# Name: Koushikraj Rajendran

# Project: Online TODO List

# GitHub: <https://github.com/koushikraj23/Online-Todo-List>

## Software Design Pattern:

### [Dependency Injection](https://en.wikipedia.org/wiki/Dependency_injection)(Creational patterns):

* Loosely couple architecture is highly configurable And eliminates or reduces unnecessary dependencies.
* Business logic changes has fewer effects, increasing system stability and reliability
* Unit Testing can be performed using mock objects.
* Increased module reusability.
* Abstraction is achieved easily thus removing complex implementations are
* Decreases coupling between a class and its dependency.

### Waterfall Software Development Model:

As the requirements where defined in the beginning of the stage and based on the complexity of project, waterfall model would be suitable for this project. Other models like Agile, Iterative Model, Incremental model would only be suitable if there were more implementations and time given.

**Day 1**: Requirement gathering and analysing, Analysis of technical aspects with respect to timeline, System Design.

**Day 2**: Development and implementation of system design.

**Day 3**: Testing and Finalizing for deployment.

# Technologies Used:

Maven: A build automation tool to manage a project's build, reporting and documentation. Act as a central repository for various jars and frameworks, thus **reducing the time taken** to search, download and install jars.

Spring Boot: A framework which implements Spring MVC and provide stand-alone, production-grade, readymade spring-based applications. Thus, time and effort taken to individually initialize, implement and integrate various technologies like JPA, Security, Authentication and Server for deployment is removed. Also, the scope of the project was suitable for a microservice architecture, thus spring boot framework would be natural choice.

H2: As the application deals with small amount of data, in-built memory database is efficient for the project. And H2 provides fast and efficient read and write data operations among other inbuild memory database.

Spring Security: Support for both Authentication and Authorization with protection against attacks like session fixation, clickjacking, cross site request forgery, etc. Available for both Servlet API integration and Spring Web MVC.

* Username: admin
  + Password: admin.
* Username: User
  + Password: User

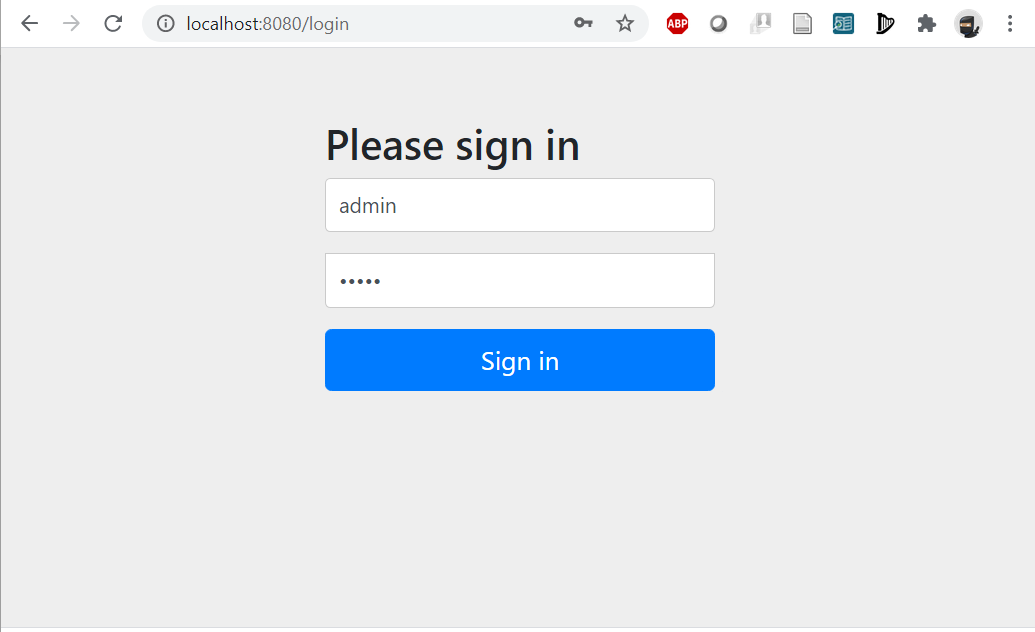
\*\*Above technologies play major part in the role Apart from that many frameworks and jar were also used.

