```
============================================================
SESSION NOTES – ECSE 429 PROJECT PART A
============================================================
```

Tester Name: Noah
Student ID: 260978984
Email: noah.munro-kagan@mail.mcgill.ca
Date: 06/10/2025
Start Time: 2:30 pm
Duration: 45 minutes
Build Under Test: runTodoManagerRestAPI-1.5.5.jar
Environment: macOS, Postman, localhost:4567

```
----------------------------------------------------------
CHARTER
----------------------------------------------------------
```
Test the behavior and stability of all "/projects" endpoints.
Focus: CRUD operations, relationship consistency with todos & categories,
error handling, and undocumented behaviors.

```
----------------------------------------------------------
TEST LOG
----------------------------------------------------------
```
Format: [Timestamp] | [Action/Endpoint] | [Result] | [Observation]

[2:30 PM] | GET /projects | 200 OK | Empty list returned successfully.
[2:32 PM] | POST /projects (valid JSON) | 201 Created | New project created with ID=15.
[2:34 PM] | GET /projects/1 | 200 OK | Data matches creation payload.
[2:36 PM] | PUT /projects/1 | 200 OK | Fields updated correctly.
[2:38 PM] | DELETE /projects/1 | 200 OK | Project deleted successfully, returned number 1.
[2:39 PM] | GET /projects/1 | 404 Not Found | Confirmed deletion.
[2:42 PM] | POST /projects/999 | 404 Not Found | No such project entitiy found with GUID or ID 999
[2:43 PM] | POST /projects (invalid JSON) | 400 Bad Request | expected name at line 5 column 2…
[2:44 PM] | POST /projects (valid JSON) | 201 Created | New project created with ID=17.
[2:45 PM] | POST /projects/17/tasks | 201 Created | Returned the ID used as input in the body.
[2:46 PM] | POST /projects/18/tasks (nonexistent project)| 404 Not Found | Proper error message
[2:48 PM] | POST /projects/17/tasks | 201 Created | Returned the ID used as input in the body.
[2:49 PM] | HEAD /projects/ | 200 OK | Returned the ID 1
[2:52 PM] | POST /projects/17/tasks | 201 Created | Returned the ID used as input in

the body.
[2:54 PM] I GET /projects/17/tasks I 200 OKI Returned tasks for project 17
[2:56 PM] I DELETE /projects/17/tasks/:1 I 200 OK I Returned 1
[2:57 PM] I GET /projects/17/tasks I 200 OKI Returned empty list, verify deletion
[3:00 PM] I POST /projects/17/categories I 201 Created I Returned body
[3:02 PM] I GET /projects/17/categories I 200 OK I Returned added category data
[3:04 PM] I DELETE /projects/17/categories I 200 OK I Returned 1
[3:06 PM] I POST /projects/999/tasks I 404 Not Found I Returned appropriate error code
[3:06 PM] I POST /projects/999/tasks I 404 Not Found I Returned appropriate error code
[3:07 PM] I Delete /projects/999/tasks/999 I 404 Not Found I Returned appropriate error code
[3:08 PM] I POST /projects (valid JSON) I 201 Created I New project created with ID=18.
[3:09 PM] I POST /projects/18/tasks (5x same body) I 201 Created I Returned the ID used as input in the body.
[3:10 PM] I GET /projects/17/tasks I 200 OK I Returned the ID, only once
[3:12 PM] I POST /projects/18/tasks (body id = "9") I 404 Not Found I Error message not clear: read could not find thing matching value for id
[3:14 PM] I PUT /projects/18 (change "id" to 12, clicked multiple times) I 200 OK I Returned request body
[3:16 PM] I POST /projects/18/tasks (5x same body) I 201 Created I Returned the ID used as input in the body.
[3:17 PM] I GET /projects/ I 200 OK I Returned project 18, no project 12

...

FINDINGS

Capabilities:
- CRUD on /projects works correctly.
- Relationships to todos and categories can be created and retrieved.
- HEAD and OPTIONS endpoints return headers as documented.

Issues:
- There is a persistent tally of task number, so when tasks or projects are created then deleted, the next created one continues has an ID one higher than the previous one
- Seems to accept an input ID for PUT on a project yet doesn't change anything
- Unclear error message for adding task to project with incorrect ID

NEW TEST IDEAS

- Try XML format requests.
- Stress test creation of 100+ projects.

- Confirm data persistence after API restart.

END OF SESSION

End Time: [3:17PM]
Summary: The API works really well, there are not issues that can be found in creating updating and deleting data at the 1 at a time scale. The error handling works as well and stops improperly formatted requests from going through.