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 InMemory database, where it allows to manage 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>api/PoliticalParty/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>api/PoliticalParty/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>api/PoliticalParty/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>api/PoliticalParty/parties/{politicalPartyId}
<table>
<tr><td>Http Method</td><td>GET</td></tr>
</table>
</td><td>Fetches the details of a political party</td></tr>
</table>

Parameter 1	Long (politicalPartyId)
Return	<PoliticalParty>

api/PoliticalParty/parties /{politicalPartyId}	Delete a political party
Http Method: DELETE Parameter 1: Long (politicalPartyId) Return: HTTP Response StatusCode	

5.2 PoliticalLeaderController

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

api/PoliticalLeader/leaders/{politicalPartyId} Http Method: DELETE Parameter 1: Long (politicalPartyId) Return: HTTP Response StatusCode	Delete a political leader from the existing leaders

5.3 DevelopmentController

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

<table>
<tr>
<td>api/Development/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
Package Structure		
controller	PoliticalPartyController PoliticalLeaderController DevelopmentController	These controllers handle all application Function, Create/Update/Edit show information
Startup.cs	Startup CS file	Contain all Services settings and InMemory Db Configuration.
Properties	launchSettings.json file	All URL Setting for API

6.2 Package: PoliticalParties.BusinessLayer

Resources

Names	Resource	Remarks
Package Structure		
Interface	IPoliticalPartyServices interface IPoliticalLeaderServices interface IDevelopmentServices interface	Inside all these interface files contains all business validation logic functions..
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.
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 InMemory database.
ViewModels	RegisterPoliticalLeaderViewModel RegisterPoliticalPartyViewModel RegisterDevelopmentViewModel	Contain all view Domain entities for show and bind data.

6.3 Package: PoliticalParties.DataLayer

Resources

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

6.4 Package: Online-Auction.Entities

Resources

Names	Resource	Remarks
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

7. Execution Steps to Follow

- **All actions like build, compile, running application, running test cases will be through Command Terminal.**

- **To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) Terminal → New Terminal.**

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

- **To build your project use command:**
  (Project_directory/PoliticalParties / dotnet build)

- **To launch your application, Run the following command to run the application:**
  (Project_directory/PoliticalParties / dotnet run)

- **This editor Auto Saves the code.**

- **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.**

- **To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**

- **To run the test cases in CMD, Run the following command to test the application:**
  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)

- **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.**

- **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.**

- **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.**
