**Timesheet Application**

**1. Introduction**

**Purpose**

The purpose of this Timesheet Management Application is to help organizations and employees efficiently manage time logs, project assignments, and task tracking. It streamlines employee productivity, tracks project hours, and automates report generation.

**Scope**

**This application enables:**

* Employees to log time spent on specific tasks or projects.
* Administrators to manage users, projects, and reports.

**2. Features**

**User Management**

* User registration, login, and role-based access control.
* Roles: Admin, User

**Project Management**

* Add, edit, and manage projects.
* Assign projects to employees.

**Time Log Management**

* Employees can submit their daily or weekly time logs.
* Automatic validation of entries.

**Timesheet Approval**

* Managers can review, approve, or reject submitted timesheets.

**Dashboard Management**

**Role-based dashboards displaying relevant information:**

* **Employees**: View assigned projects and submitted timesheets.
* **Admins**: Overall summary and system reports.

**Notifications**

* Email alerts for timesheet submissions and approvals.

**3. System Modules**

**3.1 User Management**

**Functionalities:**

* User registration and login.
* **Role management:** admin, user.

**Roles:**

**Admin**: Can create and manage users.

**Employee**: Submit and manage time logs.

**3.2 Project Management**

**Functionalities:**

Create, edit, and delete projects.

Assign employees to projects.

Set project timelines and details.

**Actors:**

Admin: Full project control.

User: View the Projects.

**3.3 Time Log Management**

**Functionalities:**

Employees can log working hours for specific projects.

Validate daily working hours (e.g., 0–24 hours).

**Actors**:

**Employee**: Submits time logs.

**Admin**: approves or rejects time logs.

**3.4 Dashboard Management**

**Functionalities:**

* **Role-specific dashboards:**
* **Employee Dashboard:** View tasks, logs, and status.
* **Admin Dashboard:** System overview, reports, and user management.

**4. Technical Stack**

**Frontend:**

**Framework**: Angular

**Styling**: Bootstrap and Material UI

**Backend:**

* **Technology**: Node.js with Express.js
* **Database**: MongoDB

**Security**:

* **Password encryption:** bcrypt
* **Authentication**: JSON Web Tokens (JWT)
* HTTPS for secure communication.

**Tools & Libraries:**

* **Backend Testing**: Postman
* **Version Control:** Git / GitHub
* **API Documentation**: Postman

**5. Workflow**

**Employee Workflow**:

* Log in, → View assigned projects, → Add time logs, and submit for approval.

**Admin Workflow:**

* Log in → Manage users, projects → View system

**6. Database Design**

**Tables**:

**Users:**

* username (PK)
* Name, Email, Password, Phone, Department, Business Unit
* usertype(Admin, User)

**Projects:**

* Project ID (PK)
* Project Name, Client Name, Address, Department, Project Type
* Assigned Employees

**TimeLogs:**

* UserId(PK)
* User Name, Date, ProjectName, Task, Hours, TaskStatus

**Timesheets:**

* Timesheet ID (PK)
* User ID (FK)
* Project ID (FK)
* Date, Hours Worked
* Status (Completed, InProcess)

**7. High-Level Design**

**1. System Architecture**

* Frontend: Angular.
* Backend: Node.js with Express.js for APIs.
* Database: Relational database like MySQL/PostgreSQL.
* Authentication: JWT-based authentication for secure login.

**2. Modules and Features**

**User Management**

* Login, Add, Edit, Delete, Search, Filter Users.

**Project Management**

* Add, edit, delete, assign users, assign tasks, search, and filter projects.

**Timelog Management**

* Add and view timelog entries (User/Admin).

**Dashboard Management**

* View planned vs. actual hours, tasks, project status, and user performance.

**3. System Flow**

* User Login → Authenticate and Redirect to Role-Specific Dashboard.

**Admin Panel**

* Manage Users → CRUD, Search, Filter.
* Manage Projects → CRUD, Assign Users and Tasks.
* View Timelogs → All User Entries.
* Generate reports in Dashboard.

