Πολυτεχνείο Κρήτης
ΠΛΗ513: Υπηρεσίες στο Υπολογιστικό Νέφος και την Ομίχλη

Σύστημα Διαχείρισης Έργων (PMS)

Αντώνιος Μουτσόπουλος
Α.Μ.: 2021030024

11 Ιανουαρίου 2026

Περιεχόμενα

1 Εισαγωγή

Η παρούσα αναφορά περιγράφει τον σχεδιασμό και την υλοποίηση ενός σύγχρονου συστήματος διαχείρισης έργων (Project Management System - PMS). Στόχος της άσκησης ηταν η δημιουργία μιας πλατφόρμας που επιτρέπει σε ομάδες να διαχειρίζονται χρήστες, ομάδες εργασίας και εργασίες (tasks) σε ένα κατανεμημένο περιβάλλον. Το σύστημα ακολουθεί την αρχιτεκτονική των μικροϋπηρεσιών (microservices), διασφαλίζοντας την ανεξαρτησία των επιμέρους λειτουργιών και την ευκολία στη συντήρηση.

2 Περιγραφή Εφαρμογής

2.1 Λειτουργικότητα

Η εφαρμογή προσφέρει τις εξής βασικές λειτουργίες:

- **Διαχείριση Χρηστών:** Εγγραφή, σύνδεση (authentication) και ρόλοι χρηστών (Admin, Team Leader, Member).
- **Διαχείριση Ομάδων:** Δημιουργία ομάδων και ανάθεση μελών σε αυτές.
- **Διαχείριση Εργασιών:** Δημιουργία, ανάθεση, παρακολούθηση κατάστασης και μεταφόρτωση αρχείων για κάθε εργασία.
- **Πίνακας Ελέγχου (Dashboard):** Συγκεντρωτική εικόνα των δραστηριοτήτων για τον χρήστη.

2.2 Τεχνολογικό Υπόβαθρο (Stack)

- **Frontend:** HTML5, CSS3 και Vanilla JavaScript. Η επιλογή του Vanilla JS έγινε για τη μείωση των εξωτερικών εξαρτήσεων και την απλούστερη χρηση τους
- **Backend:** Python 3 χρησιμοποιώντας την εγγενή βιβλιοθήκη `http.server`. Αυτό επιτρέπει τον πλήρη έλεγχο των HTTP requests/responses χωρίς την ανάγκη βαριών frameworks.
- **Database:** MySQL 8.0 3.2
- για την αποθήκευση των δεδομένων με χρήση του `mysql-connector-python` για την επικοινωνία με το backend.
- **Web Server:** Nginx για το σερβίρισμα των στατικών αρχείων του frontend.

3 Υλοποίηση

3.1 Επιτυχής Υλοποίηση

Υλοποιήθηκε πλήρως η επικοινωνία μεταξύ των υπηρεσιών μέσω REST APIs. Κάθε υπηρεσία (`user-service`, `team-service`, `task-service`) λειτουργεί σε δικό της container, διασφαλίζοντας την απομόνωση. Η βάση δεδομένων αρχικοποιείται αυτόματα με σχήμα και δεδομένα μέσω Docker volumes.

3.2 Προκλήσεις και Λύσεις

1. **Race Conditions στην Εκκίνηση:** Κατά την εκκίνηση των containers, οι υπηρεσίες Python προσπαθούσαν να συνδεθούν στη MySQL πριν αυτή είναι έτοιμη. Λύθηκε με την ενσωμάτωση ενός retry loop στον κώδικα εκκίνησης κάθε υπηρεσίας.

2. **CORS (Cross-Origin Resource Sharing):** Λόγω της φύσης των μικροϋπηρεσιών που τρέχουν σε διαφορετικά ports, απαιτήθηκε η ρύθμιση ειδικών headers στα responses για την αποδοχή requests από το frontend.

4 Ιδιαίτερα Στοιχεία / Καινοτομίες

Το σύστημα υιοθετεί μια "lightweight" προσέγγιση στο backend, αποφεύγοντας frameworks όπως το Django ή το Flask, γεγονός που καθιστά τα Docker images εξαιρετικά μικρά και γρήγορα στην εκκίνηση. Επίσης, η χρήση του `phpMyAdmin` εντός του περιβάλλοντος ανάπτυξης διευκολύνει τον άμεσο έλεγχο της κατάστασης των δεδομένων.

5 Migration – Deployment στη Google Cloud Platform

5.1 Containerization

Η εφαρμογή έχει πλήρως "containerized" μορφή. Κάθε μικροϋπηρεσία διαθέτει το δικό της `Dockerfile`, βασισμένο σε ελαφριά images της Python (`python:3.13-slim`).

6 Οδηγίες Πρόσβασης

- **Site:** `http://34.7.86.218:8000/`
- **Credentials:**

Ρόλος	Username	Password	Email
Administrator	admin	admin	admin@example.com
Team Leader	leader1	leader	leader1@example.com
Team Leader	leader2	leader	leader2@example.com
Member	member1	member	member1@example.com
Member	member2	member	member2@example.com
Member	member3	member	member3@example.com
Member	member4	member	member4@example.com

Πίνακας 1: Πίνακας διαπιστευτηρίων αρχικών χρηστών.

- **phpMyAdmin:** `http://34.7.86.218:8083/`