System Requirements Specification Index

For

Political Party System

Version 4.0

IIHT Pvt. Ltd.
IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India
fullstack@iiht.com

Political Parties APPLICATION
System Requirements Specification

1.Business-Requirement:

1.1 Problem Statement:

Political Parties Application is .Net Core web API 3.1 application integrated with MS SQL Server , where it allows the management of political parties, political leaders and developments done by political leaders in the states.

1.2 Following is the requirement specification:

	Political Parties Application
Modules	
1	Political Party
2	Political Leader
3	Development
Political Party Module Functionalities	
1	Register a Political Party
2	Update the existing Political Party
3	Get a Political Party by Id
4	Fetch all registered Political Parties
5	Delete an existing Political Party
Political Leader Module Functionalities	
1	Register a Political Leader
2	Update the existing Political Leader
3	Get a Political Leader by Id
4	Fetch all registered Political Leaders
5	Delete an existing Political Leader
6	Fetch all Political Leaders registered with a Party

Development Module Functionalities	
1	Create a Development Plan
2	Update the existing Development
3	Get a Development by Id
4	Fetch all created developments
5	Delete an existing Development
6	Fetch all Developments created for a Political Leader

2. Assumptions, Dependencies, Risks / Constraints

2.1 Political Party Constraints:

- While deleting the Political Party, if politicalPartyId does not exist then the operation should throw a custom exception.
- While fetching the political party details by id, if politicalPartyId does not exist then the operation should throw a custom exception.

2.2 Political Leader Constraints

- While deleting the political leader, if politicalLeaderId does not exist then the operation should throw a custom exception.
- While fetching the political leader details by id, if politicalLeaderId does not exist then the operation should throw a custom exception.
- While fetching all the political leader details by political party id, if politicalPartyId does not exist then the operation should throw a custom exception.

2.3 Developments Constraints

- While deleting the development, if developmentId does not exist then the operation should throw a custom exception.
- While fetching the development details by id, if developmentId does not exist then the operation should throw a custom exception.
- While fetching all the developments created for a political leader, if politicalLeaderId does not exist then the operation should throw a custom exception.

2.4 Common Constraints

- For all rest endpoints receiving @RequestBody, validation check must be done and must throw custom exception if data is invalid
- All the business validations must be implemented in model classes only.

- All the database operations must be implemented on entity object only
- Do not change, add, remove any existing methods in service layer
- In Repository interfaces, custom methods can be added as per requirements.
- All RestEndpoint methods and Exception Handlers must return data wrapped in **ResponseEntity**

3. Business Validations

3.1 Political Party Class Entities

- Political Party Id (long)is not null,Key attribute.
- Political Party name (string) is not null,min 3 and max 100 characters.
- Political party founder name (string) is not null, min 3 and max 100 characters.

3.2 Political Leader Class Entities

- Political Leader Id (long) is not null,Key attribute.
- Political Leader candidate name (string) is not null, min 3 and max 100 characters.
- Political Leader state name (string) is not null, min 3 and max 100 characters.

3.3 Development Entities

- Development Id (long) is not null,min 3 and max 100 characters.
- Development title(string) is not null, min 3 and max 100 characters.
- Development activity (string)is not null, min 3 and max 100 characters.
- Development budget(decimal) is not null, min 3 and max 100 characters.
- Development state is (string)not null, min 3 and max 100 characters.
- Development activity month(int) is not null, the range is from 1 to 12
- Development activity year(int) is not null, the range is from 2021 to 2040.

