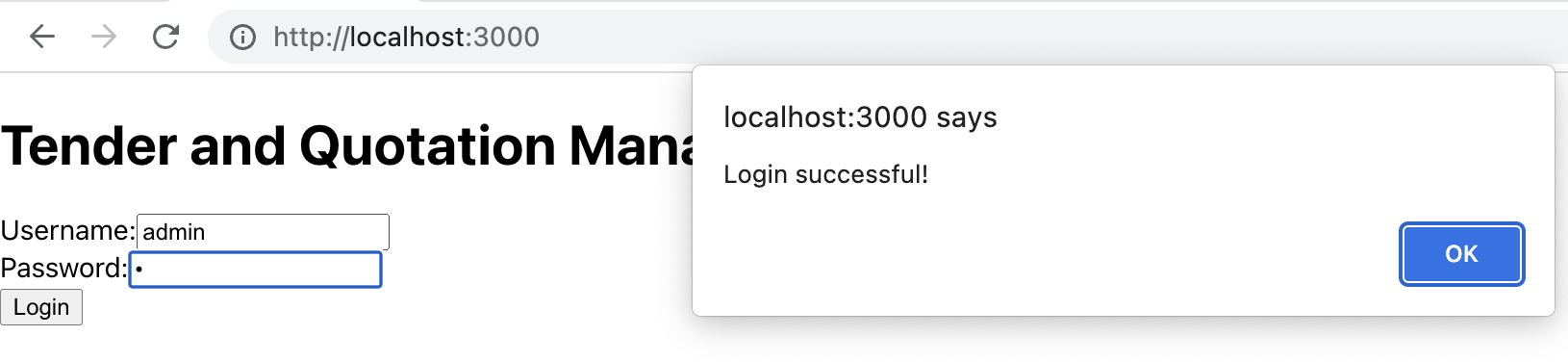
**Tender and Quotation Management System - Test Cases**

TEST CASES FROM UI:

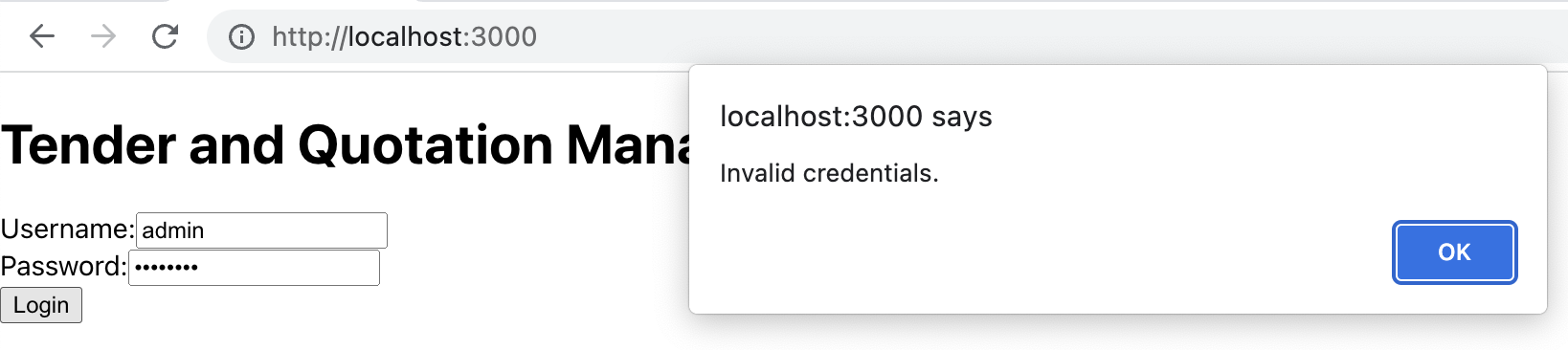
User Management:

Login:

1. Positive Test Case: Provide valid login credentials and ensure that the user is able to login successfully.



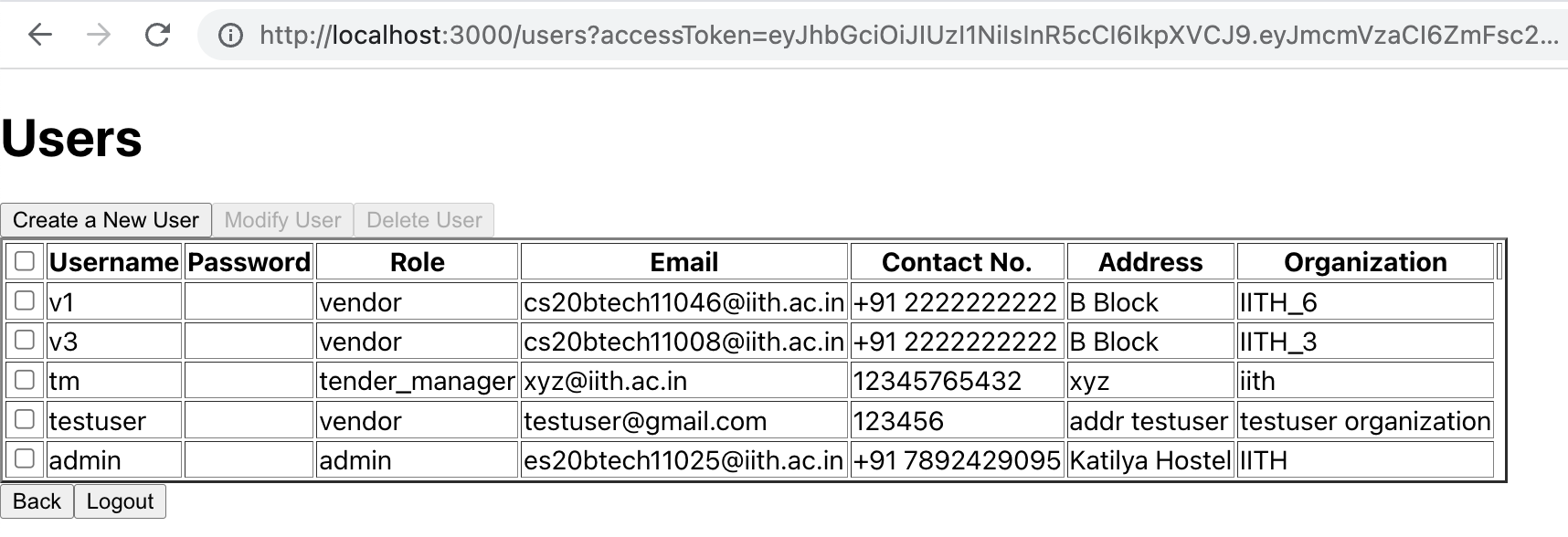
1. Negative Test Case: Provide invalid login credentials and ensure that the user is unable to login and receives an appropriate error message.



View Users:

Positive Test Case:

1. Ensure that the admin user is able to view all users registered in the system.

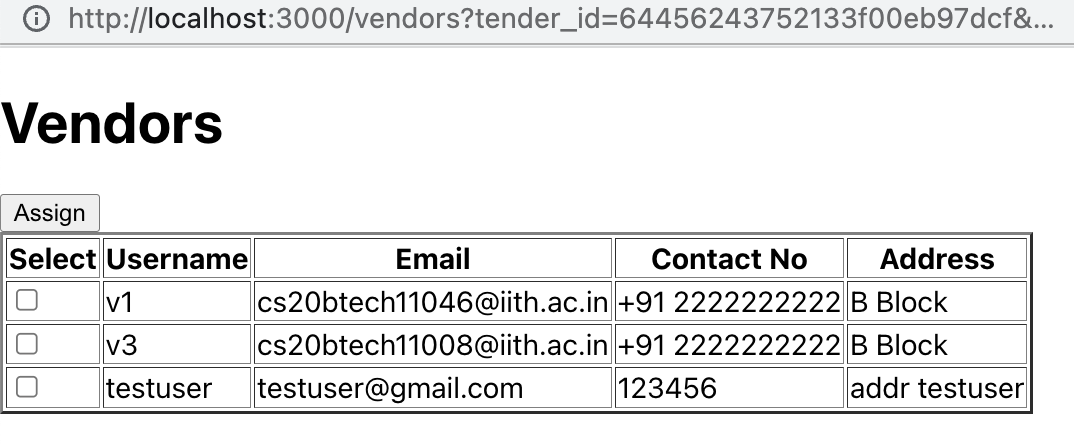


Negative Test Case:

1. Ensure that non-admin users are unable to view user information without appropriate permissions.

There is no screen to display a list of users for non-admin users, other than showing the list of

