System Requirements Specification Index

For

Political Party

Application

Version 1.0

IIHT Pvt. Ltd.

fullstack@iiht.com

**Table of Contents**

[1 Project Abstract 3](#_Toc77850439)

[2 Assumptions, Dependencies, Risks / Constraints 4](#_Toc77850440)

[2.1 Political Party Constraints: 4](#_Toc77850441)

[2.2 Political Leader Constraints 4](#_Toc77850442)

[2.3 Developments Constraints 4](#_Toc77850443)

[3 Business Validations 5](#_Toc77850445)

[4 Rest Endpoints 6](#_Toc77850446)

[4.1 PoliticalPartyController 6](#_Toc77850447)

[4.2 PoliticalLeaderController 6](#_Toc77850448)

[4.3 DevelopmentController 7](#_Toc77850449)

[5 Template Code Structure 8](#_Toc77850451)

[5.1 Package: com.iiht.training.ratings 8](#_Toc77850452)

[5.2 Package: com.iiht.training.ratings.entity 8](#_Toc77850453)

[5.3 Package: com.iiht.training.ratings.dto 9](#_Toc77850454)

[5.4 Package: com.iiht.training.ratings.filter 9](#_Toc77850455)

[5.5 Package: com.iiht.training.ratings.model.exception 10](#_Toc77850456)

[5.6 Package: com.iiht.training.ratings.repository 10](#_Toc77850457)

[5.7 Package: com.iiht.training.ratings.service 11](#_Toc77850458)

[5.8 Package: com.iiht.training.ratings.service.impl 12](#_Toc77850459)

[5.9 Package: com.iiht.training.ratings.exception 14](#_Toc77850460)

[6 Considerations 14](#_Toc77850461)

[7 Execution Steps to Follow 15](#_Toc77850462)

**Political Parties APPLICATION**

**System Requirements Specification**

# Project Abstract

**Political Parties** Application is Spring boot RESTful application with MySQL, where it allows to manage political parties, political leaders and developments done by political leaders in the states.

**Following is the requirement specifications**:

|  |  |
| --- | --- |
|  | 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 |
|  |  |

# Assumptions, Dependencies, Risks / Constraints

## Political Party Constraints:

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

## Political Leader Constraints

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

## Developments Constraints

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

**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 dto 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**

# Business Validations

* Political Party name is not null, min 3 and max 100 characters.
* Political party founder name is not null, min 3 and max 100 characters.
* Political Leader candidate name is not null, min 3 and max 100 characters.
* Political Leader state name is not null, min 3 and max 100 characters.
* Development title is not null, min 3 and max 100 characters.
* Development activity is not null, min 3 and max 100 characters.
* Development budget is not null, min 3 and max 100 characters.
* Development state is not null, min 3 and max 100 characters.
* Development activity month is not null, the range is from 1 to 12
* Development activity year is not null, the range is from 2021 to 2040.

# Rest Endpoints

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

## PoliticalPartyController

|  |  |
| --- | --- |
| **URL Exposed** | **Purpose** |
| 1. /parties  |  |  | | --- | --- | | Http Method | POST | | Parameter 1 | PoliticalPartyDto | | Return | PoliticalPartyDto | | Register a political party |
| /parties   |  |  | | --- | --- | | Http Method | PUT | | Parameter 1 | PoliticalPartyDto | | Return | PoliticalPartyDto | | Update a political party |
| /parties   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | - | | Return | List<PoliticalPartyDto> | | Fetches the list of all registered Political Parties |
| /parties/{politicalPartyId}   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | Long (politicalPartyId) | | Return | PoliticalPartyDto | | Fetches the details of a political party |
| /parties /{politicalPartyId}   |  |  | | --- | --- | | Http Method | DELETE | | Parameter 1 | Long (politicalPartyId) | | Return | PoliticalPartyDto | | Delete a political party |