4. Considerations

- There is no roles in this application
- You can perform the following 3 possible actions

PoliticalParty
PoliticalLeader
Development

5. Rest Endpoints

Rest End-points to be exposed in the controller along with method details for the same to be created

5.1 PoliticalPartyController

<table>
<tr><th>URL Exposed</th><th>Purpose</th></tr>
<tr><td>/parties
<table><tr><td>Http Method</td><td>POST</td></tr><tr><td>Parameter 1</td><td>RegisterPoliticalParty ViewModel model</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Register a political party</td></tr>
<tr><td>/parties
<table><tr><td>Http Method</td><td>PUT</td></tr><tr><td>Parameter 1</td><td>RegisterPoliticalParty ViewModel model</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Update a political party</td></tr>
<tr><td>/parties
<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>-</td></tr><tr><td>Return</td><td><IEnumerable<Political Party>></td></tr></table></td><td>Fetches the list of all registered Political Parties</td></tr>
<tr><td>/parties/{politicalPartyId}
<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>Long (politicalPartyId)</td></tr><tr><td>Return</td><td><PoliticalParty></td></tr></table></td><td>Fetches the details of a political party</td></tr>
<tr><td>/parties /{politicalPartyId}
<table><tr><td>Http Method</td><td>DELETE</td></tr><tr><td>Parameter 1</td><td>Long (politicalPartyId)</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Delete a political party</td></tr>
</table>

5.2 PoliticalLeaderController

<table>
<tr><th>URL Exposed</th><th>Purpose</th></tr>
<tr><td>/leaders
<table><tr><td>Http Method</td><td>POST</td></tr><tr><td>Parameter 1</td><td>RegisterPoliticalLeaderViewModel model</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Register a political leader</td></tr>
<tr><td>/leaders
<table><tr><td>Http Method</td><td>PUT</td></tr><tr><td>Parameter 1</td><td>RegisterPoliticalLeaderViewModel model</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Update a political leader</td></tr>
<tr><td>/leaders
<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>-</td></tr><tr><td>Return</td><td><IEnumerable<PoliticalLeader>></td></tr></table></td><td>Fetches all registered political leaders</td></tr>
<tr><td>/leaders/{politicalLeaderId}
<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>Long (politicalLeaderId)</td></tr><tr><td>Return</td><td><PoliticalLeader></td></tr></table></td><td>Fetch the details of a political leader</td></tr>
<tr><td>/leaders/by-party-id/{politicalPartyId}
<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>Long (politicalPartyId)</td></tr><tr><td>Return</td><td><PoliticalLeader></td></tr></table></td><td>Fetches the details of all the political leaders belongs to a party</td></tr>
<tr><td>/leaders/{politicalPartyId}
<table><tr><td>Http Method</td><td>DELETE</td></tr><tr><td>Parameter 1</td><td>Long (politicalPartyId)</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Delete a political leader from the existing leaders</td></tr>
</table>

5.3 DevelopmentController

<table>
<tr><th>URL Exposed</th><th>Purpose</th></tr>
<tr><td>/developments<table><tr><td>Http Method</td><td>POST</td></tr><tr><td>Parameter 1</td><td>RegisterDevelopmentViewModel model</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Register a development plan</td></tr>
<tr><td>/developments<table><tr><td>Http Method</td><td>PUT</td></tr><tr><td>Parameter 1</td><td>RegisterDevelopmentViewModel model</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Update an existing development plan</td></tr>
<tr><td>/developments<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>-</td></tr><tr><td>Return</td><td><IEnumerable<Development>></td></tr></table></td><td>Fetches all the registered developments</td></tr>
<tr><td>/developments/{developmentId}<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>Long(developmentId)</td></tr><tr><td>Return</td><td><Development></td></tr></table></td><td>Fetch the details of a development plan</td></tr>
<tr><td>/developments/by-leader-id/{politicalLeaderId}<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>Long(politicalLeaderId)</td></tr><tr><td>Return</td><td><Development></td></tr></table></td><td>Fetches all the development plans created for a political leader</td></tr>
<tr><td>/developments/{developmentId}<table><tr><td>Http Method</td><td>DELETE</td></tr><tr><td>Parameter 1</td><td>Long(developmentId)</td></tr><tr><td>Return</td><td>HTTP Response StatusCode</td></tr></table></td><td>Deletes an existing development plan</td></tr>
</table>

