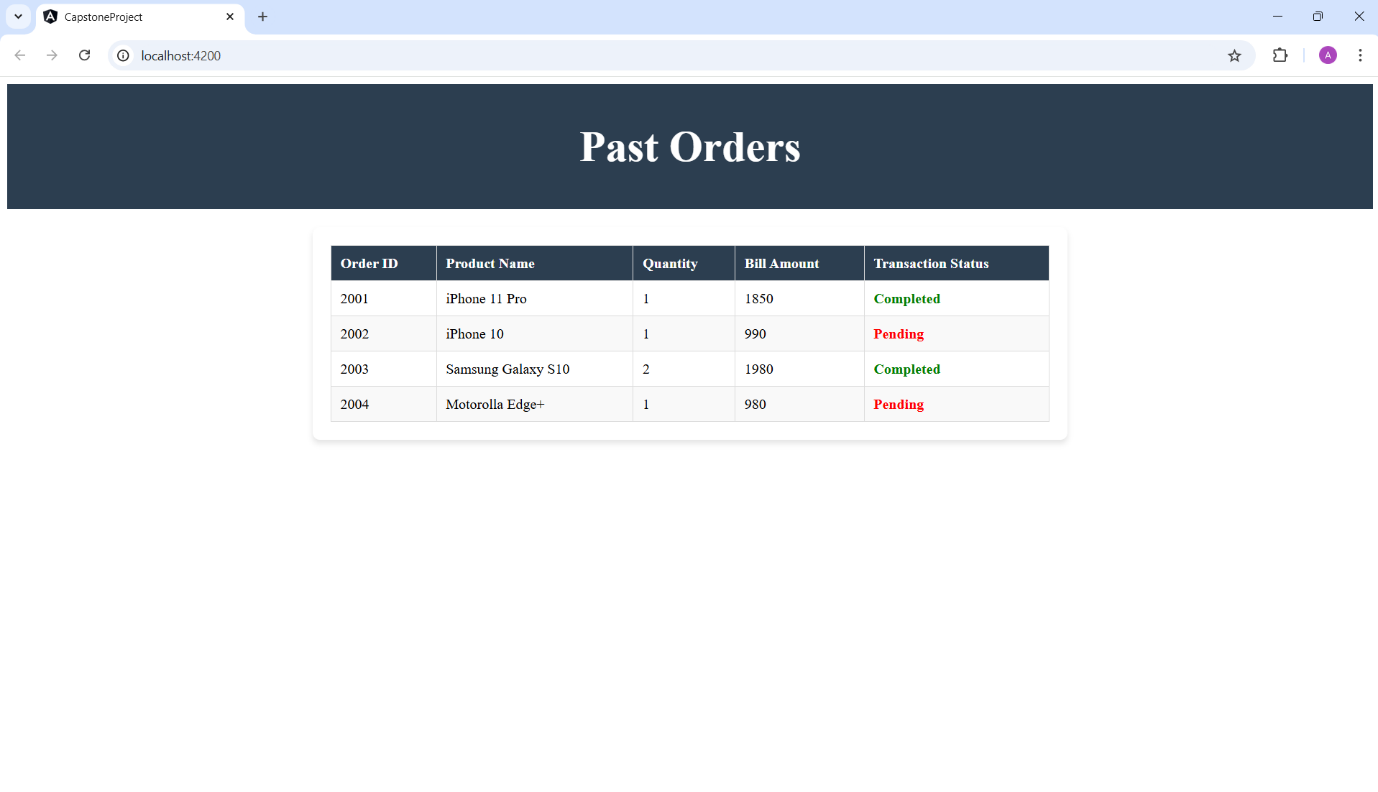
.Pending {

    color: red;

    font-weight: bold;

}

OUTPUT:



Task 4: **Create a login form with validations on username and password fields as shown below using template-driven form. (10 Marks)**

login-form.component.ts

import { Component } from '@angular/core';

@Component({

  selector: 'app-login-form',

  templateUrl: './login-form.component.html',

  styleUrls: ['./login-form.component.css']

