Sprint 1 Testing

User Front-end

Operation	No	Test case	Precondition	Result	Note
Check Home page	1.1	Access the website by the URL: http://localhost:3000/		The browser shows the home page.	
Check user icon	1.2	1. Hover on the icon in the top-right corner. 2. Click the options in the dropdown menu unfolded.	1.1	1. Unfold a dropdown menu for more actions. 2. Can jump to correct page, can log out, and allow for login/sign-up.	
Jump to login page	1.3	Click the login button in the dropdown menu unfolded by the user icon / try user operations as a visitor.	1.1, 1.2	Redirect to the login page.	
Check login	1.4	1. Enter valid user id and password. 2. Enter invalid user id or password.	1.3	1. Login successfully and redirect to the home page. 2. Alert of error message.	
Check Sign up	1.5	1. Click the sign-up button at the bottom of the login page. 2. Enter user id and password, click submit button.	1.4	1. Pop up a form for sign-up. 2. Return sign-up result. (Log in and redirect to the home page if success)	
Check Payment	1.6	1. Enter the name, card number, cvc, and expiry date of credit card. 2. Click the pay button in the payment page.	1.1	1.When the information of credit card is wrong, there will be prompt 2. Payment successfully and display the notification.	Only tested the supposed situation in Front Server. Wait for interaction between user front-end and back-end server.
Check Map Interface Display	1.7	1. Check whether the map is the latest version of MapBox. 2. Check that the component layout on the map is correct.		Version and layout are as expected.	
Check Draw Polygon	1.8	1. Click on the draw polygon button. 2. Draw a polygon anywhere on the map. 3. Keep the polygon on the map. 4. Delete a polygon drew on the map.	1.7	Drawing, retaining and deleting are successful.	

Admin Front-end

Operation	No	Test case	Precondition	Result	Note
Load the platform	1.1	1. Run the program and show the main page.	Logged in as an admin	The program can run without warning errors. The platform enables the administrator to operates on modules and users.	
See modules	1.2	1.Click the "Module Control Space" button in the left menu	1.1	The user can view all modules	Modules are searchable as optional functionality
See module information	1.3	1.Click specific module	1.2	The user could see one specific module information	
Add a new module	1.4	1.Click the Add button 2.Fill in the module information 3.Submit the form	1.2	A piece of new module information is added to the system	

Edit, Enable and Disable modules	1.5	1.Click the "Edit" button 2. Click the "Enable" button 2.1.Edit modules 2.2.Click the "Submit" button to upload updated edit 3. Click the "Disable" button	1.3	The user could change the module content and upload changes.	
See users	1.6	1.Click the "Users" button	1.1	The admin user could see all the users in a table.	Users are searchable as optional functionality
See user details	1.7	1.Select and click one of the users in the list	1.6	The user information is popped up.	

Back-end

story ID		1.1
Task		Check valid user's privilege/permission
Test Type		Functional
Objective		Safeguard user's activity in accordance to his/her authorized action
setup		1. Get a valid token from the Eratos server
Excepted Outcome		The function should return user details for a valid token.
Acceptance Criteria	**Given**	a valid token
	When	I pass the token to the authorisation function
	Then	I should get the user details successfully
Test result		PASS
Notes		The story 1 is separated into 2 sub-stories (valid users and invalid users), and this task is for testing valid users.
tester		Yan Dai
Date		05/05/2021

story ID		1.2
Task		Check invalid user's privilege/permission
Test Type		Functional
Objective		Prevent the random token from invalid users
setup		1. Generate a random token
Excepted Outcome		The function returns an error for a randomly generated token.
Acceptance Criteria	**Given**	a randomly generated token
	When	I pass the token to the authorisation function
	Then	I should get an authorisation error

Test result	PASS
Notes	The story 1 is separated into 2 sub-stories (valid user and invalid user), and this task is for testing invalid users.
tester	DEYOU ZOU
Date	05/05/2021

story ID		2.1
Task		Create a resource
Test Type		Functional
Objective		Create resources for further operation
setup		1. 1. Get a valid token from the Eratos server 2. Start the server and wait for request from the front-end 3. Parse the request from front-end into valid type and name for resources
Excepted Outcome		Get details of newly created resources from the Eratos server
Acceptance Criteria	**Given**	received a request from the front-end
	When	I parse the request and send it to the Eratos server
	Then	I should get the information about the newly created resource
Test result		PASS
Notes		The story 2 is separated into 3 sub-stories. 2.2 and 2.3 will be done in sprint 2.
tester		En Wen Tsai
Date		05/05/2021