6. Template Code Structure

6.1 Package: PoliticalParties

Resources

Names	Resource	Remarks	Status
Package Structure			
controller	PoliticalPartyController PoliticalLeaderController DevelopmentController	Controller class to expose all rest-endpoints for auction related activities.	Partially implemented
Startup.cs	Startup CS file	Contain all Services settings and SQL server Configuration.	Already Implemented
Properties	launchSettings.json file	All URL Setting for API	Already Implemented
	appsettings.json	Contain connection string for database	Already Implemented

6.2 Package: PoliticalParties.BusinessLayer

Resources

Names	Resource	Remarks	Status
Package Structure			
Interface	IPoliticalPartyServices interface IPoliticalLeaderServices interface IDevelopmentServices interface	Inside all these interface files contains all business validation logic functions.	Already implemented

Service	PoliticalParty Services CS file PoliticalLeader Services CS file Development Services CS file	Using this all class we are calling the Repository method and use it in the program and on the controller.	Partially implemented
Repository	IPoliticalPartyRepository PoliticalParty Repository IPoliticalLeaderRepository PoliticalLeader Repository IDevelopmentRepository Development Repository (CS files and interfaces)	All these interfaces and class files contain all CRUD operation code for the database. Need to provide implementation for service related functionalities	Partially implemented
ViewModels	RegisterPoliticalLeaderViewModel RegisterPoliticalPartyViewModel RegisterDevelopmentViewModel	Contain all view Domain entities for show and bind data. All the business validations must be implemented.	Partially implemented

6.3 Package: PoliticalParties.DataLayer

Resources

Names	Resource	Remarks	Status
Package Structure			
DataLayer	PoliticalPartiesDBContext cs file	All database Connection,collection setting class	Already Implemented

6.4 Package: PoliticalParties.Entities

Resources

Names	Resource	Remarks	Status
Package Structure			
Entities	PoliticalParty PoliticalLeader Development Status (CS files)	All Entities/Domain attribute are used for pass the data in controller and status entity to return response Annotate this class with proper annotation to declare it as an entity class with **Id** as primary key. Generate the **Id** using the **IDENTITY** strategy	Partially implemented

7. Execution Steps to Follow

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 the Application menu (Three horizontal lines at left top) Terminal → New Terminal.

3. On command prompt, cd into your project folder (cd <Your-Project-folder>).

4. To connect SQL server from terminal:
   (PoliticalParties /sqlcmd -S localhost -U sa -P pass@word1)
   - To create database from terminal -
     1> Create Database PoliticalParty_Db
     2> Go

5. Steps to Apply Migration(Code first approach):
   - Press Ctrl+C to get back to command prompt
   - Run following command to apply migration-
     (PoliticalParties /dotnet-ef database update)

6. To check whether migrations are applied from terminal:
   (PoliticalParties /sqlcmd -S localhost -U sa -P pass@word1)

   1> Use PoliticalParty_Db
   2> Go
   1> Select * From __EFMigrationsHistory
   2> Go

7. To build your project use command:
   (PoliticalParties /dotnet build)

8. To launch your application, Run the following command to run the application:
   (PoliticalParties /dotnet run)

9. This editor Auto Saves the code.

10. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.

11. **To test web-based applications 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

12. **To run the test cases in CMD, Run the following command to test the application: (PoliticalParties /dotnet test --logger "console;verbosity=detailed")**
    (You can run this command multiple times to identify the test case status,and refactor code to make maximum test cases passed before final submission)

13. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B - command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**

14. **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.**

15. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**

Health Club

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Problem Statement

2 Proposed Health Club Wireframe

2.1 Welcome page

2.2 View Appointments

2.3 Place Appointment

2.4 QUERY

3 Business-Requirement:

4 Constraints

5 Mandatory Assessment Guidelines

1 Problem Statement

Health Club is SPA (Single Page Application) for placing a request for appointment with personal physical trainer, manage appointments and dropping a query