## PoliticalLeaderController

|  |  |
| --- | --- |
| **URL Exposed** | **Purpose** |
| /leaders   |  |  | | --- | --- | | Http Method | POST | | Parameter 1 | PoliticalLeaderDto | | Return | PoliticalLeaderDto | | Register a political leader |
| /leaders   |  |  | | --- | --- | | Http Method | PUT | | Parameter 1 | PoliticalLeaderDto | | Return | PoliticalLeaderDto | | Update a political leader |
| /leaders   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | - | | Return | List< PoliticalLeaderDto> | | Fetches all registered political leaders |
| /leaders/{politicalLeaderId}   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | Long (politicalLeaderId) | | Return | List< PoliticalLeaderDto> | | Fetch the details of a political leader |
| /leaders/by-party-id/{politicalPartyId}   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | Long (politicalPartyId) | | Return | List< PoliticalLeaderDto> | | Fetches the details of all the political leaders belongs to a party |
| /leaders/{politicalPartyId}   |  |  | | --- | --- | | Http Method | DELETE | | Parameter 1 | Long (politicalPartyId) | | Return | Boolean | | Delete a political leader from the existing leaders |

## DevelopmentController

|  |  |
| --- | --- |
| **URL Exposed** | **Purpose** |
| /developments   |  |  | | --- | --- | | Http Method | POST | | Parameter 1 | DevelopmentDto | | Return | DevelopmentDto | | Register a development plan |
| /developments   |  |  | | --- | --- | | Http Method | PUT | | Parameter 1 | DevelopmentDto | | Return | DevelopmentDto | | Update an existing development plan |
| /developments   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | - | | Return | List<DevelopmentDto> | | Fetches all the registered developments |
| /developments/{developmentId}   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | Long(developmentId) | | Return | List<DevelopmentDto> | | Fetch the details of a development plan |
| /developments/by-leader-id/{politicalLeaderId}   |  |  | | --- | --- | | Http Method | GET | | Parameter 1 | Long(politicalLeaderId) | | Return | List<DevelopmentDto> | | Fetches all the development plans created for a political leader |
| /developments/{developmentId}   |  |  | | --- | --- | | Http Method | DELETE | | Parameter 1 | Long(developmentId) | | Return | Boolean | | Deletes an existing development plan |

# Template Code Structure

## Package: com.iiht.training.ratings

**Resources**

|  |  |  |
| --- | --- | --- |
| **PoliticianRatingsApplication (Class)** | This is the Spring Boot starter class of the application. | Already Implemented |

## Package: com.iiht.training.ratings.entity

**Resources**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class/Interface** | | **Description** | | **Status** |
| **PoliticalParty (class)** | * + Annotate this class with proper annotation to declare it as an entity class with **politicalPartyId** as primary key.   + Map this class with **politicalparty** table.   + Generate the **politicalPartyId** using **IDENTITY** strategy | | Partially implemented. | |
| **PoliticalLeader(class)** | * + This is class is partially implemented.   + Annotate this class with proper annotation to declare it as an entity class with **politicalLeaderId** as primary key.   + Map this class with **politicalleader** table.   + Generate the **politicalLeaderId** using the **IDENTITY** strategy | | Partially implemented. | |
| **Development(class)** | * + This is class is partially implemented.   + Annotate this class with proper annotation to declare it as an entity class with **developmentId** as primary key.   + Map this class with **development** table.   + Generate the **developmentId** using the **IDENTITY** strategy | | Partially implemented. | |

## Package: com.iiht.training.ratings.dto

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **PoliticalPartyDto (class)** | Use appropriate annotations from the **Java Bean Validation API** for validating attribute of this class. (Refer **Business Validation** section for validation rules). | Partially implemented. |
| **PoliticalLeaderDto (class)** | Use appropriate annotations from the **Java Bean Validation API** for validating attribute of this class. (Refer **Business Validation** section for validation rules). | Partially implemented. |
| **DevelopmentDto (class)** | Use appropriate annotations from the **Java Bean Validation API** for validating attribute of this class. (Refer **Business Validation** section for validation rules). | Partially implemented. |

## Package: com.iiht.training.ratings.model.exception

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **ExceptionResponse (class)** | Object of this class is supposed to be returned in case of exception through exception handlers | Already implemented. |