## Future Improvements:

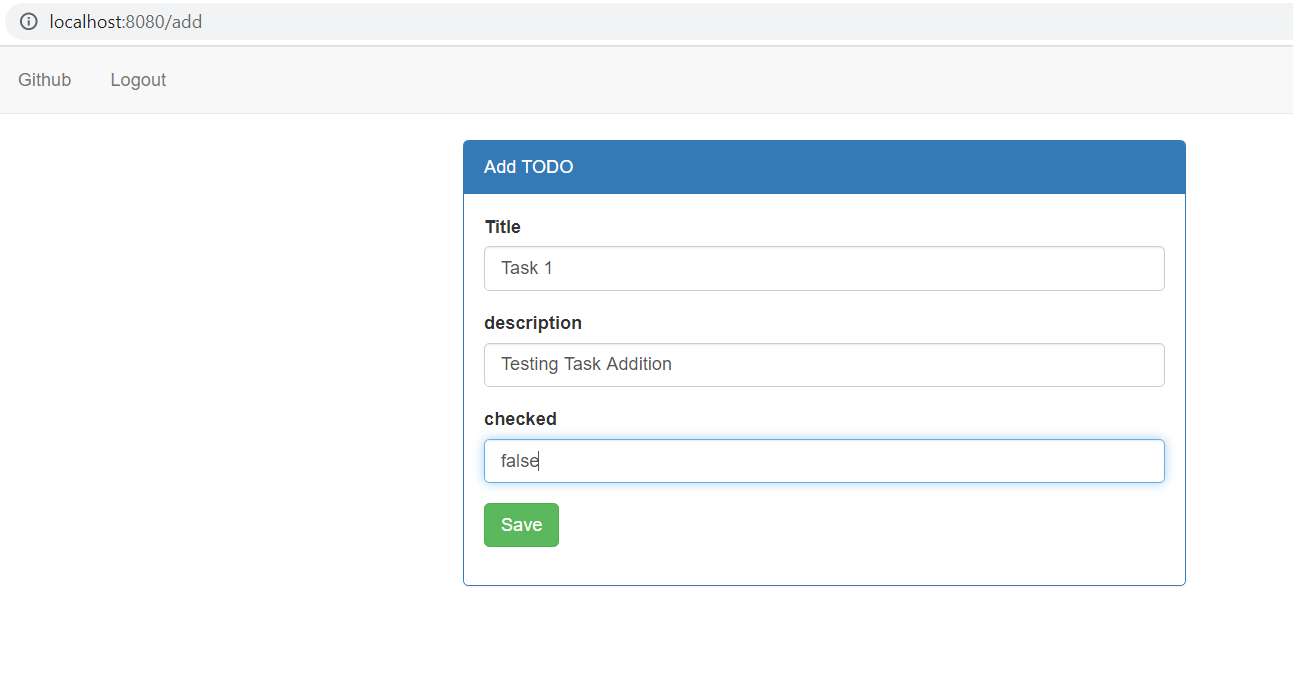
* Maintain User table in DB and maintain many to one relationship with to-do task table.
* Role based Authorization
* Enable Automation Testing.
* Increase Security.
* Connect to Cloud DB.

