### \*\*Steps to Setup\*\*

1. \*\*Clone the repository\*\*:

```bash

git clone https://github.com/your-username/EduTimeManager.git

cd EduTimeManager

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

sql

Copy code

CREATE DATABASE edu\_time\_manager;

* + Use the SQL scripts in db/ folder to create tables for **Users** and **Tasks**.

1. **Configure Tomcat:**
   * Add the project to your **Tomcat server** as a web application.
   * Set the Tomcat version to 10.1.
2. **Build the project using Maven:**

In the project root folder, run:

bash

Copy code

mvn clean install

This will build the application and create the .war file in the target/ directory.

1. **Deploy the application to Tomcat:**
   * Copy the .war file from target/ directory to the webapps/ folder of your Tomcat installation.
   * Start Tomcat and open your browser. Go to http://localhost:8080/EduTimeManager.
2. **Start Using the Application:**
   * Register a new user.
   * Log in and start managing your tasks.

**Usage Instructions**

Once deployed successfully, you can:

* **Sign Up:** Register with a username, email, and password.
* **Login:** Use your credentials to log in to your account.
* **Create Tasks:** Add tasks with details such as title, category, priority, and due date.
* **View Tasks:** Check your tasks in the dashboard and filter by priority or category.
* **Edit Tasks:** Modify existing tasks or mark them as complete.
* **Delete Tasks:** Remove tasks from your list.

**Testing**

* **JUnit Testing**: Unit tests are written to verify the correctness of service and DAO layers.
* **Mocking**: The **Mockito** framework is used to mock the dependencies in unit tests.
* **Integration Testing**: Ensures proper communication between the servlets and database.

**Security Considerations**

* **Password Security:** User passwords are hashed using **BCrypt** before being stored in the database.
* **SQL Injection Prevention:** Prepared statements are used to prevent SQL injection vulnerabilities.
* **Session Management:** User sessions are used to manage logged-in users.

**Contributing**

Feel free to contribute to this project by opening an **issue** or submitting a **pull request**.

**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 forked repository (git push origin feature-branch).
5. Open a **pull request** on GitHub.

**License**

This project is licensed under the MIT License - see the <LICENSE.md> file for details.

**Acknowledgements**

* Thanks to the open-source community for providing libraries and tools that made this project possible.
* Special thanks to the **Java Servlet** and **JSP** documentation for helpful references during development.