TASK MANAGEMENT WEB APPLICATION

1. Project Goal

This project aims to develop a web-based task management system where users can add, edit, categorize, and track the completion status of their tasks. The project also aims to practically improve both frontend and backend development skills as well as database management knowledge.

2. Groups

The project should be carried out in groups of up to 4 people. Groups must write their members on the list at the link by Friday, October 10, 2025, 17:00. After due date, the link will not be accessible.

3. Technologies to be Used

- Frontend (User Interface): A dynamic user interface including task listing, adding, editing, and statistics screens should be developed.
- Backend (Server Side): API endpoints, user authentication, and task management via database should be implemented.
- Database: User information and task records should be stored in an appropriate database model.
- Security and Encryption: To ensure data confidentiality and integrity, the encryption and security practices should be implemented.
- Additional Tools: Secure user management for login/registration, frontend-backend data communication, and graphical display of task statistics.

4. Database Structure

The database structure will include two main collections:

- users: Stores user information (id, name, email, password)
  Note: Password field must store passwords after encryption.
- tasks: Stores task information (id, title, description, category, status, dueDate, dueTime)

5. Backend API Endpoints

METHOD	ENDPOINT	DESCRIPTION
POST	/api/auth/register	Create user registration
POST	/api/auth/login	User login
GET	/api/tasks	List user's tasks
POST	/api/tasks	Add new task

PUT	/api/tasks/:id	Edit task
DELETE	/api/tasks/:id	Delete task
GET	/api/tasks/stats	Get task statistics

6. Frontend Page Flow

- Login/Registration Page: User authentication.
- Task List (Dashboard): Lists all tasks, can be filtered and edited.
- Task Add/Edit Page: Create a new task or update an existing one.
- Statistics Page: Graphical representation of tasks by category and completion status.

7. Additional Features

- Color-coding tasks based on completion status
- Alert colors for approaching task deadlines
- Graphs showing completed/incomplete task distribution by category
- Task filtering feature on the interface (by category and completion status)

8. Extended Requirements

8.1. Task-Based File Management (Attachment Module)

The system shall allow users to upload, view, download and delete files related to specific tasks.

Functional requirements:
- The system shall allow multiple files to be attached to a single task.
- Users shall be able to upload files while creating or editing a task.
- The system shall display a list of attached files with the following information:
    - File name
    - File size
    - Upload date
- Users shall be able to download or preview attachments.
- When a task is deleted, all related attachments shall also be deleted by the system.

File constraints:
- Supported file formats: **PDF, PNG, JPG, DOCX, XLSX**
- Maximum file size: **10 MB per file**
- Files shall only be accessible to authenticated users.
- For each attachment, the following data must be stored:
    - Attachment ID
    - Related Task ID
    - Original File Name
    - Storage Path / URL

- File Size
- Upload Date
- Uploader User ID

8.2. Role-Based Authorization System

The system shall implement role-based user authorization.

Roles:

- **User:** Can create, view, update and delete only their own tasks.
- **Admin:** Can view, manage and delete all tasks and assign tasks to any user.

Functional requirements:

- Each user shall have a role stored in the database.
- User role shall be included in the authentication session/token.
- Role-based access control shall be enforced on the backend.
- Admin users shall have access to an additional admin panel.

Rules:

- A normal user cannot view or modify other users' tasks.
- An admin can assign tasks to other users.
- Unauthorized actions shall be blocked by the system.

9. Presentation Plan

9.1. Midterm Presentation

Presentations will be done before the midterm exam week.

NO	REQUIREMENTS	POINTS
1	Frontend + backend integration (it will be demonstrated during presentation)	20 p
2	Database connection active (it will be demonstrated during presentation)	10 p
3	User registration and login working (user will be added during the presentation)	10 p
4	Invalid login and secure authentication tests (it will be demonstrated during presentation)	12 p
5	Storing password after encryption (it will be shown during presentation)	10 p
6	Interface draft	20 p
7.1	Report –> technologies used (frontend, backend, database, security and encryption, additional tools) and reasons for selection	3 p
7.2	Report –> visual showing frontend + backend integration	3 p
7.3	Report –> visual showing active database connection	3 p
7.4	Report –> visual showing interface draft	3 p
7.5	Report –> visual showing encrypted password	3 p
7.6	Report –> visual showing invalid login outcome	3 p
	Total	100 p

9.2. Final Presentation

Presentations will be done before the final exam week.

NO	REQUIREMENTS	POINTS
1	Managing task-based files (it will be examined on interface during presentation, the code will be shown during presentation)	14 p
2	All backend API endpoints pass tests (it will be demonstrated during presentation, the test code will be shown during presentation)	20 p
3	Role-based authorization system (it will be demonstrated during presentation)	12 p
4	Color tasks based on completion status (it will be demonstrated by adding tasks during presentation)	6 p
5	Show alert colors for approaching deadlines (it will be demonstrated during presentation)	6 p
6	Task filtering feature on the interface (by category, completed/incomplete, it will be demonstrated during presentation)	10 p
7	Graph of completed/incomplete tasks by category (graphical output required, it will be shown during presentation)	10 p
8.1	Report (will be continuation of the first report) –> code for backend API endpoint tests	3 p
8.2	Report –> test results	3 p
8.3	Report –> visual showing completed tasks on the interface	3 p
8.4	Report –> visual showing approaching deadlines	3 p
8.5	Report –> visual showing interface filtering feature	3 p
8.6	Report –> visual showing task distribution	3 p
8.7	Report –> visual showing task-based file feature	4 p
	Total	100 p