})

export class LoginFormComponent {

  userName: string = '';

  password: string = '';

  loginSuccess: boolean = false;

  loginFailed: boolean = false;

  login(){

    if(this.userName == 'admin' && this.password == 'admin'){

      this.loginSuccess = true;

      this.loginFailed = false;

    } else {

      this.loginFailed = true;

      this.loginSuccess =false;

    }

  }

}

login-form.component.html

<div class="container">

    <form #loginForm="ngForm" (ngSubmit)="login()">

      <div>

        <label for="userName">UserName</label>

        <input type="text" name="userName" [(ngModel)]="userName" #userNameInput="ngModel" required>

        <div \*ngIf="userNameInput.invalid && (userNameInput.touched || !userNameInput.value)">

          <p style="color: red;">UserName is required.</p>

        </div>

      </div>

      <div>

        <label for="password">Password</label>

        <input type="password" name="password" [(ngModel)]="password" #passwordInput="ngModel" required>

        <div \*ngIf="passwordInput.invalid && (passwordInput.touched || !passwordInput.value)">

          <p style="color: red;">Password is required.</p>

        </div>

      </div>

      <button type="submit">Submit</button>

    </form>

    <p \*ngIf="loginSuccess">Login Success</p>

    <p \*ngIf="loginFailed">Login Failed</p>

</div>

login-form.component.css

.login-form-container {

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    background-color: #eaeaea;

  }

  .login-form {

    background-color: white;

    padding: 40px;

    border-radius: 12px;

    box-shadow: 0 6px 12px rgba(0, 0, 0, 0.1);

    max-width: 400px;

    width: 100%;

    box-sizing: border-box;

  }

  .login-form h2 {

    text-align: center;

    margin-bottom: 20px;

    font-size: 24px;

    color: #120e0e;

  }

  .input-group {

    margin: 15px 0;

    width: 100%;

  }

  .input-group label {

    font-size: 16px;

    font-weight: bold;

    color: #120e0e;

  }

  .input-group input {

    padding: 10px;

    width: 100%;

    border: 1px solid #ccc;

    border-radius: 5px;

    font-size: 16px;

  }

  .input-group input:focus {

    border-color: #120e0e;

    outline: none;

  }

  button {

    padding: 12px 20px;

    background-color: #120e0e;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    font-size: 16px;

    width: 100%;

    max-width: 300px;

    transition: background-color 0.3s ease-in-out;

  }

  button:hover {

    background-color: #8b0000;

  }

  button:active {

    background-color: #600000;

  }

  .error-message {

    color: red;

    font-size: 14px;

    font-weight: bold;

    text-align: center;

  }

app.component.ts

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  userName: string = '';

  password: string = '';

  loginSuccess: boolean = false;

  loginFailed: boolean = false;

  login() {

    if (this.userName === 'admin' && this.password === 'admin') {

      this.loginSuccess = true;

      this.loginFailed = false;

    } else {

      this.loginFailed = true;

      this.loginSuccess = false;

    }

  }

}

app.component.html

<div class="container">

  <form #loginForm="ngForm" (ngSubmit)="login()">

    <div>

      <label for="userName">Username</label>

      <input type="text" name="userName" [(ngModel)]="userName" #userNameInput="ngModel" required>

      <div \*ngIf="userNameInput.invalid && (userNameInput.touched || !userNameInput.value)">

        <p style="color: red;">Username is required.</p>

      </div>

    </div>

    <div>

      <label for="password">Password</label>

      <input type="password" name="password" [(ngModel)]="password" #passwordInput="ngModel" required>

      <div \*ngIf="passwordInput.invalid && (passwordInput.touched || !passwordInput.value)">

        <p style="color: red;">Password is required.</p>

      </div>

    </div>

    <div \*ngIf="loginSuccess" class="login-message success">Login Success</div>

    <div \*ngIf="loginFailed" class="login-message failed">Login Failed</div>

    <button type="submit">Submit</button>

  </form>

</div>

app.component.css

.container {

    height: 100vh;

    display: flex;

    justify-content: center;

    align-items: center;

    background-color: #f4f4f4;

}

  form {

    display: flex;

    flex-direction: column;

    align-items: center;

    background-color: #ffffff;

    padding: 30px;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    width: 100%;

    max-width: 350px;

    box-sizing: border-box;

