System Requirements Specification Index

For

Django Rest-API
Political PartyApplication

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

Political Parties APPLICATION
System Requirements Specification

1 PROJECT ABSTRACT

Political Parties Application is Django RESTful application withSQLite database, 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 (fetch byparty_id)
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 (fetch by leader_id)

2 Assumptions, Dependencies, Risks / Constraints

2.1 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.

2.2 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.

2.3 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.

3 Business Validations

- Political Party name is max 100 characters.
- Political party founder name is max 100 characters.
- Political Leader candidate name is max 100 characters.
- Political Leader state name is max 100 characters.
- Development title is max 100 characters.
- Development activity is max 100 characters.
- Development budget is max 100 characters.
- Development state is max 100 characters.
- Development activity month is range is from 1 to 12
- Development activity year range is from 2021 to 2040.

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

Class Name	Method Name	Purpose Of Method
PoliticalPartyView	get(self,request,pk=None,format=None)	Get a political party by Id and Fetch all registered political parties.
	post(self, request,format=None)	Registering thepolitical party.
	patch(self,request,pk,format=None)	Update thepolitical party.
	delete(self,request,pk,format=None)	Delete thepolitical party.
PoliticalLeaderView	get(self,request,pk=None,format=None)	Get a Political Leader by Id and Fetch all registered Political Leaders
	post(self, request,format=None)	Register a Political Leader
	patch(self,request,pk,format=None)	Update the existing Political Leader
	delete(self,request,pk,format=None)	Delete an existing Political Leader
PoliticalLeaderByPartyView	get(self,request,pk=None,format=None)	Fetch all Political Leaders registered with a Party
DevelopmentView	get(self,request,pk=None,format=None)	Get a Development by Id and Fetch all created developments
	post(self, request,format=None)	Create a Development Plan
	patch(self,request,pk,format=None)	Update the existing Development
	delete(self,request,pk,format=None)	Delete an existing Development
DevelopmentByLeaderView	get(self,request,pk=None,format=None)	Fetch all Developments created for a Political Leader

Resources (Models)

Class	Description	Status
PoliticalPartyModel	o A model class for PoliticalParty. o It will map to the PoliticalPartyModel table.	Already implemented.
PoliticalLeaderModel	o A model class for PoliticalLeader o It will map to the PoliticalLeaderModeltable.	Already implemented.
DevelopmentModel	o A model class for Development. o It will map to the DevelopmentModel table.	Already implemented.

Resources (Serializers)

Class	Description	Status
PoliticalPartySerializer	A serializer for PoliticalPartyModel	Already implemented
PoliticalLeaderSerializer	A serializer for PoliticalLeaderModel	Already implemented
DevelopmentSerializer	A serializer for DevelopmentModel	Already implemented

Resources (Views)

Class	Description	Status
PoliticalPartyView	A class for the get, post, patch, delete functionalities on **PoliticalPartyModel**model**.**	To be implemented

PoliticalLeaderView	A class for the get, post, patch, delete functionalities on **PoliticalLeaderModel**model**.**	To be implemented
PoliticalLeaderByPartyView	A class for the get functionality on **PoliticalLeaderModel**model**.**	To be implemented
DevelopmentView	A class for the get, post, patch, delete functionalities on **DevelopmentModel**model**.**	To be implemented
DevelopmentByLeaderView	A class for the get functionality by leader on **DevelopmentModel**model**.**	To be implemented

Resources (Exceptions)

Class	Description	Status
IPoliticalPartyIdNotAvailable	Object of this exception class is supposed to be returned in case specified political party id is not available.	Already implemented.
PoliticalLeaderIdNotAvailable	Object of this exception class is supposed to be returned in case specified political leader id is not available.	Already implemented.
DevelopmentIdNotAvailable	Object of this exception class is supposed to be returned in case	Already implemented.

	specified development id is not available.	
InvalidData	Object of this exception class is supposed to be returned in case received data is invalid.	Already implemented.
PoliticalLeaderIdNotAvailable	Object of this exception class is supposed to be returned in case specified political leader id is not available.	Already implemented.
PoliticalPartyIdNotAvailable	Object of this exception class is supposed to be returned in case specified political party id is not available.	Already implemented.

6 CONSIDERATIONS

A. There are 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. The editor Auto Saves the code.
4. If you want to exit(logout) and to 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.
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. 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.
8. Install 'djangorestframework' module before running the code. For this use the following command.

   ```
   pip install djangorestframework
   ```

9. Use the following command to run the server

   ```
   python3 manage.py runserver
   ```

10. Mandatory: Before final submission run the following commands to execute testcases

    ```
    python3 manage.py test partyapp.test.test_functional

    python3 manage.py test partyapp.test.test_exceptional

    python3 manage.py test partyapp.test.test_boundary
    ```

11. To test rest end points

    Click on 'Thunder Client' or use Ctrl+Shift+R ->Click on 'New Request' (at left side of IDE)

12. 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.

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.

------- * --------