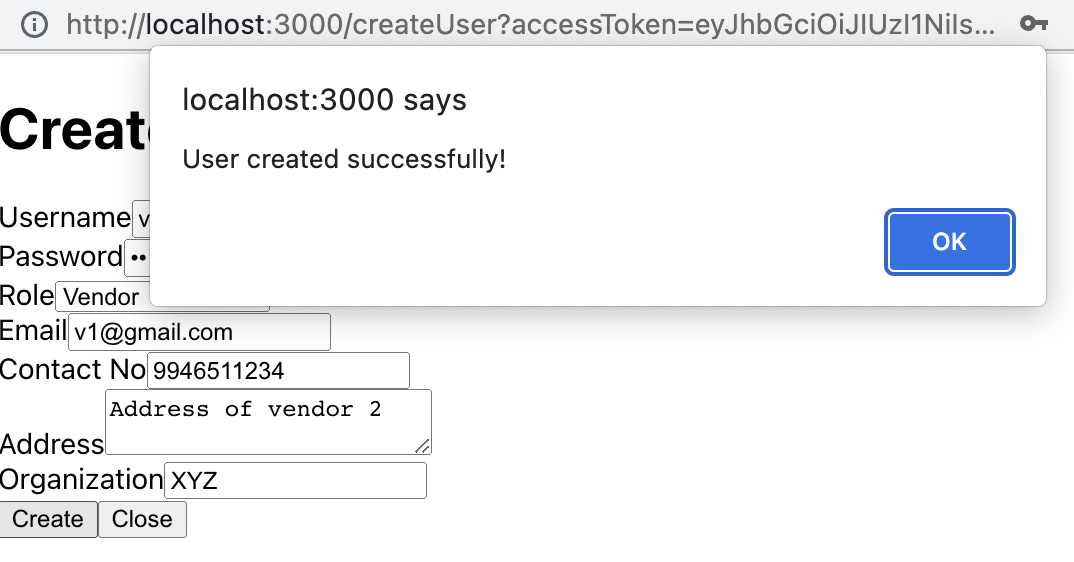
vendors to assign/notify the tender for a user with tender\_management role.



Create User:

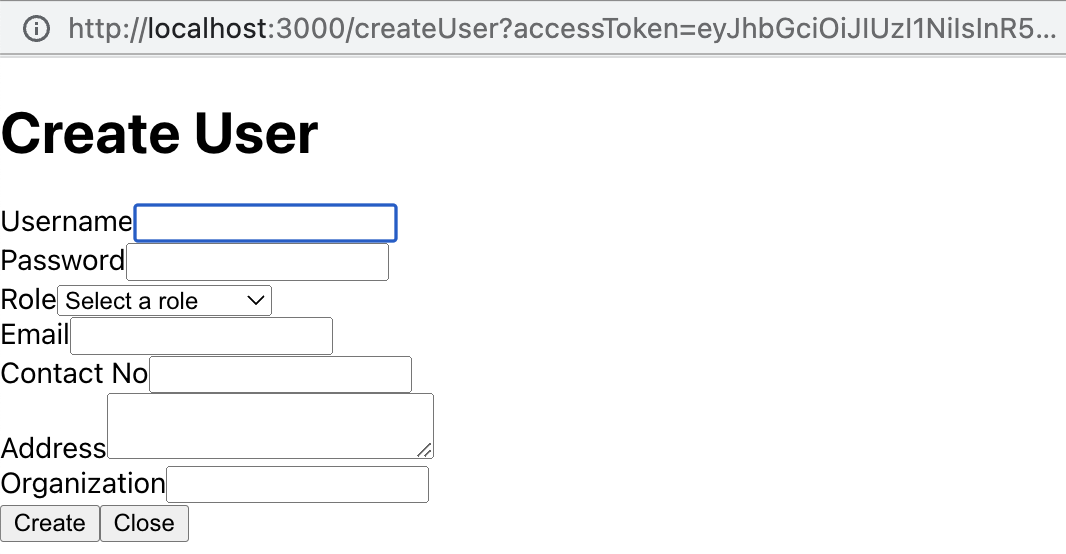
Positive Test Case:

1. Ensure that the admin user is able to create a new user with valid information and that the user is added to the database.



Negative Test Case:

1. Attempt to create a user with invalid information (such as incomplete fields) and ensure that the admin user receives an appropriate error message.



