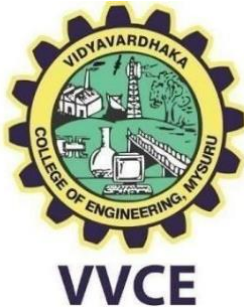


Vidyavardhaka College of Engineering

**Autonomous Institute, Affiliated to Visveswaraya Technological University, Belagavi.
Accredited by NBA, New Delhi (2017-18 to 2019-20) & NAAC with 'A' Grade Gokulam
3rd Stage, Mysuru - 570002, Karnataka, India**



Activity Based Assessment

**LOGIN PAGE
APPOINTMENT PAGE AND
FEEDBACKFORM PAGE**

Submitted by

AISHWARYA H R - 4VV21IS005

S MEGHANA REDDY-4VV21IS087

Submitted to

KASTURI RANGAN

Assistant Professor Department of ISE

FULLSTACK DEVELOPMENT 1

[21IS551]

2023-2024

TABLE OF CONTENT

Sl.no	Title	Pg.no
1	Introduction	3
2	Login page	4
3	Appointment Page	8
4	Feedback form page	12

1.INTRODUCTION

1. Login Page:

Imagine a secure entry point where users can access the application using their registered credentials. The login page typically comprises a form with fields for username or email address and password. Upon receiving user input, the application submits the form data to the server for authentication. If successful, the user is granted access to the relevant sections of the application. Conversely, if the credentials are incorrect or there are network issues, error messages guide the user towards a solution. It's crucial to prioritize robust security measures like using secure communication protocols and implementing strong password practices to protect user data.

2. Appointment Page:

This page empowers users to schedule appointments for specific services offered by the application. The appointment form usually includes options like selecting a service from a dynamic dropdown menu (populated with data retrieved from the server), choosing a date using a calendar or date picker, and selecting a time slot from another dropdown that displays available options based on the chosen service and date. Some applications might integrate a calendar for visual appointment selection, enhancing user experience. To ensure smooth scheduling, the application might perform an availability check before submitting the appointment, notifying the user if the selected time slot is unavailable.

3. Feedback Form Page:

The feedback form page allows users to express their opinions and experiences with the application. It typically consists of a simple form with components like a rating system (e.g., star rating) and an optional text area for further comments. Upon submitting the form, the application transmits the feedback data to the server, where it can be analyzed to improve the user experience and service quality. This valuable feedback can help the application evolve and cater to the needs of its users more effectively.

2.LOGIN PAGE

App.model.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ReactiveFormsModule } from '@angular/forms';
import { ProjectComponent } from '../project/project.component';

@NgModule({
  declarations: [
    ProjectComponent
  ],
  imports: [
    BrowserModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [ProjectComponent]
})
export class AppModule { }
```

project.component.css

```
body {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  margin: 0;
  background-color: #f4f4f4;
  /* Optional background color */
}

form {
  width: 300px;
  /* Adjust the width as needed */
  padding: 20px;
  background-color: white;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

form div {
  margin-bottom: 15px;
}
```

```

label {
  display: block;
  margin-bottom: 5px;
}

input {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
  border: 1px solid #ccc;
  border-radius: 3px;
}

button {
  background-color: #4caf50;
  color: white;
  padding: 10px 15px;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}

button:disabled {
  background-color: #aaa;
  cursor: not-allowed;
}

```

Project.component.html

```

<h1>
  LOGIN
</h1>
<form [formGroup]="projectForm" (ngSubmit)="submitForm()">
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" formControlName="name">
  </div>
  <div>
    <label for="phone">Phone:</label>
    <input type="number" id="phone" formControlName="phone">
  </div>
  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" formControlName="email">
  </div>
  <div>
    <label for="date">Date of birth:</label>
    <input type="date" id="date" formControlName="date">
  </div>

```

```

    </div>
    <div>
      <label for="password">Password:</label>
      <input type="password" id="password" formControlName="password">
    </div>

