

Introduction.....	3
Building Process	3
Database.....	3
Thought process	5
Encountered Issues.....	5
Pros and cons of design	5
Api Architecture	5
Screenshots & github repository	5
Potential enhancements.....	6

Introduction

This is the introductino

Building Process

Started from designing the database(Entity-relational Diagram)

Database

Διάγραμμα Οντοτήτων Συσχετίσεων

... βλέπε αρχείο **ΔΟΣ.pdf**

Υποθέσεις

1. Κάθε έργο (**project**) αποτελείται από πολλές επιμέρους θυγατρικές εργασίες (**tasks**). 1-N
2. Κάθε εργασία (**task**) ανήκει αποκλειστικά σε ένα γονικό έργο (**parent project**).
3. Κάθε έργο (**project**) διαθέτει έναν και μόνο ιδιοκτήτη χρήστη (**owner user**). 1-N
4. Κάθε χρήστης (**user**) μπορεί να είναι ιδιοκτήτης πολλών έργων (**projects**).
5. Σε περίπτωση διαγραφεί το έργο(project), θα διαγράφεται αυτόματα και οι θυγατρικές του εργασίες(tasks) από την βάση δεδομένων(CASCADE on Deletion).
6. Σε περίπτωση διαγραφεί ένας χρήστης(user) θα διαγραφεί αυτόματα και τα έργα που του ανήκουν από την βάση δεδομένων (CASCADE on Deletion).
7. Κατά την εισαγωγή κάθε

Ενεργοποιητής

- **Prevent_edit_completed_task:** Εάν μια εργασία (task) είναι ολοκληρωμένη, η βάση δεδομένων αποτρέπει οποιαδήποτε επεξεργασία της. Επιπλέον, όταν το πεδίο progress λάβει την τιμή 100 (ολοκλήρωση), ενημερώνεται αυτόματα και το πεδίο finish_date.
- **Update_project_progress:** Κάθε φορά που τροποποιείται μια εργασία (task), το πεδίο progress του αντίστοιχου γονικού έργου (parent project) ενημερώνεται αυτόματα βάσει του ποσοστού ολοκλήρωσης των εργασιών του.
- **Update_project_progress_after_insert:** Παρόμοια με τον προηγούμενο ενεργοποιητή, αλλά ενεργοποιείται κατά την εισαγωγή (insert) νέας εργασίας σε ένα έργο.

Sql Ερωτήματα(queries):

Projects CRUD operations

- Get all projects of user with id = ?:

```
SELECT projects.*  
FROM projects  
JOIN users ON users.id = projects.owner_id  
WHERE users.id = ?;
```

- Insert new project:

```
INSERT INTO projects (title, description, owner_id)  
VALUES (?, ?, ?);
```

- Update project with id = ?:

```
UPDATE projects  
SET title = ?,  
description = ?  
WHERE owner_id = ?;
```

- Delete a project with id = ?:

```
DELETE  
FROM projects  
WHERE projects.id = ?;
```

Tasks CRUD operations

- Get all tasks of project with id = ?:

```
SELECT tasks.* FROM tasks JOIN projects ON projects.id = tasks.parent_project_id  
WHERE projects.id = ?;
```

- Insert a new task on project with id = ?:

```
INSERT INTO tasks (title, description, progress, parent_project_id) VALUES (?, ?, ?, ?);
```

- Update task with id = ?:

```
UPDATE tasks  
SET
```

```
title = ?,  
description = ?,  
progress = ?  
WHERE id = ?;
```

- Delete task with id = ?:

```
DELETE  
FROM tasks  
WHERE id = ?;
```

Though process

I though create databse, implement CRUD operations, add OAth2 authentication/authorization using keyCloak

Encountered Issues

While building the CRUD operation POST method

Pros and cons of design

Although the api provides ...

Api Architecture

Initially, ...

Screenshots & github repository

In this section, we provide ...

Potential enhancements

Let's discuss the feature implementations...