## Package: com.iiht.training.ratings.repository

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **PoliticalPartyRepository (interface)** | 1. Repository interface exposing CRUD functionality for **PoliticalParty** Entity. 2. You can go ahead and add any custom methods as per requirements | Partially implemented |
| **PoliticalLeaderRepository (interface)** | 1. Repository interface exposing CRUD functionality for **PoliticalLeader** Entity. 2. You can go ahead and add any custom methods as per requirements | Partially implemented |
| **DevelopmentRepository (interface)** | 1. Repository interface exposing CRUD functionality for **Development** Entity. 2. You can go ahead and add any custom methods as per requirements | Partially implemented |

## Package: com.iiht.training.ratings.service

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **PoliticalPartyService (interface)** | Interface to expose method signatures for political party related functionality.  Do not modify, add or delete any method | Already implemented. |
| **PoliticalLeaderService (interface)** | Interface to expose method signatures for political leader related functionality.  Do not modify, add or delete any method | Already implemented. |
| **DevelopmentService (interface)** | Interface to expose method signatures for Developments related functionality.  Do not modify, add or delete any method | Already implemented. |

## Package: com.iiht.training.ratings.service.impl

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **PoliticalPartyServiceImpl (class)** | * Implements **PoliticalPart**y**Service**. Contains template method implementation. * Need to provide implementation for political party related functionalities * Add required repository dependency * Do not modify, add or delete any method signature | To be implemented. |
| **PoliticalLeaderServiceImpl (class)** | * Implements **PoliticalLeaderService**. Contains template method implementation. * Need to provide implementation for political leader related functionalities * Add required repository dependency * Do not modify, add or delete any method signature | To be implemented. |
| **DevelopmentServiceImpl (class)** | * Implements **DevelopmentService**. Contains template method implementation. * Need to provide implementation for developments related functionalities * Add required repository dependency * Do not modify, add or delete any method signature | To be implemented. |

## Package: com.iiht.training.ratings.exception

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **GlobalHandler (class)** | * RestControllerAdvice Class for defining global exception handlers. * Contains Exception Handler for **InvalidDataException** class. * Use this as a reference for creating exception handler for other custom exception classes | Partially implemented. |

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **PoliticalPartyNotFoundException (Class)** | * Custom Exception to be thrown when trying to fetch or delete the political party info which does not exists. * Need to create Exception Handler for same wherever needed (local or global) | Already created. |
| **PoliticalLeaderNotFoundException (Class)** | * Custom Exception to be thrown when trying fetch or delete political leader info which dose not exists. * Need to create Exception Handler for same wherever needed (local or global) | Already created. |
| **DevelopmentNotFoundException (Class)** | * Custom Exception to be thrown when trying to fetch or delete a development info which does not exists. * Need to create Exception Handler for same wherever needed (local or global) | Already created. |

## Package: com.iiht.training.ratings.controller

**Resources**

|  |  |  |
| --- | --- | --- |
| **Class/Interface** | **Description** | **Status** |
| **PoliticalPartyController (Class)** | * Controller class to expose all rest-endpoints for Political Party related activities. * May also contain local exception handler methods | To be implemented |
| **PoliticalLeaderController (Class)** | * Controller class to expose all rest-endpoints for Political Leader related activities. * May also contain local exception handler methods | To be implemented |
| **DevelopmentController (Class)** | * Controller class to expose all rest-endpoints for Developments related activities. * May also contain local exception handler methods | To be implemented |

# Considerations

1. There is no roles in this application
2. You can perform the following 3 possible actions

|  |
| --- |
| PoliticalParty |
| PoliticalLeader |
| Development |

# 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 Application menu (Three horizontal lines at left top) 🡪 Terminal 🡪New Terminal**
3. **To build your project and run test cases use command:**

**mvn clean package**

1. **To launch your application, move into target folder (cd target). Run the following command to run the application:**

**java -jar politicians-0.0.1-SNAPSHOT.jar**

1. **This editor Auto Saves the code**
2. **If the 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.**
3. **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.**
4. **Default credentials for MySQL:**
   1. **Username: root**
   2. **Password: pass@word1**
5. **To login to mysql instance: Open new terminal and use following command:**
   1. **sudo systemctl enable mysql**
   2. **sudo systemctl start mysql**
   3. **mysql -u root -p**

**The last command will ask for password which is ‘pass$word1’**

1. **Mandatory: Before final submission run the following command:**

**mvn test**