The core modules of Health Club app are:

1. Welcome Page
2. Apply for an appointment
3. View Appointment
4. Query

2 Proposed Health Club Wireframe

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

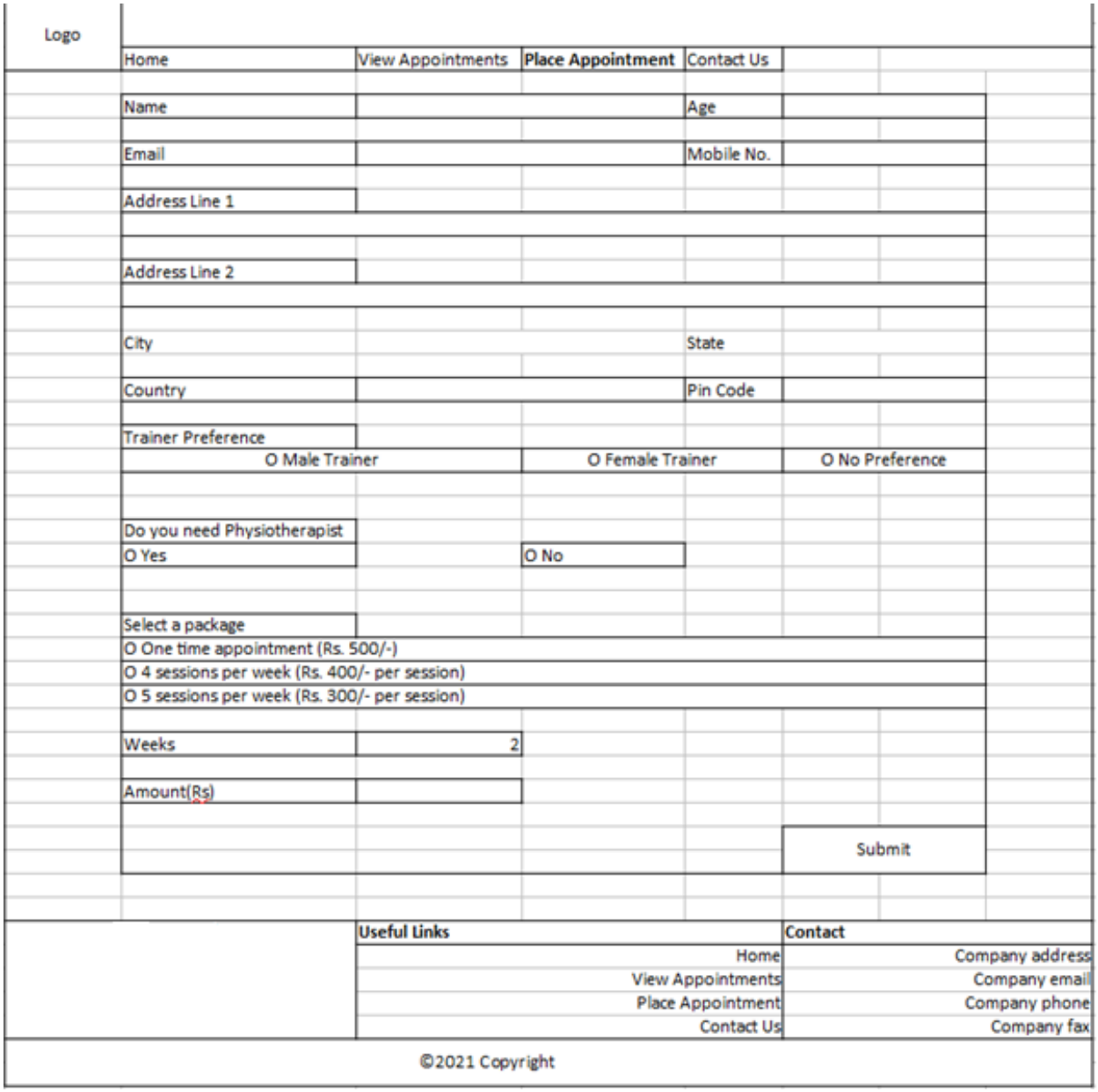
2.1 Welcome page

Logo				
	Home	View Appointments	Place Appointment	Contact Us
Introduction Text (any)				
	Useful Links		**Contact**	
	Home		Company address	
	View Appointments		Company email	
	Place Appointment		Company phone	
	Contact Us		Company fax	
©2021 Copyright **GET-FIT Health Club**				

