Charter Introduction:

Identify capabilities and areas of potential instability of the "rest api todo list manager". Identify documented and undocumented "rest api todo list manager" capabilities.
For each capability create a script or small program to demonstrate the capability.
Exercise each capability identified with data typical to the intended use of the application.

Environment:
MacOS, Screen Resolution 1440x900 (Arturo Mory Ramirez)
MacOS, Screen Resolution 2560x1600 (Yida Pan)

Testers:
Arturo Mory Ramirez (260989669)
Yida Pan (260942327)

Date:
19:30 PM, 01/02/2024

Run:
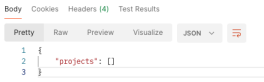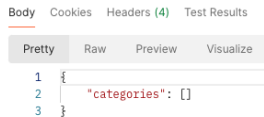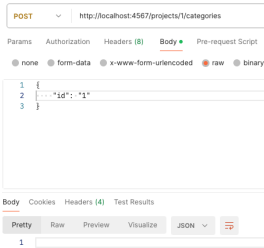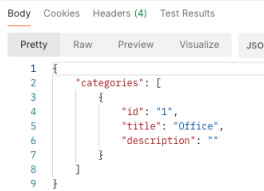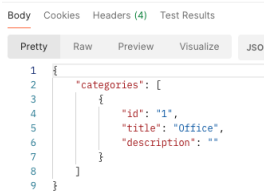java -jar runTodoManagerRestAPI-1.5.5.jar

Area:
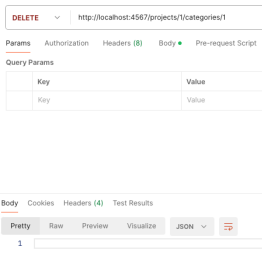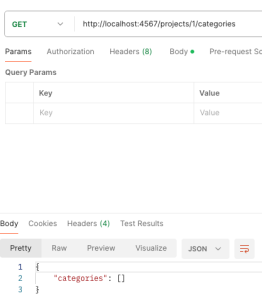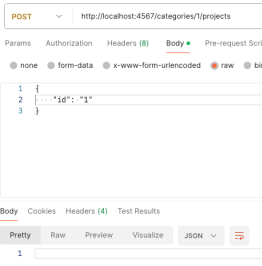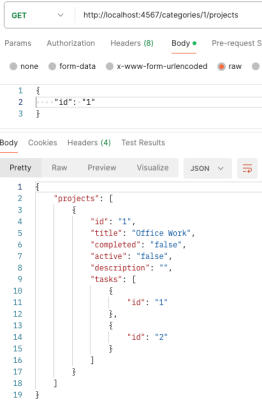Interoperability of the main functions and capabilities of the system
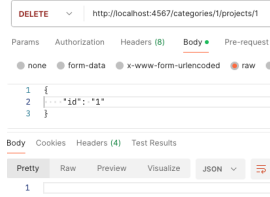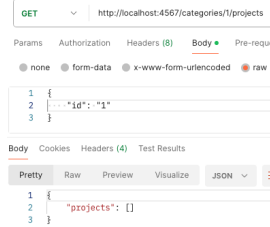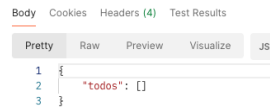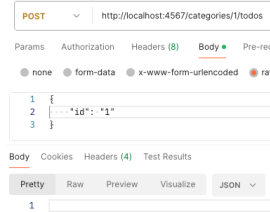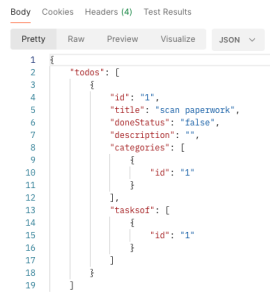
Session Breakdown:
45 minutes total
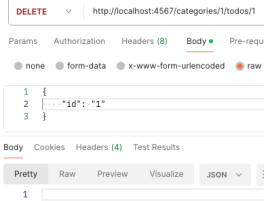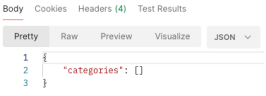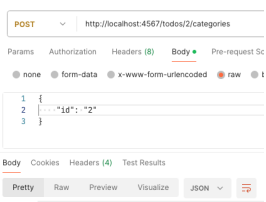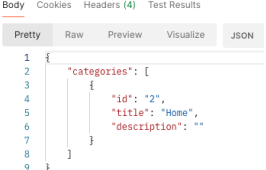
Test Detail:
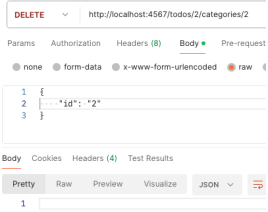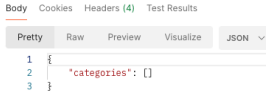
Time	Tester	Request	Type	Note	Output in JSON file
19:30	Arturo	http://localhost:4567/projects	GET	Worked as expected, gets all the projects	