# Output:

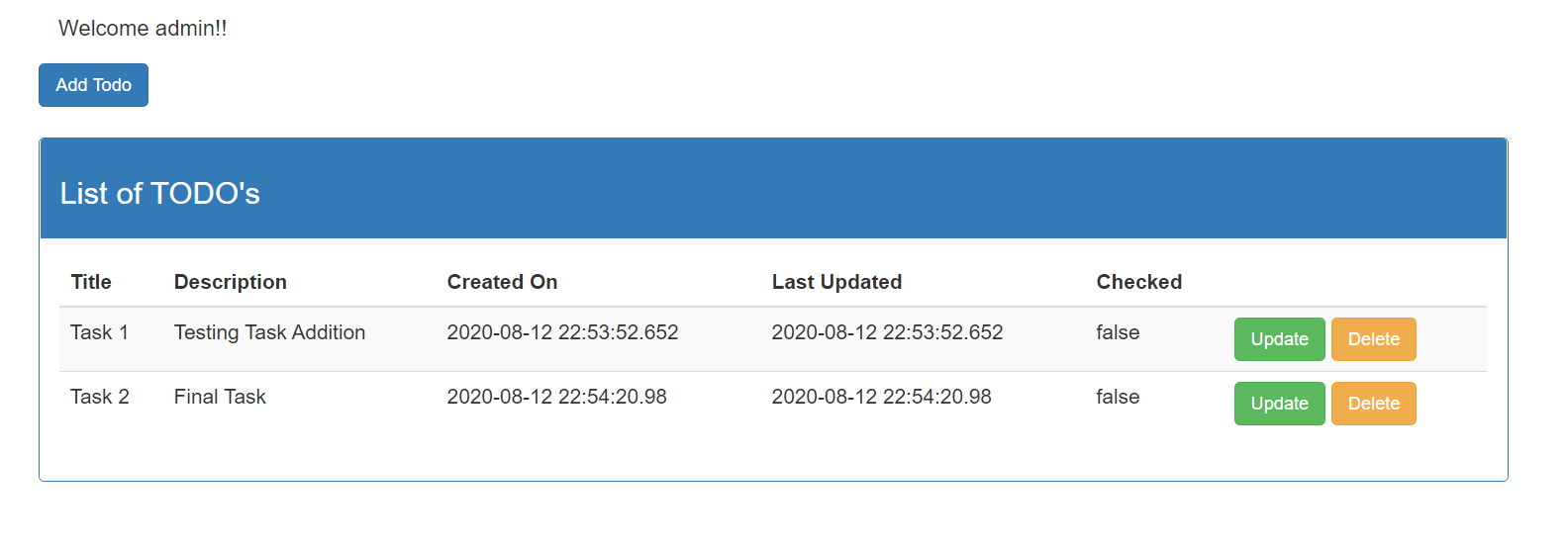
## Login Page



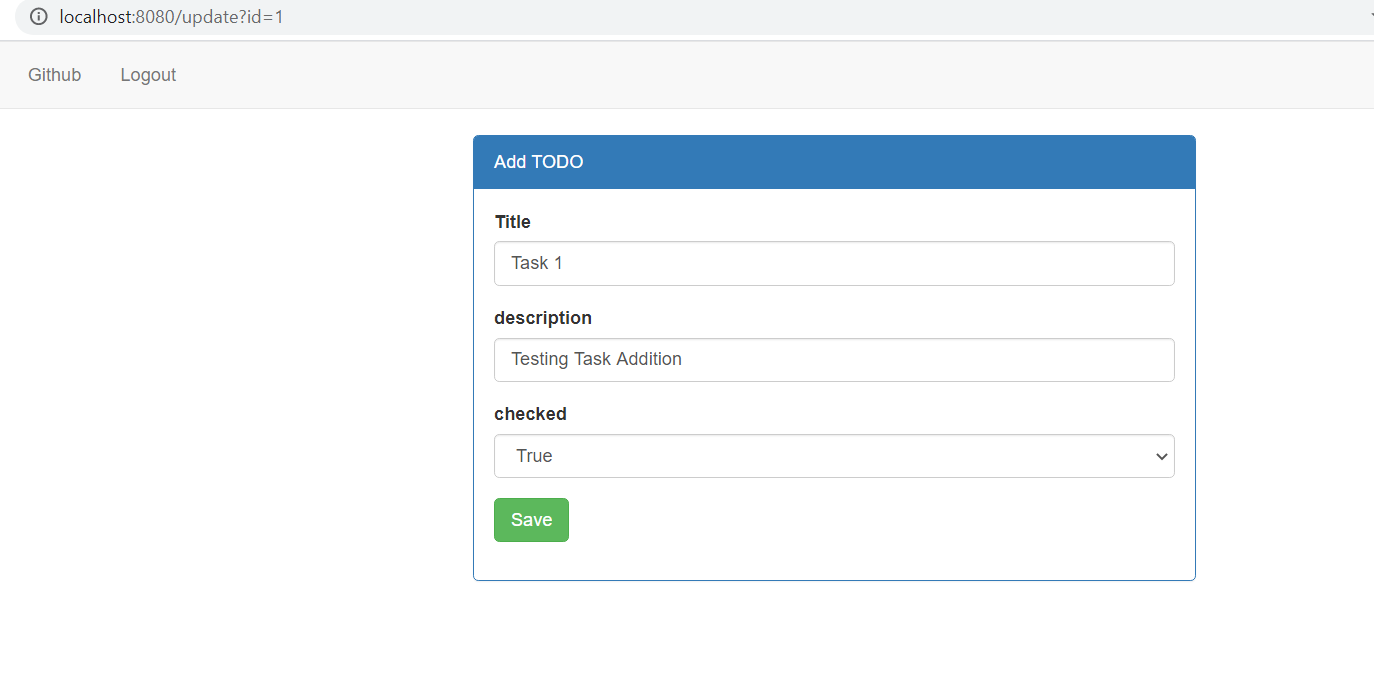
## Task Addition PAge