**User Panel** → Add/View Timelogs, View Assigned Projects.

**8. Low-Level Design**

**8.1 Database Schema**

**User Table:**

| **Column** | **Type** |
| --- | --- |
| username | Varchar |
| email | Varchar |
| phone | Varchar |
| password | Varchar |
| department | Varchar |
| business\_unit | Varchar |

**Projects Table:**

| **Column** | **Type** |
| --- | --- |
| project\_name | Varchar |
| client\_name | Varchar |
| address | Varchar |
| department | Varchar |
| business\_unit | Varchar |
| project\_type | Varchar |

**Tasks Table:**

| **Column** | **Type** |
| --- | --- |
| task\_id | int |
| project\_id | int |
| task\_name | Varchar |
| planned\_hours | int |
| status | Varchar |

**Tasks Table:**

| **Column** | **Type** |
| --- | --- |
| log\_id | int |
| user\_id | int |
| project\_id | Varchar |
| task\_name | int |
| date | Date |
| hours \_spent | int |
| task\_status | Varchar |

### **8.2 API Endpoints**

#### **User Management**

* **POST /api/users/login**: Login user.
* **GET /api/users**: List all users.
* **POST /api/users**: Add new user.
* **PUT /api/users/:id**: Edit user details.
* **DELETE /api/users/:id**: Delete user.
* **GET /api/users/search?query=xyz**: Search users.
* **GET /api/users/filter?department=IT&business\_unit=HR**: Filter users.

#### **Project Management**

* **GET /api/projects**: List all projects.
* **POST /api/projects**: Add new project.
* **PUT /api/projects/:id**: Edit project.
* **POST /api/projects/:id/assign**: Assign users/tasks.
* **GET /api/projects/search?query=abc**: Search projects.
* **GET /api/projects/filter?department=IT**: Filter projects.

#### **Timelog Management**

* **POST /api/timelogs**: Add new log entry.
* **GET /api/timelogs/user/:id**: View user’s logs.
* **GET /api/timelogs/admin**: View all logs (admin).

#### **Dashboard Management**

* **GET /api/dashboard/planned-vs-actual**: Planned vs Actual Hours.
* **GET /api/dashboard/task-status**: Task status by project.
* \*\*GET /api/dashboard

**9. Class Diagram**

+-----------------------------+

| **User**  |

+-----------------------------+

| user\_id: INT |

| username: STRING |

| email: STRING |

| phone: STRING |

| role: STRING |

| department: STRING |

| business\_unit: STRING |

+-----------------------------+

| + login() |

| + createUser() |

| + editUser() |

| + deleteUser() |

+-----------------------------+

+-----------------------------+

| **Project**  |

+-----------------------------+

| project\_id: INT |

| project\_name: STRING |

| client\_name: STRING |

| department: STRING |

| business\_unit: STRING |

| project\_type: STRING |

+------------------------------+

| + addProject() |

| + editProject() |

| + assignUsers() |

| + assignTasks() |

+------------------------------+

+-----------------------------+

| Task |

+-----------------------------+

| task\_id: INT |

| project\_id: INT |

| task\_name: STRING |

| planned\_hours: INT |

| status: STRING |

+----------------------------+

| + assignTask() |

| + editTask() |

+----------------------------+

+--------------------------+

| **Timelog**  |

+---------------------------+

| log\_id: INT |

| user\_id: INT |

| project\_id: INT |

| task\_name: STRING |

| date: DATE |

| hours\_spent: INT |

| task\_status: STRING |

+----------------------------+

| + addLogEntry() |

| + viewLogs() |

+----------------------------+

+----------------------------+

| **Dashboard**  |

+----------------------------+

| planned\_hours: INT |

| actual\_hours: INT |

| task\_status: STRING |

| user\_performance:INT |

+------------------------------+

| + generateReports() |

+------------------------------+

**Relationships**:

* User 1..\* → 1 Project
* Project 1..\* → 1 Task
* User 1 → \* Timelog
* Timelog \* → 1 Project

**10. Conclusion**

The Timesheet Management Application is an efficient solution for organizations to streamline employee time tracking, project monitoring, and reporting. Its user-friendly design and robust backend architecture ensure scalability, reliability, and security.