
System Requirements Specification Index

For

Political Party Application

Version 1.0

TABLE OF CONTENTS

1	Project Abstract	3
2	Assumptions, Dependencies, Risks / Constraints	4
2.1	Political Party Constraints:	4
2.2	Political Leader Constraints	4
2.3	Developments Constraints	4
3	Business Validations	5
4	Rest Endpoints	6
4.1	PoliticalPartyController	6
4.2	PoliticalLeaderController	6
4.3	DevelopmentController	7
5	Template Code Structure	8
5.1	Package: com.iiht.training.ratings	8
5.2	Package: com.iiht.training.ratings.entity	8
5.3	Package: com.iiht.training.ratings.dto	9
5.4	Package: com.iiht.training.ratings.filter	9
5.5	Package: com.iiht.training.ratings.model.exception	10
5.6	Package: com.iiht.training.ratings.repository	10
5.7	Package: com.iiht.training.ratings.service	11
5.8	Package: com.iiht.training.ratings.service.impl	12
5.9	Package: com.iiht.training.ratings.exception	14
6	Considerations	14
7	Execution Steps to Follow	15

Political Parties APPLICATION

System Requirements Specification

1 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

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.

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**

3 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.

4 REST ENDPOINTS

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

4.1 POLITICALPARTYCONTROLLER

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

4.2 POLITICALLEADERCONTROLLER

URL Exposed		Purpose
/leaders		Register a political leader
Http Method	POST	
Parameter 1	PoliticalLeaderDto	
Return	PoliticalLeaderDto	
/leaders		Update a political leader
Http Method	PUT	
Parameter 1	PoliticalLeaderDto	
Return	PoliticalLeaderDto	

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

4.3 DEVELOPMENTCONTROLLER

URL Exposed		Purpose
/developments		Register a development plan
Http Method	POST	
Parameter 1	DevelopmentDto	
Return	DevelopmentDto	
/developments		Update an existing development plan
Http Method	PUT	
Parameter 1	DevelopmentDto	
Return	DevelopmentDto	
/developments		Fetches all the registered developments
Http Method	GET	
Parameter 1	-	
Return	List<DevelopmentDto>	
/developments/{developmentId}		Fetch the details of a development plan
Http Method	GET	
Parameter 1	Long(developmentId)	
Return	List<DevelopmentDto>	

/developments/by-leader-id/{politicalLeaderId}		Fetches all the development plans created for a political leader
Http Method	GET	
Parameter 1	Long(politicalLeaderId)	
Return	List<DevelopmentDto>	
/developments/{developmentId}		Deletes an existing development plan
Http Method	DELETE	
Parameter 1	Long(developmentId)	
Return	Boolean	

5 TEMPLATE CODE STRUCTURE

5.1 PACKAGE: COM.IIHT.TRAINING.RATINGS

Resources

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

5.2 PACKAGE: COM.IIHT.TRAINING.RATINGS.ENTITY

Resources

Class/Interface	Description	Status
PoliticalParty (class)	<ul style="list-style-type: none"> o Annotate this class with proper annotation to declare it as an entity class with politicalPartyId as primary key. o Map this class with politicalparty table. o Generate the politicalPartyId using IDENTITY strategy 	Partially implemented.
PoliticalLeader(class)	<ul style="list-style-type: none"> o This class is partially implemented. o Annotate this class with proper annotation to declare it as an entity class with politicalLeaderId as primary key. o Map this class with politicalleader table. 	Partially implemented.

	<ul style="list-style-type: none"> o Generate the politicalLeaderId using the IDENTITY strategy 	
Development(class)	<ul style="list-style-type: none"> o This class is partially implemented. o Annotate this class with proper annotation to declare it as an entity class with developmentId as primary key. o Map this class with development table. o Generate the developmentId using the IDENTITY strategy 	Partially implemented.

5.3 PACKAGE: COM.IIHT.TRAINING.RATINGS.DTO

Resources

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

--	--	--

5.4 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.

5.5 PACKAGE: COM.IIHT.TRAINING.RATINGS.REPOSITORY

Resources

Class/Interface	Description	Status
PoliticalPartyRepository (interface)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 1. Repository interface exposing CRUD functionality for Development Entity. 2. You can go ahead and add any custom methods as per requirements 	Partially implemented

5.6 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.

5.7 PACKAGE: COM.IIHT.TRAINING.RATINGS.SERVICE.IMPL

Resources

Class/Interface	Description	Status
PoliticalPartyServiceImpl (class)	<ul style="list-style-type: none">Implements PoliticalPartyService. Contains template method implementation.	To be implemented.

	<ul style="list-style-type: none"> • Need to provide implementation for political party related functionalities • Add required repository dependency • Do not modify, add or delete any method signature 	
PoliticalLeaderServiceImpl (class)	<ul style="list-style-type: none"> • 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)	<ul style="list-style-type: none"> • 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.

5.8 PACKAGE: COM.IIHT.TRAINING.RATINGS.EXCEPTION

Resources

Class/Interface	Description	Status
GlobalHandler (class)	<ul style="list-style-type: none">● 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)	<ul style="list-style-type: none">● Custom Exception to be thrown when trying to fetch or delete the political party info which does not exist.● Need to create Exception Handler for same wherever needed (local or global)	Already created.
PoliticalLeaderNotFoundException (Class)	<ul style="list-style-type: none">● Custom Exception to be thrown when trying to fetch or delete political leader info which does not exist.● Need to create Exception Handler for same	Already created.

	wherever needed (local or global)	
DevelopmentNotFoundException (Class)	<ul style="list-style-type: none"> • Custom Exception to be thrown when trying to fetch or delete a development info which does not exist. • Need to create Exception Handler for same wherever needed (local or global) 	Already created.

5.9 PACKAGE: COM.IIHT.TRAINING.RATINGS.CONTROLLER

Resources

Class/Interface	Description	Status
PoliticalPartyController (Class)	<ul style="list-style-type: none"> • Controller class to expose all rest-endpoints for Political Party related activities. • May also contain local exception handler methods 	To be implemented
PoliticalLeaderController (Class)	<ul style="list-style-type: none"> • Controller class to expose all rest-endpoints for Political Leader related activities. • May also contain local exception handler methods 	To be implemented
DevelopmentController (Class)	<ul style="list-style-type: none"> • Controller class to expose all rest-endpoints for Developments related activities. • May also contain local exception handler methods 	To be implemented

6 CONSIDERATIONS

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

PoliticalParty
PoliticalLeader
Development

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 Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. To build your project use command:
mvn clean package -Dmaven.test.skip
4. To launch your application, move into the target folder (**cd target**). Run the following command to run the application:
java -jar politician-ratings-0.0.1-SNAPSHOT.jar
5. This editor Auto Saves the code.
6. 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.
7. 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.
8. 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.

9. 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

10. Default credentials for MySQL:

- a. Username: **root**
- b. Password: **pass@word1**

11. To login to mysql instance: Open new terminal and use following command:

- a. **sudo systemctl enable mysql**
- b. **sudo systemctl start mysql**

NOTE: After typing the second sql command (sudo systemctl start mysql), you may encounter a warning message like :

System has not been booted with systemd as init system (PID 1).
Can't operate. Failed to connect to bus: Host is down

**>> Please note that this warning is expected and can be disregarded.
Proceed to the next step.**

- c. **mysql -u root -p**

The last command will ask for password which is '**pass@word1**'

12. Mandatory: Before final submission run the following command:

mvn test

13. 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.