View Tenders:

Positive Test Case:

1. Ensure that the user is able to view all tenders created by them.

Negative Test Case:

1. Ensure that the user is unable to view tenders created by other users.

View Vendor Tenders:

Positive Test Case:

1. Ensure that the user is able to view all tenders that have been assigned to them.

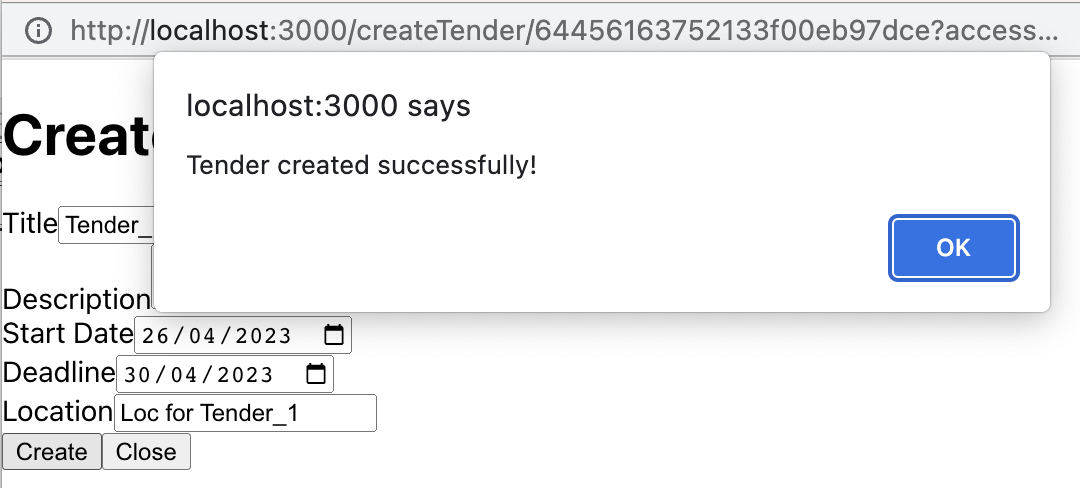
Negative Test Case:

1. Ensure that the user is unable to view tenders assigned to other vendors.

Create Tender:

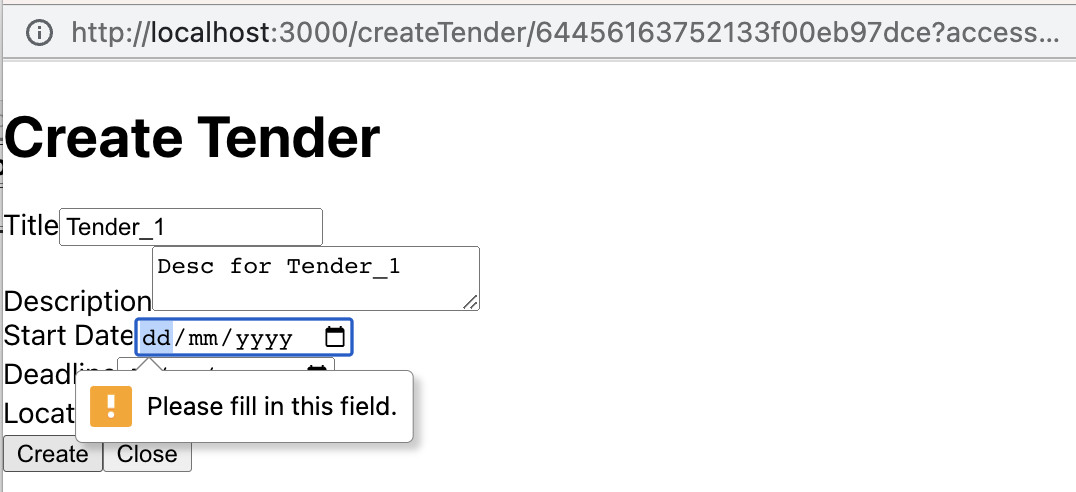
Positive Test Case:

1. Ensure that the user is able to create a tender with valid information and that the tender is added to the database.



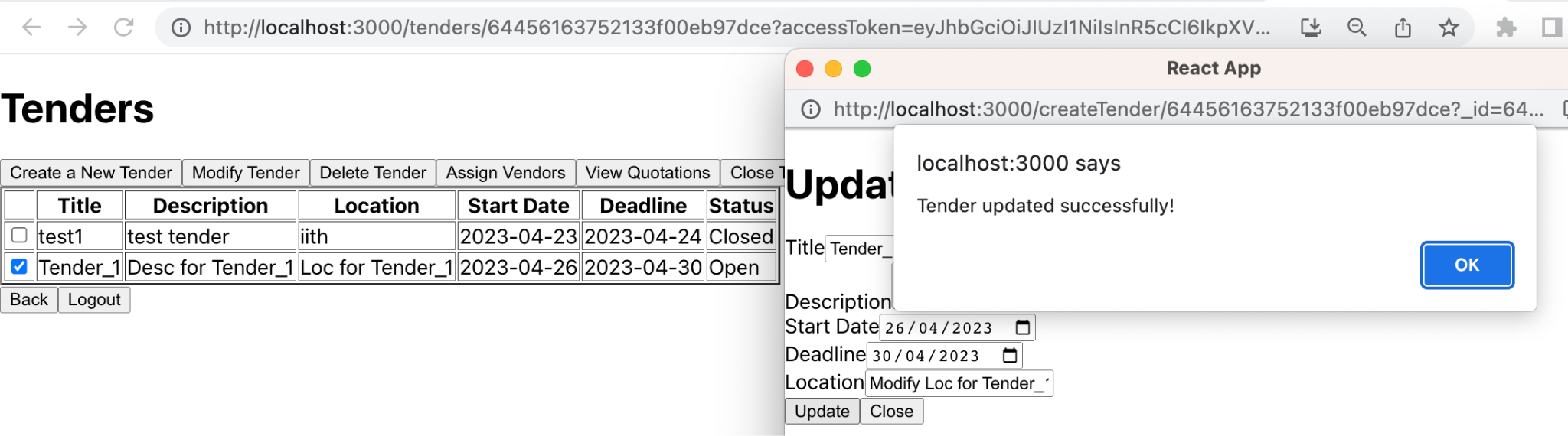
Negative Test Case:

1. Attempt to create a tender with invalid information (such as incomplete fields) and ensure that the user receives an appropriate error message.

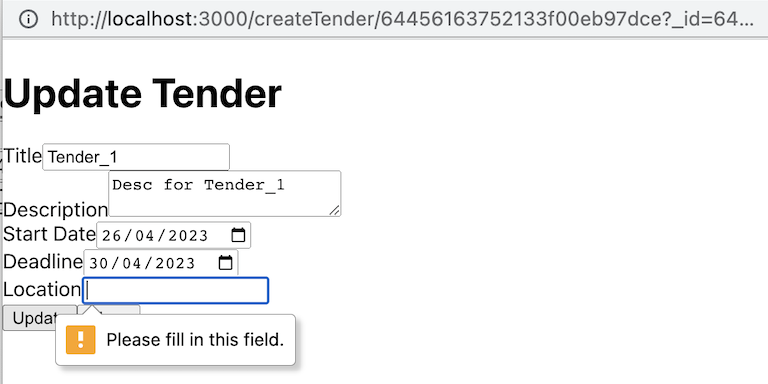


Modify a Tender:

1. Ensure that the user is able to update a tender with valid information and that the tender is added to the database.



1. Attempt to modify a tender with invalid information (such as incomplete fields) and ensure that the user receives an appropriate error message.



Assign a Tender to set of Vendors:

Delete a Tender:

View Quotations:

Positive Test Case:

1. Ensure that the user is able to view all quotations that have been received for a particular tender.

Negative Test Case:

1. Ensure that the user is unable to view quotations for tenders that have not been assigned to them.

Create Quotation:

Positive Test Case:

1. Ensure that the vendor is able to create a quotation for a tender with valid information and that the quotation is added to the database.

Negative Test Case:

1. Attempt to create a quotation with invalid information (such as incomplete fields) and ensure that the vendor receives an appropriate error message.

Modify Quotation:

Delete Quotation:

View Vendors:

Positive Test Case:

1. Ensure that the user is able to view all vendors registered in the system.

Negative Test Case:

1. Ensure that the user is unable to view vendor information without appropriate permissions.

TEST CASES FROM API:

Positive test cases:

User Management:

Test case for logging in:

25. Test with valid login credentials of an existing user and check if the login is successful and the access token is generated.

curl -X POST http://127.0.0.1:5000/login -H 'Content-Type: application/json' -d '{

"username": "admin",

"password": "a"

}'