    <button type="submit" [disabled]="!projectForm.valid">Submit</button>
  </form>

```

Project.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({
  selector: 'my-app',
  templateUrl: './project.component.html',
  styleUrls: ['./project.component.css']
})
export class ProjectComponent implements OnInit {
  projectForm: FormGroup;

  constructor(private formBuilder: FormBuilder) { }

  ngOnInit() {
    this.projectForm = this.formBuilder.group({
      name: ['', Validators.required],
      phone: ['', Validators.required],
      email: ['', Validators.required],
      date: ['', Validators.required],
      password: ['', Validators.required]
    });
  }

  submitForm() {
    if (this.projectForm.valid) {
      console.log('Form submitted:', this.projectForm.value);
    }
  }
}

```

Snapshots:

LOGIN

Name:

Phone:

Email:

Date of birth:

Password:

Submit

LOGIN

Name:

Aishwarya H R

Phone:

0808231933

Email:

aishwaryagowda1503@gmail.com

Date of birth:

15-03-2003

Password:

Submit

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

Recorder

Filter

Default levels

48 Issues: 1 47 1 hidden

workerMain.js:7

9dab468...sb.js:1

console.ts:15

project.component.ts:30

project.component.ts:30

project.component.ts:30

project.component.ts:30

⚠ Duplicate definition of module 'typescript'

Console was cleared

Angular is running in development mode. Call enableProdMode() to enable production mode.

Form submitted: ▾ Object {
 date: "2003-03-15"
 email: "aishwaryagowda1503@gmail.com"
 name: "Aishwarya H R"
 password: "123aishu"
 phone: 8088231933
 [[Prototype]]: Object
}

Form submitted: ▾ Object {
 date: "2003-03-15"
 email: "aishwaryagowda1503@gmail.com"
 name: "Aishwarya H R"
 password: "123aishu"
 phone: 8088231933
 [[Prototype]]: Object
}

Form submitted: ▾ Object {
 date: "2003-03-15"
 email: "aishwaryagowda1503@gmail.com"
 name: "Aishwarya H R"
 password: "123aishu"
 phone: 8088231933
 [[Prototype]]: Object
}

Form submitted: ▾ Object {
 date: "2003-03-15"
 email: "aishwaryagowda1503@gmail.com"
 name: "Aishwarya H R"
 password: "123aishu"
 phone: 8088231933
 [[Prototype]]: Object
}

Windows Update

Your device will restart to update outside of active hours (estimate: 3 min)

Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.

3.Appointment Page

App.model.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ReactiveFormsModule } from '@angular/forms';
import { AppointmentComponent } from '../appointment/appointment.component';

@NgModule({
  declarations: [
    AppointmentComponent
  ],
  imports: [
    BrowserModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppointmentComponent]
})
export class AppModule { }
```

appointment.component.css

```
body {
  background-color: cyan;
}
form {
  max-width: 400px;
  margin: 0 auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  background-color: #e0f7fa;
}
label {
  display: block;
  margin-bottom: 5px;
}
input[type="text"],
input[type="date"],
input[type="time"] {
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
button {
  background-color: #007bff;
```



```
    color: #fff;
    border: none;
    border-radius: 5px;
    padding: 10px 20px;
    cursor: pointer;
}
button:hover {
    background-color: #0056b3;
}
```

Appointment.component.html