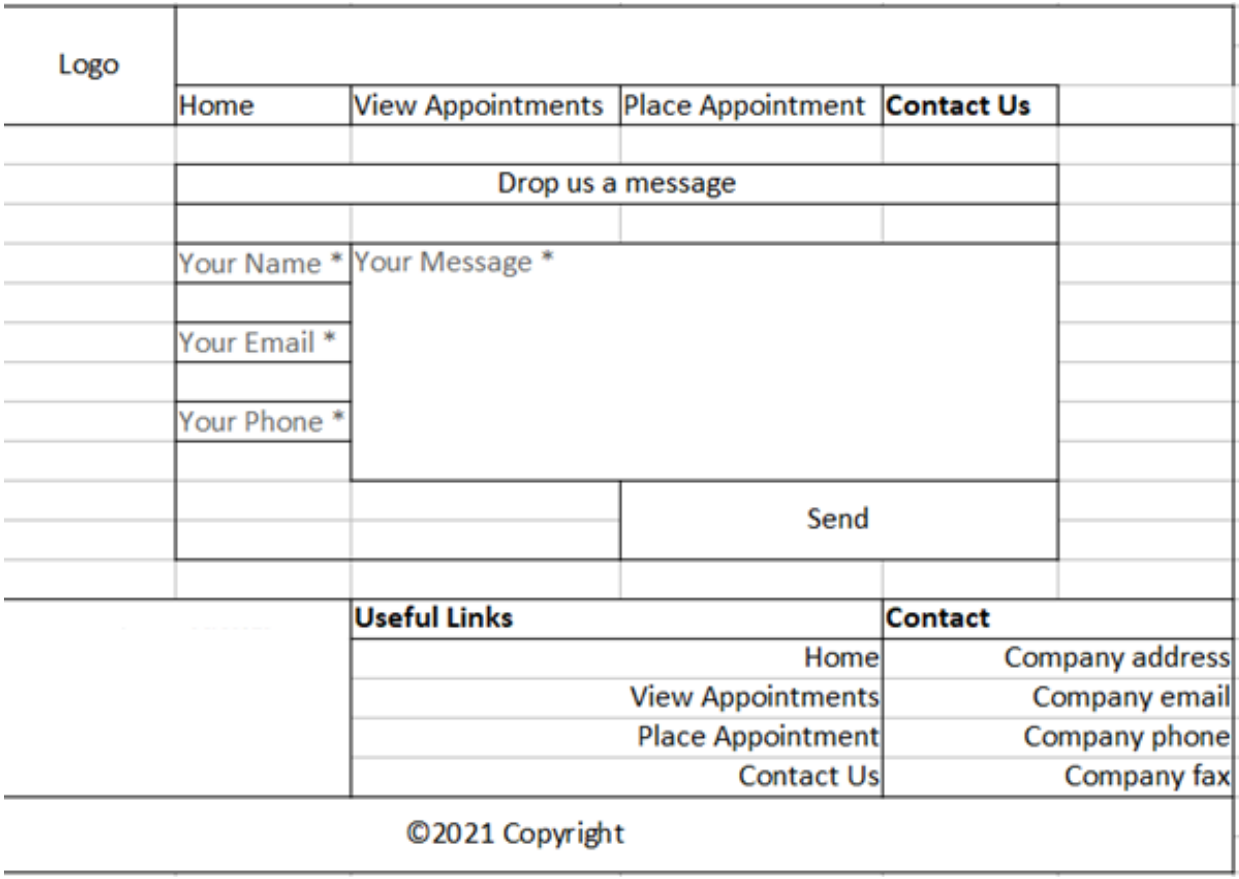
2.2 View Appointments

Logo									
	Home	**View Appointments**	Place Appointment		Contact Us				
S.No.	Name	Phone	email	Age	Complete Address	Trainer Preference	Physio Required	Package	Total Amount
		Useful Links				**Contact**			
		Home				Company address			
		View Appointments				Company email			
		Place Appointment				Company phone			
		Contact Us				Company fax			
©2021 Copyright									

2.3 Place Appointment



2.4 QUERY


Logo
Home
View Appointments
Place Appointment
Contact Us
Drop us a message
Your Name *
Your Message *
Your Email *
Your Phone *
Send
Useful Links
Home
View Appointments
Place Appointment
Contact Us
Contact
Company address
Company email
Company phone
Company fax
©2021 Copyright

3 Business-Requirement:

As an application developer, develop the Health Club App (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as default page. Acceptance criteria: 1. User can click any button given in the menu bar.
US_02	Post Appointment	As a user I should be able to post an appointment Acceptance criteria: 1. As a user I should be able to furnish following details at the time of placing an appointment 1.1 Name 1.2 Age 1.3 Email 1.4 Mobile No 1.5 Address Line 1 1.6 Address Line 2 1.7 City 1.8 State 1.9 Country 1.10 Pin Code 1.11 Trainer Preference 1.12 Physiotherapist requirement (Yes or No) 1.13 Select a package 1.14 Weeks 1.15 Amount 2. Weeks number type input box should be visible when 2nd or 3rd package option is selected.

		3. If physiotherapist is required add additional 200/- in final amount 4. Amount should be disabled and should be calculated automatically based on the selected package. 2. All details fields must be mandatory. 3. Address line 2 may contain the same address as address line 1. 4. Email& Mobile must be unique. 5. If any constraint is not satisfied, a validation message must be shown. 6. A success or failure message should be visible after the submit button is clicked.
US_03	Manage Appointment	1. As a user I should be able to view all appointment requests, and after selecting any appointment Acceptance criteria: 1. Message should be visible if no appointment is available to show.
US_04	Query	As a user I should be able to post a feedback/query/message Acceptance criteria: 1. As a user I should be able to furnish following details at the time of filling contact us form a. Name b. Email c. Phone d. Message 2. Message should not go beyond 200 characters. 3. All four fields must be mandatory. 4. A success or failure message should be visible after the submit button is clicked.

4 Constraints

1. On the page load, input focus must come to the first name input field.
2. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
3. On click of “Submit” button, appointment details must be saved via fake-rest API in health-club.json.
4. Fake rest api is implemented with json-server.

Example JSON for reference of fields to be used for placing appointment:


```
{
    "firstname": "test",
    "lastname": "test",
    "age": 24,
    "phonenumber": 9988776655,
    "email": "test@test.com",
    "streetaddress": "test",
    "city": "test",
    "state": "test",
    "country": "india",
    "pincode": 560058,
    "trainerpreference": "Male Trainer",
    "physiotherapist": "Yes",
    "packages": "500",
    "inr": 1000,
    "paisa": 10,
    "id": 1
  }
```

5 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. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in next login.**
5. **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.**
6. **To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.**
7. **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
8. **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 <Ctrl> key while clicking on localhost:4200 to open project in browser -> takes 2 to 3 min**
   c. **npm run json-server -> to deploy fake rest api created with json-server -> takes 10 to 15 seconds**

d. **npm run jest -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min**

9. **You may also run "npm run jest" while developing the solution to re-factor the code to pass the test-cases.**
10. **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.**
11. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**