

TEMPLATE DRIVEN FORMS

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1	Project Abstract	3
2	Problem Statement	3
3	Proposed Template driven forms Application Wireframe	4
3.1	Screenshots	4
4	Business-Requirement:	6
5	Constraints	8
6	Mandatory Assessment Guidelines	9

1 PROJECT ABSTRACT

Form validation is a foundational feature in web applications, ensuring data integrity and enhancing user experience. This assignment focuses on implementing **Template-Driven Forms with Validations** in Angular. Template-driven forms offer a declarative way to handle form inputs, with logic primarily defined in the HTML template.

The objective is to create a user registration form with validations embedded directly in the template using Angular's form directives, providing immediate feedback and ensuring correct user input.

2 PROBLEM STATEMENT

You are tasked with building a **Template-Driven Single Page Application** in Angular.

The application must:

- Present a registration form with multiple fields.
- Apply built-in validations using Angular directives.
- Provide immediate, real-time feedback as the user interacts with the form.
- Restrict form submission until all fields are valid.

3 PROPOSED TEMPLATE DRIVEN FORMS APPLICATION WIREFRAME

UI needs improvisation and modification as per given use case and to make test cases passed.

3.1 SCREENSHOTS

A screenshot of a web browser window at localhost:4200/. The page title is "Template Driven Form with Validations". The form contains the following fields and a button:

- Name:
- Email:
- Age:
- Password:
- Confirm Password:
- Submit:

Validations

A screenshot of the same web browser window showing the form with validation error messages in red text:

- Name:
Name is required and must be at least 3 characters long.
- Email:
Enter a valid email address.
- Age:
Age is required and must be 18 or older.
- Password:
Password is required and must be at least 6 characters long.
- Confirm Password:
Password confirmation is required and must match the password.
- Submit:

localhost:4200/

Template Driven Form with Validations

Name

Name is required and must be at least 3 characters long.

Email

Enter a valid email address.

Age

Age is required and must be 18 or older.

Password

Password is required and must be at least 6 characters long.

Confirm Password

Password confirmation is required and must match the password.
Passwords do not match.

Submit

Valid Entries

localhost:4200/

Template Driven Form with Validations

Name

Email

Age

Password

Confirm Password

Submit

4 BUSINESS-REQUIREMENT:

As an application developer, develop the Template driven forms (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	<p>As a user I should be able to visit the welcome page as the default page.</p> <p>Acceptance criteria:</p> <p>Template Driven Form with Validations</p> <p>User Story</p> <p>As a user, I want to fill out a form with real-time validation messages using Angular's template-driven form features so that I can be guided to submit correct information.</p> <p>Acceptance Criteria</p> <ol style="list-style-type: none">1. Display a heading: "Template Driven Form with Validations"2. Form fields required:<ul style="list-style-type: none">• Name• Email• Age• Password• Confirm Password3. Apply validation rules:<ul style="list-style-type: none">- Show error messages only when the field is invalid and has been touched- Confirm Password must match Password4. The Submit button is disabled until the form is valid5. On successful submission, log the form values to the console <p>Component Overview</p> <p>Selector: <code>app-root</code></p> <p>Template Used: <code>app.component.html</code></p> <p>Form Setup (Template-Driven)</p> <ol style="list-style-type: none">1. Uses Angular directives such as <code>ngModel</code>, <code>required</code>, <code>minlength</code>, <code>email</code>, <code>min</code>.2. Form controls are bound directly to the template using two-way binding (<code>ngModel</code>).3. Validation states (<code>invalid</code>, <code>touched</code>) are monitored using local template reference variables (e.g., <code>#name="ngModel"</code>).

		<h2>Method - onSubmit()</h2> <ol style="list-style-type: none"> 1. Triggered via (<code>ngSubmit</code>) when the form is submitted. 2. If <code>form.valid</code>: <ul style="list-style-type: none"> • Logs form data to the console. 3. If <code>form.invalid</code>: <ul style="list-style-type: none"> • Logs a validation failure message. <h2>HTML Structure (app.component.html)</h2> <ul style="list-style-type: none"> • Form Container: <code><div class="container"></code> • Heading: <code><h2></code> displaying "Template Driven Form with Validations" • Form Element: <code><form></code> with form reference and submit handler <h2>Form Fields</h2> <p>Each input is inside a <code><div class="form-group"></code> and includes:</p> <ul style="list-style-type: none"> • A <code><label></code> for the field • An <code><input></code> element using <code>ngModel</code> with validation attributes • A <code><div class="text-danger"></code> that conditionally shows validation messages <h2>Field Descriptions</h2> <ol style="list-style-type: none"> 1. Name Field (<code>name</code>) with label "Name" <ul style="list-style-type: none"> • Required • Minimum 3 characters • Error message: "Name is required and must be at least 3 characters long." 2. Email Field (<code>email</code>) with label "Email" <ul style="list-style-type: none"> • Required • Must be in valid email format • Error message: "Enter a valid email address." 3. Age Field (<code>age</code>) with label "Age" <ul style="list-style-type: none"> • Required • Must be 18 or older • Error message: "Age is required and must be 18 or older." 4. Password Field (<code>password</code>) with label "Password" <ul style="list-style-type: none"> • Required • Minimum 6 characters • Error message: "Password is required and must be at least 6 characters long." 5. Confirm Password Field (<code>confirmPassword</code>) with label "Confirm Password" <ul style="list-style-type: none"> • Required • Minimum 6 characters • Must match the Password field
--	--	--

		<ul style="list-style-type: none"> • Error message: <ul style="list-style-type: none"> → “Password confirmation is required and must match the password.” → If mismatched: “Passwords do not match.” <h3>Password Match Check</h3> <p>To ensure the Confirm Password input matches the Password, a manual comparison is performed within the template using Angular's template reference variables.</p> <ul style="list-style-type: none"> • The application tracks the values of both password and confirmPassword inputs. • If the values do not match and the Confirm Password field has been interacted with (touched), a validation message is displayed. • This message, “<i>Passwords do not match,</i>” appears dynamically beneath the Confirm Password field, providing immediate feedback to the user. • The check only activates after the user has interacted with the Confirm Password input, maintaining a clean UI until necessary. <h3>Submit Button</h3> <ul style="list-style-type: none"> • Labeled 'Submit' • Disabled until the form is fully valid • Submits the form when all fields pass validation <h3>Dynamic Behavior</h3> <ul style="list-style-type: none"> • Angular evaluates field validity in real-time. • Error messages are shown only after the user has interacted with (touched) a field. • Password mismatch is dynamically evaluated and displayed in real-time. • Submit button activates only when all form fields are valid. <p>** Kindly refer to the screenshots for any clarifications. **</p>

5 CONSTRAINTS

1. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.

6 MANDATORY ASSESSMENT GUIDELINES

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. This editor Auto Saves the code.
4. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
5. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

Note: The application will not run in the local browser

6. You can follow series of command to setup Angular environment once you are in your project-name folder:
 - a. npm install -> Will install all dependencies -> takes 10 to 15 min.
 - b. npm run start -> To compile and deploy the project in browser. You can press the <Ctrl> key while clicking on localhost:4200 to open the project in the browser -> takes 2 to 3 min.
 - c. npm run test -> to run all test cases. **It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.**
7. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on **"Submit Assessment"** after you are done with code.