

Test Scenario(frontend)	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
creating an account	fill out all the input fields (email, username , password and confir password => select your rank => select profile picture => submit	"email : i@gmail.com, username :manny, password :manny457, confir password:manny457"	creates user and redirect to login page	creates user and redirects to login page	Pass
login	fill out all the input fields (email, password ) => submit	"email : i@gmail.com , password :manny457, " python channel"	logs in and redirects to the channels page	logs in and redirects to the channels page	Pass
create and view channels	write out the name of the channel to be created in the input field => click the create button	"message: how can i learn python" image "image.jpg"	creates and displays python channel	creates and displays python channel	pass
create and view questions	write out the name of the question to be created in the input field => add a picture optionally => click the create button	"message: how can i learn python" image "image.jpg"	creates and displays inputted question and the image	creates and displays inputtedquestion and the image	pass
create and view replies	click on the specific question you want to relp to => write out the name of the reply to be created in the input field => add a picture optionally => click the create button	null	creates and displays inputted replies and the image	creates and displays inputted replies and the image	pass
like messages/replies	click on the specific question like button for the message or reply you want to like	null	displays the updated like count of the message	displays the updated like count of the message	pass
dislike messages/replies	click on the specific question like button for the message or reply you want to dislike	null	displays the updated the dislike count of the message	displays the updated dislike count of the message	pass
remove like or dislike	after already liking or dialking a message => click on the specific question like button for the message or reply you want to re,ove like or dislike	null	displays the updated the like or dislike count of the message	displays the updated the like or dislike count of the message	pass
search	click search on the navbar to go to the search page => choose whether to search through channels or messages , => input search query	select channels and search "python" or select messages and search " how can i learn python"	displays "python channel" or displays "how can i learn python "message	displays "python channel" or displays "how can i learn python "message	pass
view and edit profile	click the username on the navbar to go to the profile page to view it => click onn edit , => fill it out => click submit	username "newusername" rank ="intermediate"	app logs user out to handle change and update user	app logs user out to handle change and update user	pass
logout	click on logout on the sidebar	null	user gets logged out and redirected back to the landing page	user gets logged out and redirected back to the landing page	pass
testing if a user who isnt logged in can access pages	change the url on the site	url : localhost300/profile	user gets kicked back out and redirected back to the landing page	user gets kicked back out and redirected back to the landing page	pass
Test Scenario(backend) Endpoints	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
POST /register	send a POST request to http://localhost:8080/register with the required parameters (username, password, email, rank) in the request body.	(username, password, email, rank) in the request body.	the user is registered successfully.	the user is registered successfully.	
POST /login	Check the response to ensure that the user is registered successfully. send a POST request to http://localhost:8080/login with the email and password you used during registration.	email and password you used during registration.	successful login channel is created successfully.	successful login channel is created successfully.	
POST /create	Send a POST request to http://localhost:8080/create with the required parameters (channelName, userid, username)	(channelName, userid, username)	message is posted successfully.	message is posted successfully.	
POST /create_message	Send a POST request to http://localhost:8080/create_message with the required parameters (channel_id, content, userid, username).	(channel_id, content, userid, username).	you receive the list of messages.	you receive the list of messages.	
GET /messages	a GET request to http://localhost:8080/messages to get a list of messages.	channel_id	the reply is posted successfully.	the reply is posted successfully.	
POST /create_replies	Send a POST request to http://localhost:8080/create_replies with the required parameters (reply, message_id, userid, username).	(reply, message_id, userid, username).	you receive the list of replies.	you receive the list of replies.	
GET /replies	Send a GET request to http://localhost:8080/replies to get a list of replies.	message_id	the user information is updated successfully.	the user information is updated successfully.	
POST /updateuser	Send a POST request to http://localhost:8080/updateuser with the required parameters (editusername, editedUserRank) and user_id in the query parameters.	(editusername, editedUserRank) and user_id in the query parameters.	the upvote is processed successfully.	the upvote is processed successfully.	
POST /upvote	Send a POST request to http://localhost:8080/upvote with the required parameters (messageid, userid).	parameters (messageid, userid).	the downvote is processed successfully	the downvote is processed successfully.	
POST /downvote	Send a POST request   http://localhost:8080/downvote with the required parameters (messageid, userid).	parameters (messageid, userid).	the user is logged out.	the user is logged out.	
GET /logout	Send a GET request to http://localhost:8080/logout to log out the user	n/a	the correct authentication status is returned.	the correct authentication status is returned.	
GET /authUser	Send a GET request to http://localhost:8080/authUser to check if the user is authenticated	n/a	retrieves information about all users, messages, and channels.	retrieves information about all users, messages, and channels.	
GET /adminusers	Send a GET request to <a href="http://localhost:8080/adminusers">http://localhost:8080/adminusers</a>	n/a	retrieve information about all users, messages, and channels.	retrieves information about all users, messages, and channels.	
GET /adminmessages	Send a GET request to http://localhost:8080/adminmessages	n/a	retrieve information about all users, messages, and channels.	retrieves information about all users, messages, and channels.	
GET /adminchannels	Send a GET request to <a href="http://localhost:8080/adminchannels">http://localhost:8080/adminchannels</a>	n/a	retrieve information about all users, messages, and channels.	retrieves information about all users, messages, and channels.	