{"access\_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjE0MjE2NCwianRpIjoiYzgzZjI3YTgtYWIyMi00NzNkLWE0ZTUtZTAwNDNlOTJhNzJiIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMTQyMTY0LCJleHAiOjE2ODIxNDMwNjQsInJvbGUiOiJhZG1pbiJ9.EHrKNgQ7AyPM7ycUbe3osUZyfTapmU-1bWrdP8U\_z28","role":"admin","success":true,"userid":"64407d244f79048c309cd7c7"}

Test case for registering a new user:

26. Test with valid user details and check if the user is registered successfully.

curl -X POST -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saXJeGbyUHQ' -H 'Content-Type: application/json' -d '{"username":"testuser","password":"testuser","role":"vendor","email":"testuser@gmail.com","contactNo":"123456","address":"addr testuser","organization":"testuser organization"}' http://localhost:5000/users

{"success":true}

27. Test with username already existing in the database and check if the registration fails and returns the message "User already exists".

curl -X POST -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saXJeGbyUHQ' -H 'Content-Type: application/json' -d '{"username":"testuser","password":"testuser","role":"vendor","email":"testuser@gmail.com","contactNo":"123456","address":"addr testuser","organization":"testuser organization"}' http://localhost:5000/users

{"message":"User already exists.","success":false}

Test case for getting all users:

27. Test with valid admin credentials and check if all users are returned successfully.

curl -X GET -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saXJeGbyUHQ' http://localhost:5000/users

{"success":true,"users":[{"\_id":"64407f534f79048c309cd7cc","address":"B Block","contactNo":"+91 2222222222","email":"cs20btech11046@iith.ac.in","organization":"IITH\_6","role":"vendor","username":"v1"},{"\_id":"64407f654f79048c309cd7ce","address":"B Block","contactNo":"+91 2222222222","email":"cs20btech11008@iith.ac.in","organization":"IITH\_3","role":"vendor","username":"v3"},{"\_id":"6443d217cc5fd2e0d20078e5","address":"Katilya Hostel","contactNo":"+91 7892429095","email":"es20btech11025@iith.ac.in","organization":"IITH","role":"admin","username":"admin"},{"\_id":"64456163752133f00eb97dce","address":"xyz","contactNo":"12345765432","email":"xyz@iith.ac.in","organization":"iith","role":"tender\_manager","username":"tm"},{"\_id":"64461e797931b3049f3ae3cc","address":"addr testuser","contactNo":"123456","email":"testuser@gmail.com","organization":"testuser organization","role":"vendor","username":"testuser"}]}

Test case for getting a user by user\_id:

29. Test with valid admin credentials and a valid user\_id and check if the user is returned successfully.

curl -X GET -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saXJeGbyUHQ' http://localhost:5000/users/64461e797931b3049f3ae3cc

{"success":true,"user":{"\_id":"64461e797931b3049f3ae3cc","address":"addr testuser","contactNo":"123456","email":"testuser@gmail.com","organization":"testuser organization","password":"sha256$SJwCxLh2RmWG5i05$1cb773144057ffebb0f8fcca6bbcf097fec4fcd451cb9cf61b771d91da11b9a6","role":"vendor","username":"testuser"}}

30. Test with invalid admin credentials and a valid user\_id and check if the user retrieval fails and returns the message "Not authorized to access user."

curl -X GET -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saYJeGbyUHQ' http://localhost:5000/users/64461e797931b3049f3ae3cc

{"msg":"Signature verification failed"}

Test case for updating an existing user:

31. Test with valid admin credentials and a valid user\_id and check if the user details are updated successfully.

curl -X PUT -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNzU0NCwianRpIjoiN2U0YTMxZTYtYzAzYi00NTg5LTg3ODAtMTMzMzBjZjFhOGZhIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE3NTQ0LCJleHAiOjE2ODIzMTg0NDQsInJvbGUiOiJhZG1pbiJ9.EprN6AJFCh-osb2pVtn3DbZQ6KFu3yh0GAS7Jg6Ht3s' -H 'Content-Type: application/json' -d '{"username":"testuser","password":"testuser","role":"vendor","email":"testuser1@gmail.com","contactNo":"123456","address1":"addr testuser1","organization":"testuser organization1"}' http://localhost:5000/users/6443d217cc5fd2e0d20078e5

{"success":true,"user":{"\_id":"6443d217cc5fd2e0d20078e5","address":"Katilya Hostel","contactNo":"123456","email":"testuser1@gmail.com","organization":"testuser organization1","password":"sha256$zHYsgfQPhRBtinCC$736294217b5af26b591628f86795445eaef3d2cac07a63165930acbee5c4a807","role":"vendor","username":"testuser"}}