}

  input {

    margin: 10px 0;

    padding: 12px;

    width: 100%;

    max-width: 260px;

    border: 1px solid #ccc;

    border-radius: 5px;

    font-size: 16px;

}

  input:focus {

    border-color: #120e0e;

    outline: none;

}

  p {

    text-align: center;

    font-size: 14px;

    color: #b22222;

    font-weight: bold;

}

  button {

    padding: 12px 20px;

    background-color: #120e0e;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    font-size: 16px;

    width: 100%;

    max-width: 300px;

    transition: background-color 0.3s ease-in-out;

}

  button:hover {

    background-color: #8b0000;

}

  button:active {

    background-color: #600000;

}

  .login-message {

    margin-bottom: 15px;

    font-size: 16px;

    text-align: center;

    font-weight: bold;

    padding: 10px;

    width: 100%;

    max-width: 300px;

}

  .success {

    color: green;

    background-color: #d4edda;

    border: 1px solid #c3e6cb;

}

  .failed {

    color: red;

    background-color: #f8d7da;

    border: 1px solid #f5c6cb;

}

  @media (max-width: 480px) {

    form {

      padding: 20px;

}

    input,

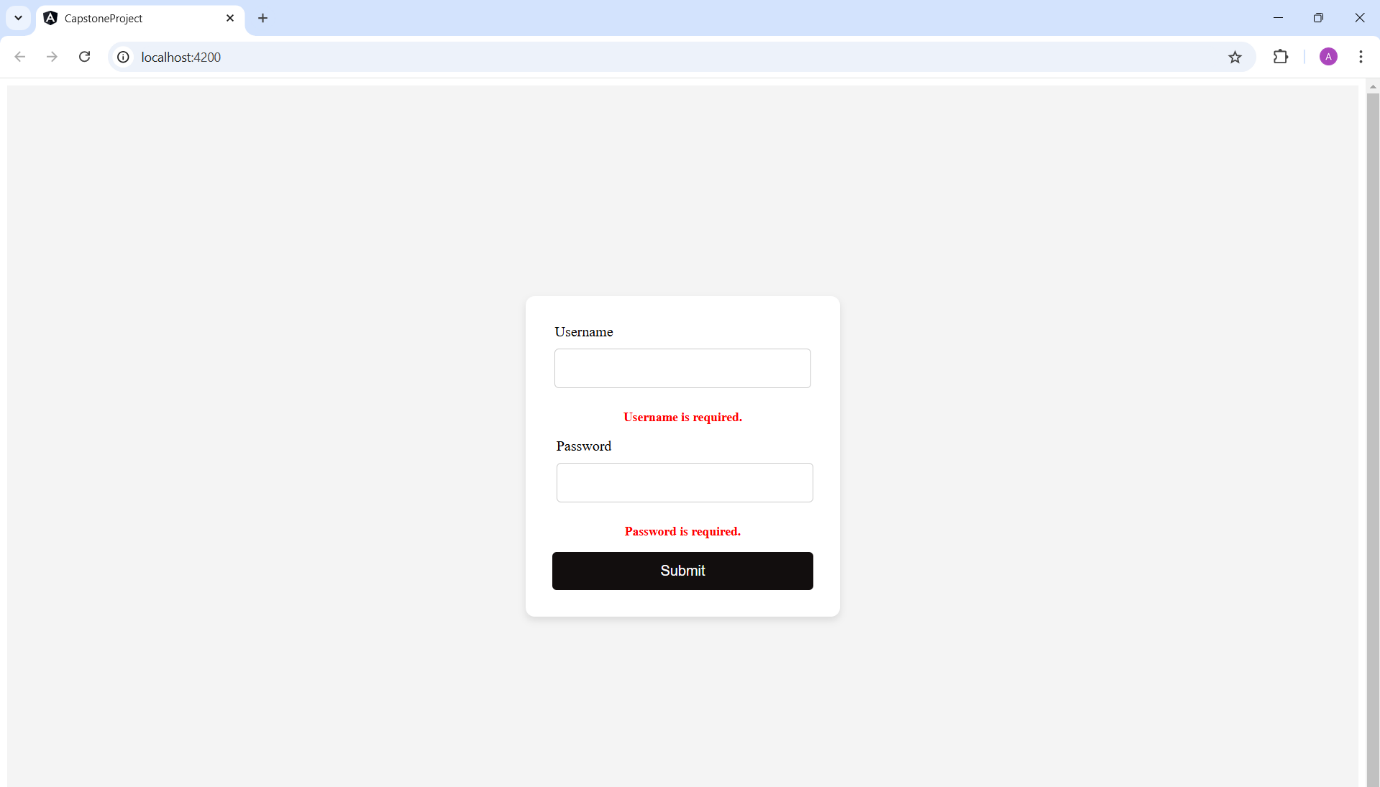
    button {

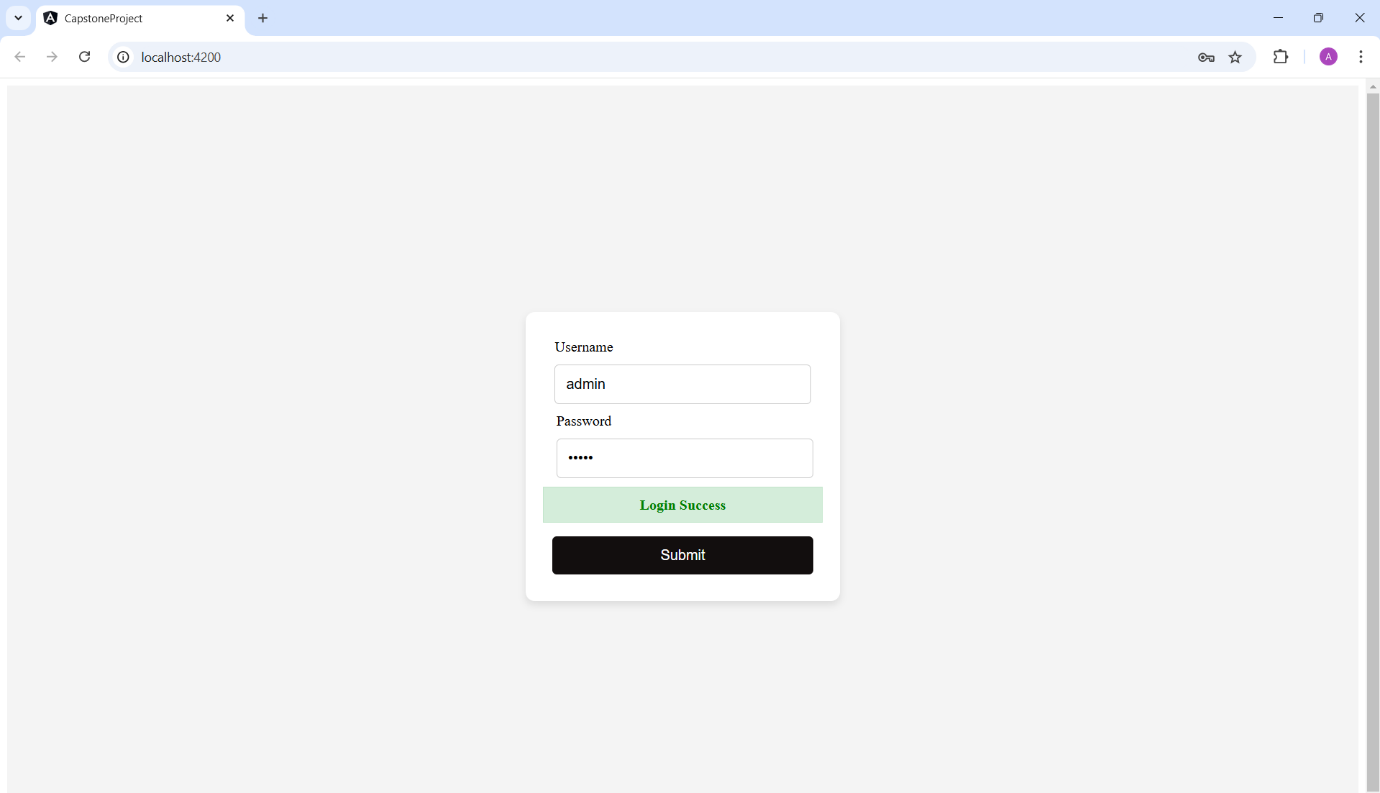
      max-width: 90%;

