**1. Introduction**

The **Employee Time Management System** is designed to help organizations track their employees' work hours, monitor productivity, and optimize task management. The system automates time logging, provides detailed reports, integrates with payroll systems, and offers features to manage tasks effectively, fostering transparency and accountability within the workplace.

**2. Project Setup**

**Prerequisites**

Before setting up the Employee Time Management System, ensure the following prerequisites are installed:

* **JDK 11** or higher.
* **Apache Tomcat 10.1** or higher.
* **MySQL** or any other preferred database.
* **Maven** (for building the project).

**Installation**

Follow these steps to get the system up and running:

1. **Clone the Repository:**

bash

Copy code

https://github.com/Khushikumari25/EmployeeTimeManagement.git

cd EmployeeTimeManagement

1. **Configure the MySQL Database:**
   * Create a new database:

sql

Copy code

CREATE DATABASE employee\_time\_management;

* + Use the SQL scripts located in the db/ folder to create necessary tables for **Users**, **Tasks**, and **Time Logs**.

1. **Configure Tomcat:**
   * Add the project as a web application in Apache Tomcat.
   * Ensure Tomcat is configured with **version 10.1** or higher.
2. **Build the Project Using Maven:** In the root directory of the project, run the following command:

bash

Copy code

mvn clean install

This will create a .war file in the target/ directory.

1. **Deploy to Tomcat:**
   * Copy the .war file from the target/ directory to the webapps/ folder of your Tomcat installation.
   * Start the Tomcat server and visit http://localhost:8080/EmployeeTimeManagement in your browser.

**Configuration**

* **Database Configuration:** Edit the application.properties or database-config.xml (depending on the implementation) to configure database connection strings, user credentials, etc.
* **Tomcat Configuration:** Set the context path for the web application to avoid port conflicts if necessary.

**3. System Overview**

The **Employee Time Management System** is built using Java Servlet technology with a MySQL database for data storage. The application is designed with a modular structure, enabling easy management of time logs, tasks, and user accounts.

Key components:

* **Frontend:** A user-friendly web interface using JSP and HTML, accessible via browsers.
* **Backend:** Java Servlets handle business logic, with data interactions managed through JDBC.
* **Database:** MySQL is used to store user information, task details, time logs, and reports.

**4. Features**

**Core Features:**

* **Time Logging:** Employees can track time spent on tasks using the start/stop functionality or manual entry.
* **Task Management:** Employees and managers can create, edit, view, and delete tasks. Tasks can be categorized by priority, status, and deadline.
* **Task Assignment:** Managers can assign tasks to employees with deadlines and expected time to complete.
* **Reports:** Detailed time reports allow employees and managers to analyze time allocation and productivity.
* **User Registration & Login:** Employees can sign up and log in securely to access their task lists and time logs.
* **Admin Dashboard:** Admin users can manage users, tasks, and settings.

**Additional Features:**

* **Notifications:** Automated notifications for upcoming task deadlines and reminders.
* **Integration with Payroll:** The system integrates with payroll software to calculate employee salaries based on logged hours.

**5. Usage Instructions**

Once the system is deployed successfully, follow these steps to start using the application:

1. **Sign Up:** Register as a new user by providing your username, email, and password.
2. **Login:** Use your credentials to log into your account.
3. **Create Tasks:** Add new tasks, specifying title, priority, category, and due date.
4. **Log Time:** Start and stop logging time against tasks. View logged time in your dashboard.
5. **Edit Tasks:** Modify existing tasks, mark them as complete, or update their details.
6. **View Reports:** Generate detailed reports showing time spent on tasks, project progress, and individual performance.

**6. Testing**

**Unit Testing:**

The application uses **JUnit** for unit testing the service and DAO layers. Unit tests are written to verify the correctness of each module, ensuring the application’s core functionality works as expected.

**Mocking:**

The **Mockito** framework is used to mock dependencies during unit testing, ensuring isolated and accurate tests.

**Integration Testing:**

Tests are conducted to ensure the proper interaction between servlets and the database. Data flow is tested to verify that users can interact with the system without errors.

**7. Security Considerations**

**Password Security:**

User passwords are hashed using **BCrypt** before being stored in the database, ensuring that passwords are never stored in plain text.

**SQL Injection Prevention:**

Prepared statements are used in all database interactions, effectively preventing SQL injection attacks.

**Session Management:**

The system uses **HTTPS** for secure communication and session management to prevent unauthorized access to user accounts.

**8. Contributing**

We welcome contributions to the **Employee Time Management System**!

Steps to contribute:

1. Fork the repository.
2. Create a new branch (git checkout -b feature-branch).
3. Make your changes and commit them (git commit -am 'Add feature').
4. Push the changes to your fork (git push origin feature-branch).
5. Open a pull request on GitHub for review.

**9. License**

This project is licensed under the **MIT License**. See the LICENSE.md file for details.

**10. Acknowledgements**

* **Java Servlet API**: The Java Servlet technology was used to create the backend of this application.
* **Open-Source Libraries**: We acknowledge the use of open-source libraries that contributed to the development process.
* **Java EE Documentation**: Special thanks to the official Java EE documentation for guidance on building web applications.

This **Employee Time Management System** documentation provides a comprehensive guide on setting up, configuring, and using the application, as well as contributing to its further development.