

Sr. No.	Test Scenario	Test Case	Test Steps	Test Data	Expected Behaviour	Actual Behaviour	Result
1	Entercandidate endpoint	Add New Candidate Verify entercandidate api saves a candidate's name with vote count initialized to 0	- Type /entercandidate?name={name} in URL, here name is the name of candidate - Click enter	name=Rajiv Eg. http://localhost:8080/entercandidate?name=Rajiv	Valid Candidate should be created with vote count initialized to 0	Same as Expected	PASS
2	Entercandidate endpoint	Don't Add Existing Candidate Verify entercandidate api doesn't save an already existing candidate's name. An appropriate reponse should be given to user	- Type /entercandidate?name={name} in URL, here name is the name of existing candidate - Click enter	name=Pranav Eg. http://localhost:8080/entercandidate?name=Pranav	Existing candidate should not be updated. Appropriate validation response should be shown	Same as Expected	PASS
3	Entercandidate endpoint	Don't Add Invalid Candidate Verify entercandidate doesn't save an empty or invalid parameter as a candidate and returns an appropriate response i.e Candidate name Validation	- Type /entercandidate?name={} in URL, here name is empty or invalid characters	name="" / name=122344, name= Eg. http://localhost:8080/entercandidate?name=	Invalid name should not be saved. Appropriate validation response should be shown	Same as Expected	PASS
4	Entercandidate endpoint	Same Data across Users / Sessions Verify candidate entered by one user is shown in another user's session	- Open multiple browser sessions with different users - Enter a candidate in one window and see if the data is also reflected in others	name=Mohan Eg. http://localhost:8080/entercandidate?name=Mohan	Candidate saved in one user session should be visible to another user	Same as Expected	PASS
5	Castvote endpoint	Increase vote count by 1 Verify castvote api increases the vote count for existing candidate by 1 and returns the count value	- Type /castvote?name={} in URL, here name is a valid existing candidate - Click enter	name=Pranav Eg. http://localhost:8080/castvote?name=Pranav	Vote count of the entered candidate should increase by one. The latest vote count should be shown	Same as Expected	PASS
6	Castvote endpoint	Validation Response Verify castvote api returns proper validation response when given candidate name is: - Empty / Invalid - Doesn't Exist	Empty/Invalid - Type /castvote?name={} in URL, here name is empty or invalid - Click enter Doesn't Exist - Type /castvote?name={} in URL, here name is valid but doesn't exist - Click enter	Empty/Invalid name="" name=122344, name= Eg. http://localhost:8080/castvote?name= Doesn't Exist name=Babar Eg. http://localhost:8080/castvote?name=Babar	Empty/Invalid Response with message "Please enter a valid name" should be shown Doesn't Exist Response with message "Candidate does not exist" should be shown	Same as Expected	PASS
7	Countvote endpoint	Latest Count Verify countvote returns the latest vote count for a given candidate	- Type /countvote?name={name} in URL, here name is the name of candidate - Click enter	name=Rajiv Eg. http://localhost:8080/countvote?name=Rajiv	Latest vote count for existing candidate must be shown as response	Same as Expected	PASS

Sr. No.	Test Scenario	Test Case	Test Steps	Test Data	Expected Behaviour	Actual Behaviour	Result
8	Countvote endpoint	Validation Response Verify countvote api returns proper validation response when given candidate name is: <ul style="list-style-type: none"> - Empty / Invalid - Doesn't Exist 	<ul style="list-style-type: none"> - Type /countvote?name=(name) in URL, here name is empty or invalid - Click enter 	name="" name=234354364kjoI Eg. http://localhost:8080/countvote?name=	Empty/Invalid Response with message "Please enter a valid name" should be shown Doesn't Exist Response with message "Candidate does not exist" should be shown	Same as Expected	PASS
9	Listvote endpoint	Candidate Names and Votes Verify listvote returns all candidate names and their latest vote counts as a JSON reponse	<ul style="list-style-type: none"> - Type /listvote in URL - Click enter 	Eg. http://localhost:8080/listvote	All existing candidates along with their vote count should be returned as response	Same as Expected	PASS
10	Listvote endpoint	Only Valid Candidates Verify listvote doesn't return invalid candidate names as a JSON reponse	<ul style="list-style-type: none"> - Type /listvote in URL - Click enter 	Eg. http://localhost:8080/listvote	The response should not contain invalid/empty names	Same as Expected	PASS
11	Listvote endpoint	Same Data Across Users / Sessions Verify listvote returns the latest and correct data for multiple users and sessions at the same time	<ul style="list-style-type: none"> - Open multiple browser windows (different users, incognito mode) - Type /listvote in URL - Click enter 	Eg. http://localhost:8080/listvote	The response of all existing candidates along with their vote count should be same across multiple user sessions.	Same as Expected	PASS
12	Getwinner endpoint	Single Winner Verify getwinner returns name of candidate with largest number of votes	<ul style="list-style-type: none"> - Type /getwinner in URL - Click enter 	Eg. http://localhost:8080/getwinner	Candidate with maximum votes should be returned as response	Same as expected	PASS
13	Getwinner endpoint	Multiple Candidates (Tie) Verify getwinner returns multiple candidates in case of tie	<ul style="list-style-type: none"> - Type /getwinner in URL - Click enter 	Eg. http://localhost:8080/getwinner	Candidates with maximum votes should be returned as response	Same as expected	PASS
14	Getwinner endpoint	Same Data Across Users / Sessions Verify getwinner returns the same valid response when accessed by multiple users / sessions	<ul style="list-style-type: none"> - Open multiple browser windows (different users, incognito mode) - Type /getwinner in URL - Click enter 	Eg. http://localhost:8080/getwinner	Winner responses should be same across multiple Users, Browser sessions.	Same as expected	PASS