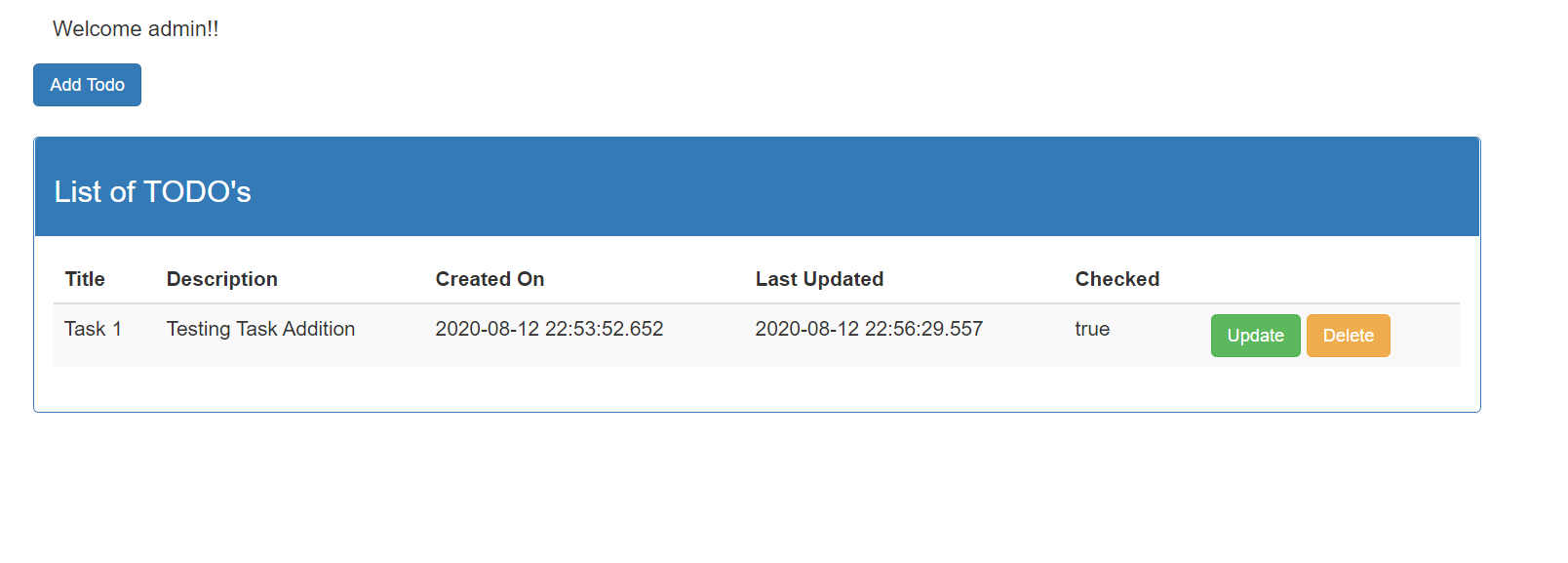
## Task View /Welcome Page



## Task Update Page



## Task Deleted



## Junit Test Cases