19:31	Arturo	http://localhost:4567/categories	GET	Worked as expected, gets all the categories	
19:32	Arturo	http://localhost:4567/categories/projects	GET	Worked as expected, gets all the projects in the categories (empty)	
19:33	Arturo	http://localhost:4567/projects/categories	GET	Worked as expected, gets all the categories in the projects (empty)	
19:35	Arturo	http://localhost:4567/projects/1/categories	POST	Create relationship between a category and project with no category id, error message as expected	
19:37	Arturo	http://localhost:4567/projects/1/categories	POST	Create relationship with a proper id of "1"	
19:39	Arturo	http://localhost:4567/projects/1/categories	GET	Verify that for project id 1 a category has been created (successful relationship)	
19:40	Arturo	http://localhost:4567/projects/categories	GET	Verify that for all the projects the category just created appears too	

19:42	Arturo	http://localhost:4567/projects/1/categories/1	DELETE	Delete the relationship between the category and the project	
19:44	Arturo	http://localhost:4567/projects/1/categories	GET	Verify that it was deleted for projects and that categories is now empty	
19:45	Arturo	http://localhost:4567/categories/1/projects	POST	Create new category of id 1 in project for category of id 1	
19:47	Arturo	http://localhost:4567/categories/1/projects	GET	Verify that relationship has successfully been created	

19:49	Arturo	http://localhost:4567/categories/1/projects/1	DELETE	Delete the relationship of project id 1 for category id 1	
19:51	Arturo	http://localhost:4567/categories/1/projects	GET	Verify that it was deleted for categories of id 1 and that projects is now empty	
19:52	Arturo	http://localhost:4567/categories/todos	GET	Verify that it was deleted for categories of all ids and that projects is now empty	
19:53	Arturo	http://localhost:4567/todos/categories	GET	Retrieve successfully all categories for todos	
19:55	Arturo	http://localhost:4567/categories/1/todos	POST	Create a new todos of id 1 for category of id 1	
19:57	Arturo	http://localhost:4567/categories/1/todos	GET	Verify that the relation has successfully been created as now there is a todos for category 1	

19:59	Arturo	http://localhost:4567/categories/1/todos/1	DELETE	Delete the relationship of todos id 1 for category id 1	
20:01	Arturo	http://localhost:4567/categories/todos	GET	Verify that it has been successfully deleted and that todos is now empty for all categories	
20:03	Arturo	http://localhost:4567/todos	GET	Retrieve successfully all the todos	
20:04	Arturo	http://localhost:4567/todos/2/categories	GET	Retrieve as expected an empty categories for todos of id 2	
20:06	Arturo	http://localhost:4567/todos/2/categories	POST	Create a category with id 2 for the todos id 2	
20:08	Arturo	http://localhost:4567/todos/2/categories	GET	Verify that the relation for todos id 2 and a category of id 2 has successfully been created	

20:10	Arturo	http://localhost:4567/todos/2/categories/2	DELETE	Delete this category of id 2 for the todos of id 2	
20:12	Arturo	http://localhost:4567/todos/2/categories	GET	Verify that the relation has been successfully deleted by retrieving a now empty categories for todos id of 2	
20:15	Arturo	http://localhost:4567/todos/categories	GET	Verify that only the relation that we deleted has been successfully deleted, leaving with another already existing category of id 1 for todos of id 1	