32. Test with invalid admin credentials and a valid user\_id and check if the user details update fails and returns the message "Not authorized to access user."

curl -X PUT -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saXJeGbyUHQ' -H 'Content-Type: application/json' -d '{"username":"testuser","password":"testuser","role":"vendor","email":"testuser1@gmail.com","contactNo":"123456","address1":"addr testuser1","organization":"testuser organization1"}' http://localhost:5000/users/6443d217cc5fd2e0d20078e5

{"msg":"Token has expired"}

33. Test cases for deleting a user by user\_id:

curl -X DELETE -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNzU0NCwianRpIjoiN2U0YTMxZTYtYzAzYi00NTg5LTg3ODAtMTMzMzBjZjFhOGZhIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE3NTQ0LCJleHAiOjE2ODIzMTg0NDQsInJvbGUiOiJhZG1pbiJ9.EprN6AJFCh-osb2pVtn3DbZQ6KFu3yh0GAS7Jg6Ht3s' -H 'Content-Type: application/json' http://localhost:5000/users/6443d217cc5fd2e0d20078e5

{"success":true}

34. Deleting ‘admin’ user should not be allowed

curl -X DELETE -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNjUxNywianRpIjoiOWMwOGUxNmItZjAxYy00MzBhLWJjNGQtZWMyOTk1NTEwYWE0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjgyMzE2NTE3LCJleHAiOjE2ODIzMTc0MTcsInJvbGUiOiJhZG1pbiJ9.yDef59oBCsLA5qgbfNTZ99RGT7UyCRx1saXJeGbyUHQ' -H 'Content-Type: application/json' http://localhost:5000/users/6443d217cc5fd2e0d20078e5

{"message":"The admin user cannot be deleted.","success":false}

Tender Management:

For the create\_tender function:

35. Create a new tender with valid inputs and ensure that the response has a success value of True.

For the get\_all\_tenders function:

36. Get all tenders for a specific userid with valid JWT credentials and ensure that the response has a success value of True.

For the get\_tender function:

37. Get a specific tender with a valid tender\_id and valid JWT credentials, and ensure that the response has a success value of True.

For the delete\_tender function:

38. Delete a specific tender with a valid tender\_id and valid JWT credentials, and ensure that the response has a success value of True.

For the assign\_tender function:

39. Assign a tender to a list of valid vendor IDs with a valid tender ID and valid JWT credentials, and ensure that the response has a status value of 'success'.

For the get\_tenders\_assigned\_to\_vendor function:

40. Get all tenders assigned to a specific vendor with a valid vendor\_id and valid JWT credentials, and ensure that the response has a success value of True.

For the update\_tender() function:

41. Valid tender\_id is provided and a valid role tender\_manager in the JWT payload. All the fields such as title, description, start\_date, deadline, location, status are updated successfully.

Quotation Management:

For the create\_quotation() function:

42. Valid tender\_id, vendor\_id, and required fields such as amount, currency, validity\_days are provided. A new quotation is successfully created.

For the get\_quotations\_for\_tender() function:

43. Valid tender\_id is provided and a valid role tender\_manager in the JWT payload. The function returns the list of all quotations for the given tender.

For the get\_quotation\_for\_tender\_and\_vendor() function:

44. Valid tender\_id and vendor\_id are provided and a valid role vendor in the JWT payload. The function returns the quotation for the given tender and vendor.

45. Update an existing quotation with valid details and verify that the quotation details have been successfully updated.

46. Decide on a quotation with a valid status of 'accepted' and verify that the quotation decision has been successfully updated.

47. Decide on a quotation with a valid status of 'rejected' and verify that the quotation decision has been successfully updated.

