StudyTask – Technical Documentation (High-Level)

1. System Overview

StudyTask is a web-based task and group management application designed for students. The system follows a classic client-server architecture with a PHP-based backend, a browser-based frontend, and a relational MySQL database.

2. Architecture

The application is structured into three main layers:
- Frontend (HTML, CSS, JavaScript)
- Backend (PHP REST-like API)
- Database (MySQL hosted on AWS RDS)

The frontend communicates with the backend exclusively via HTTP requests (fetch API). All business logic and authorization checks are handled server-side.

3. Frontend

The frontend is implemented using plain HTML, CSS, and JavaScript. It provides separate views for users and administrators.

Key responsibilities:
- User interaction and navigation
- Sending API requests
- Rendering server responses
- Basic client-side validation

4. Backend

The backend is implemented in PHP and organized in a modular folder structure.

Core components:
- Authentication and session handling
- Authorization (admin-only endpoints)
- Business logic (tasks, groups, users)
- Centralized database access via PDO

The backend exposes endpoints that return JSON responses only.

5. Database

The database is a MySQL instance hosted on AWS RDS.

Main entities:
- users
- tasks
- groups
- group_members
- contact_messages

Relational integrity is enforced using foreign keys.

6. Authentication & Authorization

Authentication is session-based using PHP sessions.

Authorization rules:
- Regular users can manage their own tasks and groups
- Administrators can manage users and system-wide data

Admin-only endpoints are protected by a dedicated middleware file.

7. Deployment & Environment

The system is designed to run locally using XAMPP or in a hosted environment.

Configuration is managed via environment variables with fallback defaults defined in the database configuration file.

8. Security Considerations

Security measures include:
- Password hashing using PHP password_hash
- Prepared statements to prevent SQL injection
- Session-based access control
- Separation of admin and user privileges

9. Limitations

This project is a prototype developed for academic purposes.

Not intended for production use without further hardening, logging, and scalability improvements.