

Project and Work Tracking System

Technical documentation

Mentor: Prof. Klen Čopič Pucihar, PhD

Authors: Una Novaković and Luka Matič

University of Primorska

FAMNIT

2026

Contents

1. Purpose of the document	3
2. System architecture and Technology stack	3
3. Project Structure	4
4. Installation and Configuration	5
4.1 Database setup	5
4.2 Database Configuration.....	5
4.3 Running the Application.....	6
5. User Roles	6
6. Future improvements	7

1. Purpose of the document

This document provides technical documentation for the *Project Work Tracking System*. It describes the system architecture, database design, project structure, installation process, and main functionalities. The system was developed as a web-based application for managing projects, tasks, and employee work hours.

2. System architecture and Technology stack



Figure 2.1 High-level system architecture

Table 2.1 Technology stack

Layer	Technology
Frontend	HTML, CSS, BOOTSTRAP
Backend	PHP
Database	MySQL
Development environment	XAMPP
Version Control	Git, GitHub

3. Project Structure

```
project-work-tracking/  
|  
+-- backend/  
|   +-- db.php  
|   +-- db.example.php  
|  
+-- database/  
|   +-- schema.sql  
|   +-- seed.sql  
|  
+-- public/  
|   +-- login.php  
|   +-- register.php  
|   +-- projects.php  
|   +-- tasks.php  
|   +-- time_entries.php  
|   +-- ...  
|  
+-- README.md  
+-- .gitignore
```

Figure 3.1 Project directory structure

Figure 3.1 shows the directory structure of the Project Work Tracking System. The project is organized into separate directories for the backend, database scripts, public web pages, and project configuration files.

Seed Data

The file *seed.sql* contains sample data used for testing and demonstration purposes. It inserts predefined roles, users, projects, tasks, and time entries so that the system can be explored immediately after installation without requiring manual data entry.

4. Installation and Configuration

4.1. Database setup

1. Start Apache and MySQL using XAMPP.
2. Open phpMyAdmin.
3. Create a new database named: `project_work_tracking`
4. Import the file: `database/schema.sql`

This file creates the database structure and all required tables.

5. Import the file: `database/seed.sql`

This file inserts sample data including users, roles, projects, tasks, and time entries.

6. Verify that all tables have been created successfully.

4.2. Database Configuration

Open:

`backend/db.php`

Configure the database connection settings:

- Host
- Database name

- Username
- Password

Example:

Host: localhost

Database: project_work_tracking

Username: root

4.3. Running the Application

1. Start Apache and MySQL in XAMPP.
2. Place the project inside: xampp/htdocs/
3. Open a web browser and navigate to: <http://localhost/project-work-tracking/public/>
4. Register a new account or log in using an existing account.

5. User Roles

The system supports three user roles with different access permissions.

Administrator

- Manage users
- Manage projects
- Manage tasks
- Record work hours
- View project progress

Project Manager

- Create and manage projects
- Create and assign tasks
- Record work hours
- View project progress

Employee

- View assigned tasks
- Update task status
- Record work hours
- View project progress

Security features

- Passwords are hashed using PHP password hashing functions.
- User sessions are used for authentication.
- Role-based access control restricts access to protected functionalities.
- Server-side validation is performed before database operations.

6. Future improvements

Future maintenance may include:

- **Email notifications** to automatically inform users about new task assignments, approaching deadlines, and project updates.
- **File attachments** to allow users to upload and share documents, images, and other files related to projects and tasks.

- **Advanced reporting** to provide more detailed insights into project progress, employee productivity, and work-hour statistics.
- **Gantt chart** integration to visualize project schedules, task dependencies, and deadlines in a timeline format.
- **Deployment to a production server** to make the system accessible to multiple users through the internet and support real-world organizational use.
- In a production version, the **administrator role could be split** into a technical and an operational administrator. This would reduce unnecessary access rights and follow the principle of least privilege.