48. Delete an existing quotation and verify that it has been successfully deleted.

Negative test cases:

User Management:

Test case for registering a new user:

49. Test with incomplete user details and check if the registration fails and returns the message "Bad Request".

50. Test with empty user details and check if the registration fails and returns the message "Bad Request".

Test case for logging in:

51 Test with invalid login credentials of a non-existing user and check if the login fails and returns the message "Invalid credentials".

curl -X POST http://127.0.0.1:5000/login -H 'Content-Type: application/json' -d '{

"username": "admin",

"password": "aa"

}'

{"message":"Invalid credentials.","success":false}

52.. Test with empty login credentials and check if the login fails and returns the message "Bad Request".

curl -X POST http://127.0.0.1:5000/login -H 'Content-Type: application/json' -d '{

"username": "",

"password": ""

}'

{"message":"Invalid credentials.","success":false}

Test case for getting all users:

53. Test with invalid admin credentials and check if the user retrieval fails and returns the message "Not authorized to access users."

curl -X GET -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY4MjMxNzg3NSwianRpIjoiYmIwNDg3OTktNWY4ZS00NjMzLTk5NzctZGE3ZjQ1MzJjNmYzIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6InYxIiwibmJmIjoxNjgyMzE3ODc1LCJleHAiOjE2ODIzMTg3NzUsInJvbGUiOiJ2ZW5kb3IifQ.jAItP9MeAsxnZbye5TmN0xRTBnKa1wsuTpKaP1rMte4' http://localhost:5000/users

{"message":"Not authorized to access users.","success":false}

54. Test with invalid tender\_manager credentials and check if the user retrieval fails and returns the message "Not authorized to access users."

Test case for getting a user by user\_id:

55. Test with invalid admin credentials and an invalid user\_id and check if the user retrieval fails and returns the message "Not authorized to access user."

56. Test with empty user\_id and check if the user retrieval fails and returns the message "Bad Request".

Test case for updating an existing user:

57. Test with invalid admin credentials and an invalid user\_id and check if the user details update fails and returns the message "Not authorized to access user."

58. Test with empty user\_id and check if the user details update fails and returns the message "Bad Request".

Tender Management:

For the create\_tender function:

59. Create a new tender with an existing title and ensure that the response has a success value of False and a message value of 'Tender already exists.'.

60. Create a new tender without a valid JWT and ensure that the response has a success value of False and a message value of 'Not authorized to create tender.'.

For the get\_all\_tenders function:

61. Get all tenders for a specific userid without a valid JWT and ensure that the response has a success value of False and a message value of 'Not authorized to access tenders.'.

For the get\_tender function:

62. Get a specific tender with an invalid tender\_id and valid JWT credentials, and ensure that the response has a success value of False and a message value of 'Tender not found.'.

63. Get a specific tender with a valid tender\_id and invalid JWT credentials, and ensure that the response has a success value of False and a message value of 'Not authorized to access tender.'.

For the delete\_tender function:

64. Delete a specific tender with an invalid tender\_id and valid JWT credentials, and ensure that the response has a success value of False and a message value of 'Tender not found.'.

65. Delete a specific tender with a valid tender\_id and invalid JWT credentials, and ensure that the response has a success value of False and a message value of 'Not authorized to delete tender.'.

For the assign\_tender function:

66. Assign a tender to a list of vendor IDs that do not exist with a valid tender ID and valid JWT credentials, and ensure that the response has a status value of 'fail' and a message value of 'No vendors found with provided IDs'.

67. Assign a tender without a valid JWT and ensure that the response has a status value of 'fail' and a message value of 'Unauthorized access'.

68. For the update\_tender() function:

Invalid tender\_id is provided.

Invalid fields are provided.

Invalid role is provided in the JWT payload.

For the create\_quotation() function:

69. Invalid tender\_id, vendor\_id or required fields such as amount, currency, validity\_days are provided.

70. A quotation with the same tender\_id and vendor\_id already exists.

Invalid role is provided in the JWT payload.

For the get\_quotations\_for\_tender() function:

71. Invalid tender\_id is provided.

72. Invalid role is provided in the JWT payload.

For the get\_quotation\_for\_tender\_and\_vendor() function:

73. Invalid tender\_id or vendor\_id is provided.

74. Invalid role is provided in the JWT payload.

75. Update an existing quotation with invalid quotation ID and verify that the quotation is not found.

76. Update an existing quotation with invalid details and verify that the quotation details have not been updated.

77. Decide on a quotation with an invalid status and verify that the quotation decision has not been updated.

78. Delete a non-existing quotation and verify that it is not found.

79. Delete a quotation with invalid vendor ID and verify that the quotation is not found.

80. Attempt to delete a quotation with a tender ID that does not match the quotation's tender ID and verify that it is not found.

81. Attempt to update a quotation with a user who is not a vendor and verify that the update is not authorized.

82. Attempt to decide on a quotation with a user who is not a tender manager and verify that the decision is not authorized.