```
body {
    background-color: cyan;
}
form {
    max-width: 400px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #e0f7fa;
}
label {
    display: block;
    margin-bottom: 5px;
}
input[type="text"],
input[type="date"],
input[type="time"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}
button {
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 5px;
    padding: 10px 20px;
    cursor: pointer;
}
button:hover {
    background-color: #0056b3;
}
```

Appointment.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';

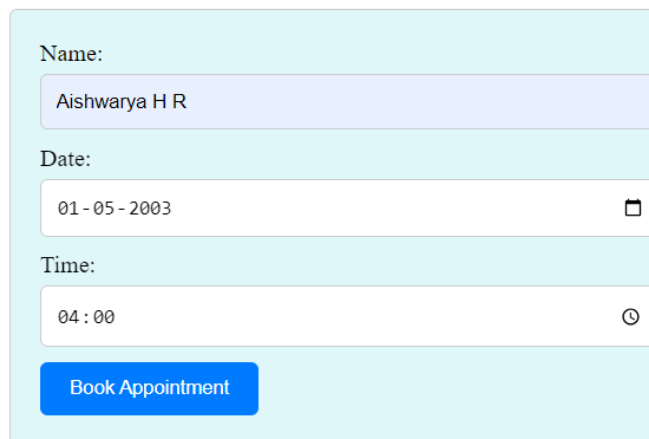
@Component({
  selector: 'my-app',
  templateUrl: './appointment.component.html',
  styleUrls: ['./appointment.component.css']
})
export class AppointmentComponent implements OnInit {
  appointmentForm: FormGroup;

  constructor(private formBuilder: FormBuilder) { }

  ngOnInit() {
    this.appointmentForm = this.formBuilder.group({
      name: ['', Validators.required],
      date: ['', Validators.required],
      time: ['', Validators.required]
    });
  }

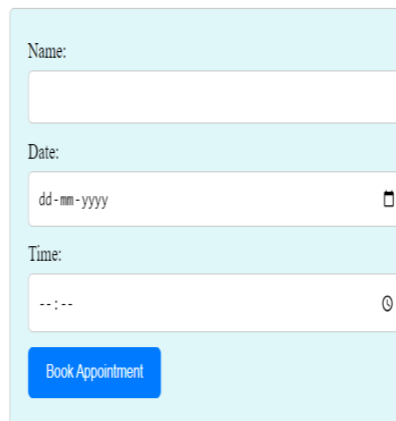
  submitForm() {
    if (this.appointmentForm.valid) {
      console.log('Form submitted:', this.appointmentForm.value);
    }
  }
}
```

Snapshots:



A screenshot of a web form for booking an appointment. The form is light blue and contains three input fields: 'Name' with the value 'Aishwarya H R', 'Date' with the value '01-05-2003', and 'Time' with the value '04:00'. Each input field has a small icon to its right (a calendar for date and a clock for time). Below the inputs is a blue button labeled 'Book Appointment'.

⌕ ↻ /



A screenshot of the same appointment form, but it is broken. The 'Name' field is empty. The 'Date' field shows the placeholder 'dd-mm-yyyy' and the 'Time' field shows '--:--'. The 'Book Appointment' button is still present.

⚠ Duplicate definition of module 'typescript' [workerMain.js:7](#)

Console was cleared [9dab468...sb.js:1](#)

Angular is running in development mode. Call enableProdMode() to enable production mode. [console.ts:15](#)

Form submitted: ▼ Object [i](#) [appointment.component.ts:28](#)

```
date: "2003-05-01"
name: "Aishwarya H R"
time: "04:00"
```

4.Feedback Form Page

App.model.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ReactiveFormsModule } from '@angular/forms';
import { AppComponent } from './app.component';
import { FeedbackComponent } from './feedback/feedback.component';

@NgModule({
  declarations: [
    AppComponent,
    FeedbackComponent
  ],
  imports: [
    BrowserModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html

```
<div style="text-align: center;">
  <h1>Welcome to Feedback Form App!</h1>
  <app-feedback></app-feedback>
</div>
```

App.component.ts

```
import { Component, VERSION } from '@angular/core';

@Component({
  selector: 'my-app',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent { }
```

Feedback.component.css

```
body {
  background-color: cyan;
}

form {
  max-width: 400px;
  margin: 0 auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
  background-color: #8af2f7;
  box-shadow: 0 4px 8px rgb(4, 242, 255);
}

label {
  display: block;
  margin-bottom: 8px;
  font-weight: bold;
}

input[type="text"],
input[type="email"],
textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
}

button {
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 5px;
  padding: 10px 20px;
  cursor: pointer;
  font-size: 16px;
}

button:hover {
  background-color: #0056b3;
}

.error-message {
  color: #dc3545;
  font-size: 14px;
```

```

    margin-top: 5px;
}

.success-message {
    color: #28a745;
    font-size: 16px;
    margin-top: 10px;
}

```

Feedback.component.html

```

<form [formGroup]="feedbackForm" (ngSubmit)="submitFeedback()">
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" formControlName="name">
    <div class="error-message" *ngIf="feedbackForm.get('name').invalid &&
feedbackForm.get('name').touched">
      Name is required
    </div>
  </div>
  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" formControlName="email">
    <div class="error-message" *ngIf="feedbackForm.get('email').invalid &&
feedbackForm.get('email').touched">
      Email is required and must be valid
    </div>
  </div>
  <div>
    <label for="message">Message:</label>
    <textarea id="message" formControlName="message"></textarea>
    <div class="error-message" *ngIf="feedbackForm.get('message').invalid &&
feedbackForm.get('message').touched">
      Message is required
    </div>
  </div>
  <button type="submit" [disabled]="!feedbackForm.valid">Submit Feedback</button>
</form>