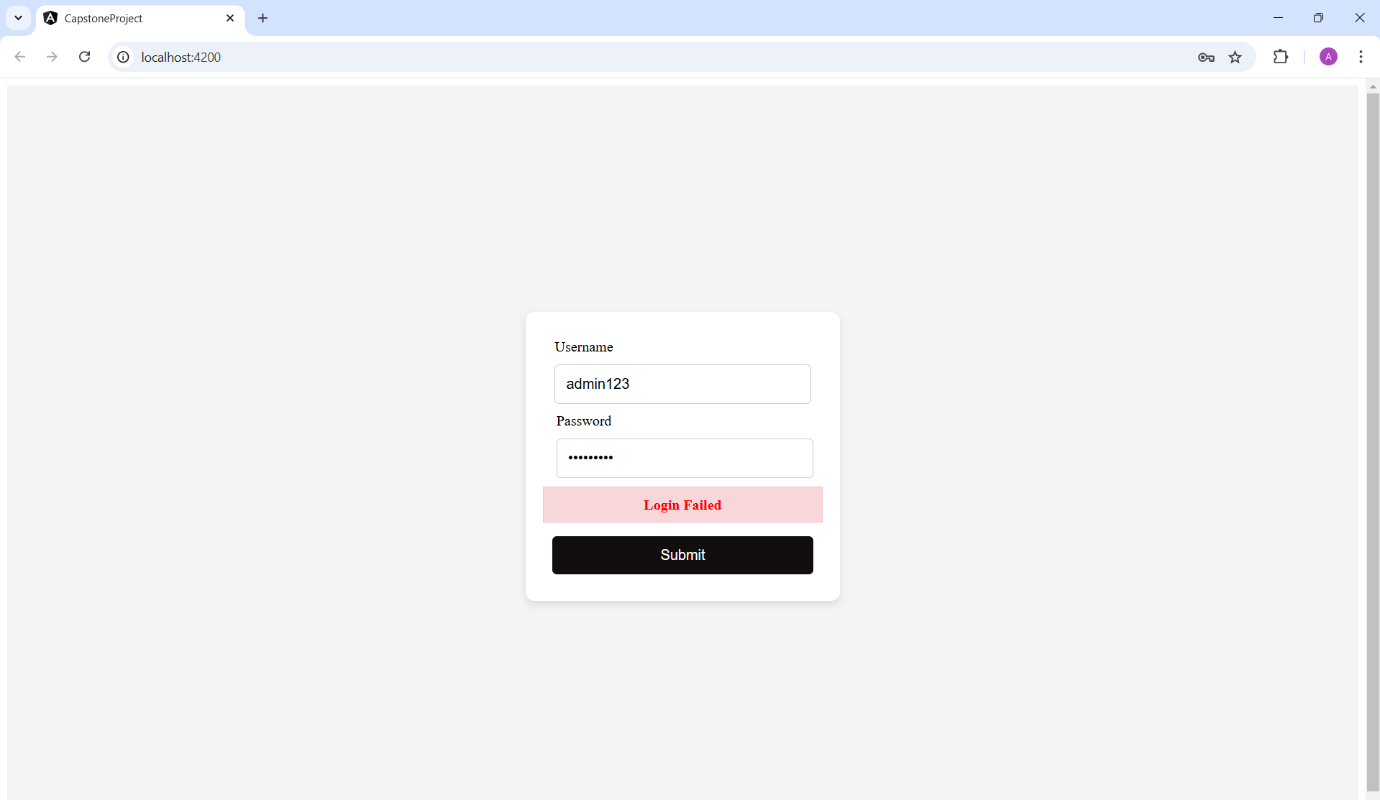
    }

}

OUTPUT:







Task 5: **Angular CRUD Application with JSON**

1. Admin can create a user. (10 Marks)
2. Admin can view a user's details. (10 Marks)
3. Admin can update existing user details. (10 Marks)
4. Admin can delete a user. (10 Marks)
5. Validation. (5 Marks)

OUTPUT:

