System Requirements Specification Index

For

Political Party System

Version 1.0

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

TABLE OF CONTENTS

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 users to add political parties, update political parties, delete political parties, get single political parties by id, get all political parties and search political parties.

1.2 Following is the requirement specification:

	Political Parties Application
Modules	
1	Political Party
Political Party Module Functionalities	
1	Create a Political Party
2	Update the existing Political Party
3	Get a Political Party by Id
4	Get a Political Party by Political Party Name
5	Get a Political Party by Founder Name
6	Fetch all registered Political Parties
7	Delete an existing Political Party

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 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.
- Is Deleted bool

4. Considerations

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

PoliticalParty

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

URL Exposed		Purpose
/parties		
Http Method	POST	Register a political party

Parameter 1	RegisterPoliticalParty ViewModel model	
Return	HTTP Response StatusCode	
/parties/{id}		Update a political party
Http Method	PUT	
Parameter 1	Long Id	
Parameter 2	RegisterPoliticalParty ViewModel model	
Return	HTTP Response StatusCode	
/parties		Fetches the list of all registered Political Parties
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Political Party>>	
/parties/{id}		Fetches the details of a political party
Http Method	GET	
Parameter 1	Long (id)	
Return	<PoliticalParty>	
/parties /{id}		Delete a political party
Http Method	DELETE	
Parameter 1	Long (id)	
Return	HTTP Response StatusCode	
/parties /search		Fetch political party by founder name
Http Method	GET	
Parameter 1	String founderName	
Return	<PoliticalParty>	
/parties /searchParty		Fetch political party by party name
Http Method	GET	
Parameter 1	String name	
Return	<PoliticalParty>	

6. Template Code Structure

6.1 Package: PoliticalParties

Resources

Names	Resource	Remarks	Status
Package Structure			
controller	PoliticalPartyController	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	Inside all these interface files contains all business validation logic functions.	Already implemented

Service	PoliticalParty 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 (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	RegisterPoliticalPartyView Model	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 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

1 PROBLEM STATEMENT

Political Party system is SPA (Single Page Application), it allows users to add political parties, update political parties, delete political parties, get single political parties by id, get all political parties and search political parties.

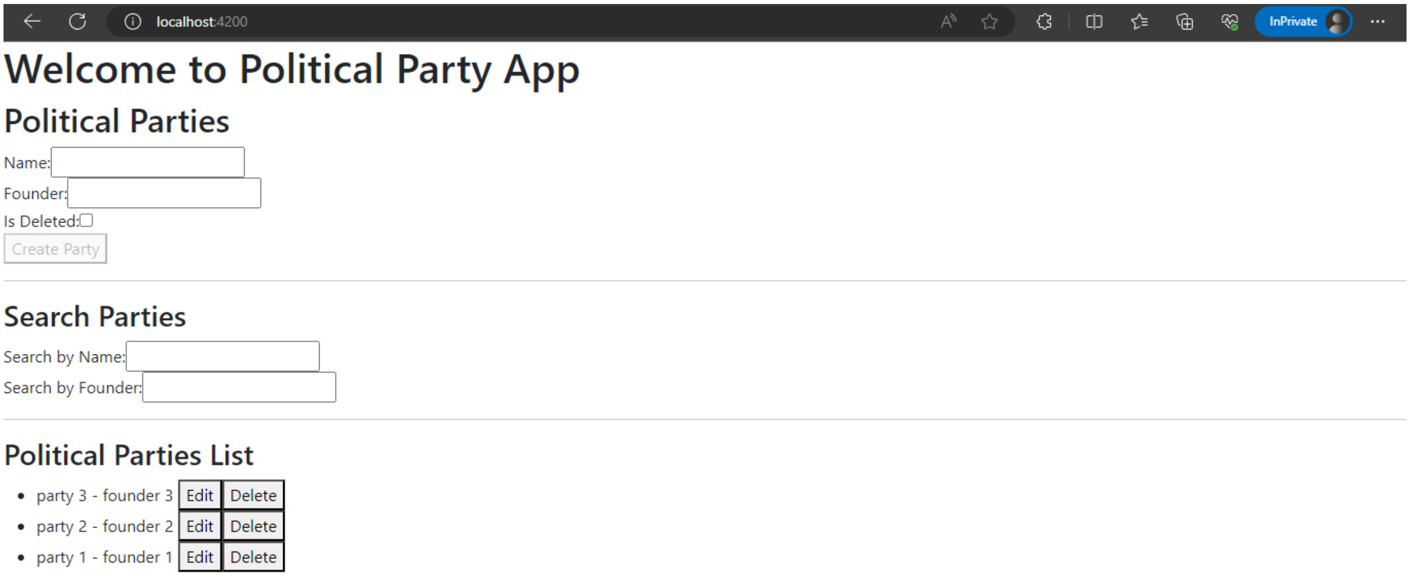
The core modules of Political party app are:

1. Home Page

2 PROPOSED POLITICAL PARTY SYSTEM WIREFRAME

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

2.1 HOME PAGE



Welcome to Political Party App

Political Parties

Name:

Founder:

Is Deleted:□

Create Party

Search Parties

Search by Name:

Search by Founder:

Political Parties List

- party 3 - founder 3 Edit Delete
- party 2 - founder 2 Edit Delete
- party 1 - founder 1 Edit Delete

3 BUSINESS-REQUIREMENT:

As an application developer, develop the Political Party System (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Home Page	As a user I should be able to visit the Home page as the default page.
US_01	Home Page	As a user I should be able to see the homepage and perform all operations: Acceptance criteria: 1. As a user I should be able to furnish the following details at the time of creating a political party. 1.1 Name 1.2 Founder 1.3 Deleted 2. Create party button should be disabled by default, and should be enabled when form is valid. 3. Name field min length is 3 and max length 100. 4. Founder field min length is 3 and max length 100. 5. Deleted should be a checkbox. 6. Name & Founder fields are mandatory. If any constraint is not satisfied, a validation message must be shown.

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.

7. Execution Steps to Follow For Backend

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, first cd into the "Backend" folder and then cd into your project folder (cd PoliticalParties).**

4. **To connect SQL server from terminal:**
   (PoliticalParties /sqlcmd -S localhost -U sa -P pass@word1)
   - **To create database from terminal -**

   ```
   1>  Create Database PoliticalPartyDb
   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 PoliticalPartyDb
   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.

8. Execution Steps to Follow For Frontend

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 is a web-based application, to run the application on a browser, use the internal browser in the environment.**
4. **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 test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min**
5. **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.**