```

Feedback.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({
  selector: 'app-feedback',
  templateUrl: './feedback.component.html',
  styleUrls: ['./feedback.component.css']
})
export class FeedbackComponent implements OnInit {
  feedbackForm: FormGroup;

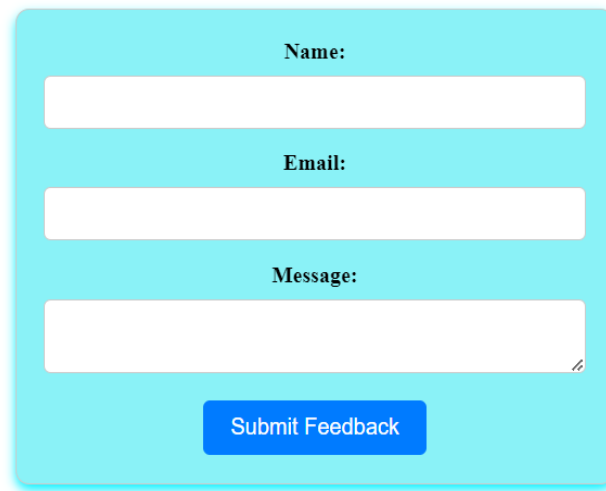
  constructor(private formBuilder: FormBuilder) { }

  ngOnInit() {
    this.feedbackForm = this.formBuilder.group({
      name: ['', Validators.required],
      email: ['', [Validators.required, Validators.email]],
      message: ['', Validators.required]
    });
  }

  submitFeedback() {
    if (this.feedbackForm.valid) {
      console.log('Feedback submitted:', this.feedbackForm.value);
    }
  }
}
```

Snapshots:

Welcome to Feedback Form App!



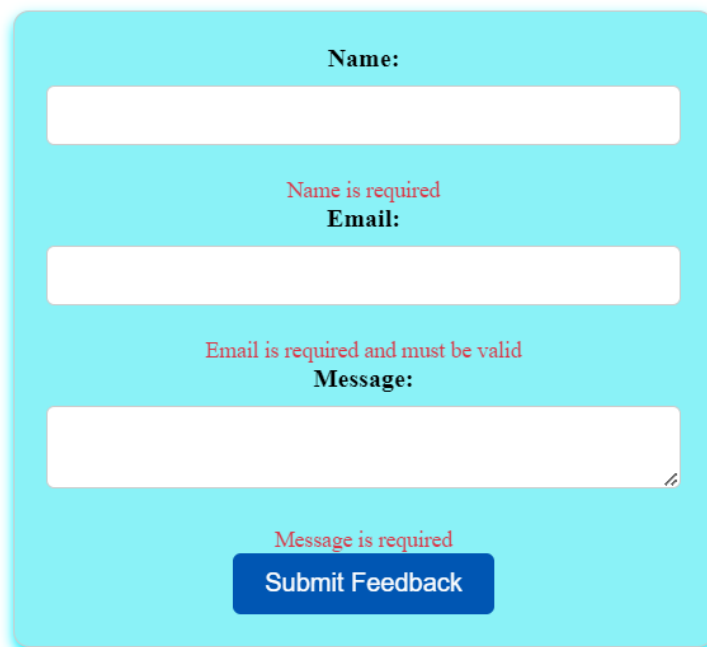
Name:

Email:

Message:

Submit Feedback

Welcome to Feedback Form App!



Name:

Name is required

Email:

Email is required and must be valid

Message:

Message is required

Submit Feedback

Welcome to Feedback Form App!

Name:

Email:

Message:

Console was cleared

Angular is running in development mode. Call enableProdMode() to enable production mode.

Feedback submitted: ▼ Object  

```
  email: "aishwaryagowda1503@gmail.com"
  message: "hello"
  name: "Aishwarya H R"
  ► [[Prototype]]: Object
```

>

