**Todo list web application**

# Approach and Design Pattern

To develop Todo list web application, used **Spring Boot** open source framework. Spring framework provides a comprehensive infrastructure support for developing Java applications.

This helps us to build **microservices** by **dockerization** and orchestrate in the **kubernetes** environment.

# Why Spring Boot?

The problems with spring frameworks are it’s a huge framework, require multiple steps, and also multiple builds and deploy. Whereas spring boot is more of **convention over configuration.** Moreover, its production ready and standalone. Spring boot lets you bootstrap a spring application from the scratch. Spring boot makes it easy to create standalone, production-grade spring-based applications that you can “just run” and quickly get deployed and helps to save enough time for other activities of the project.

# MVC Architecture

Spring MVC is one of the popular and best approaches for developing web based applications. MVC stands for Model, View and Controller and separates all these layers. Basically, it provides loose coupling between the layers. In Spring MVC, the **DispatcherServlet** acts as a front controller. It is responsible to manage the flow of the spring MVC application .I have used **MVC design pattern** to build the web application.

# Spring Data JPA

* To interact with the database, I have used Spring data **JPA**(Java Persistence API).It provides JPA template class to integrate spring application with JPA.
* It gives abstract repositories that are implemented at run time by spring container and perform CRUD operations. As a developer we have to provide the abstract methods in the interface.
* The biggest advantage is Spring data JPA reduces the amount of boiler plate code required to write data access layer.

# Views

For the views I have used Java Server Pages JSP, because performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files. Regular HTML, that is static HTML, does not contain dynamic [information](https://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information). So it does not react to user input and is also not fit for accessing server side resources. JSP contains both static and non-static content. As static part, it contains HTML.

# Database

Used **H2** in memory database. The DB consists of two tables *Users* and *Tasks* table to hold the user and task data respectively. H2 database makes it handy for the developers to test the application.

**What Extra I would have done**

**1. Spring Security**

It is a framework that focuses on providing both authentication and authorization to Java applications. The real power of Spring Security is found in how easily it can be extended to meet custom requirements. Spring security is comprehensive, makes it easy for Servlets API integration and provides protection against attacks. Due to time constraints, I could not implement spring security. Spring provides authentication frameworks one of them is **JWT** based authentication. where a token will be generated at server side and validated for each user.

# 2. JUnit Testing

For unit testing we can use **Mockito**, a mocking framework Java based library that is used for effective unit testing of JAVA applications. Mockito is used to mock interfaces so that a dummy functionality can be added to a mock interface that can be used in unit testing that is used for effective unit testing of **JAVA** applications. I have written few assert tests but due to time constraints I could not write for all the functionalities.

# 3. Session Management

Spring Session provides an API and implementations for managing a user’s session information. while also making it trivial to support clustered sessions without being tied to an application container-specific solution Spring Session decouples the session management logic from the application, making it more tolerant.

Spring Session keeps information in the database, so it’s great to use in a clustered environment with multiple server nodes. Because of this, we don’t need to rely on the sticky session or session replication logic.

As session data is stored in the database, user session data is not lost if the application crashes. When the application started again, it picks up the user session from the database.

**Redis** is one of the widely used tool for for the session management

**4. Logging**

**Logback** logger is used in this application. Logging is the process of writing log messages during the execution of a program to a central place. This logging allows you to report and persist info, debug, traces, error, fatal and warning messages so that the messages can later be retrieved and analysed. The object which performs the logging in applications is typically just called Logger. Loggers will be very useful mainly in production environment.

**Other Constraints to be considered**

Below are the few points that we should keep in mind for developing web applications.

# 1. Security Constraints

A web application must be taken care of vulnerabilities and data breaches. Unfortunately, web applications are the most common ways of exploiting. Following cases must be considered for securing a web application:

**Protect APIs using a Firewall.**

By implementing secure authentications like two-factor authentication.

To the process of securely handling multiple requests to a web-based application from users.

1. To avoid injection attacks in SQL using store procedure.
2. Form Validations for passwords.

# 2. Application Testing

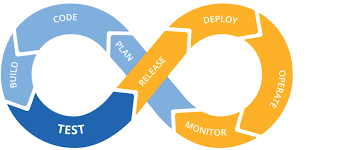
After Unit testing we can go for SIT and UAT testing. Based on System requirements and design document, System Integration Testing (SIT) is conducted to support the interactions after the development of all the modules and integration so that the dependency on the availability of lower-level module functionality is verified.

After SIT testing , User Acceptance Testing (UAT), also known as end-user testing must be used to test the software by the users to decide whether it can be accepted before production build once the functional, system and regression testing are completed. Finally applications will be moved to production.

**3. DevOps – CI/CD**

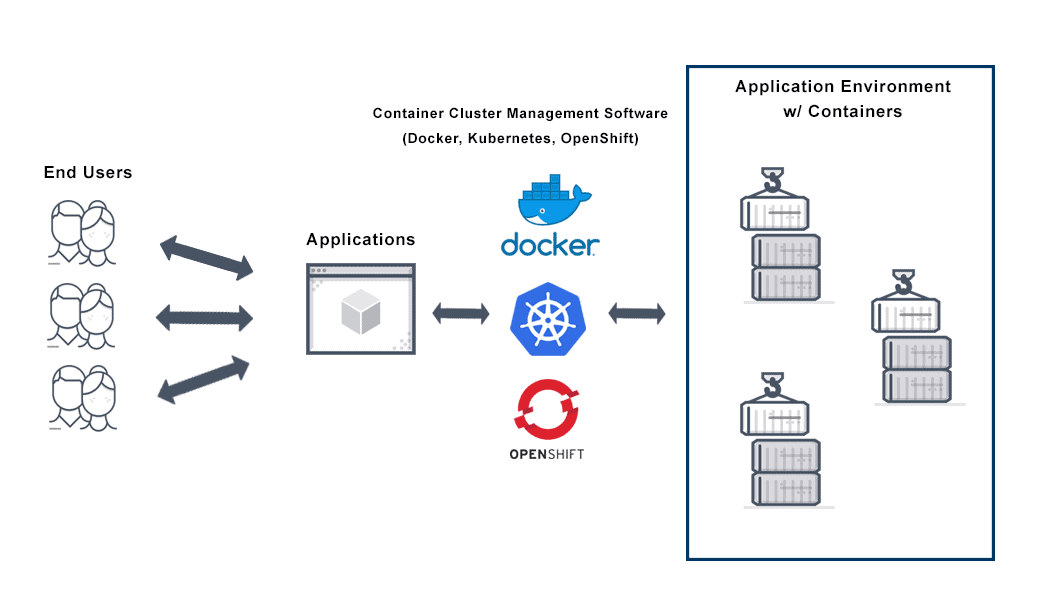
CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment

**Jenkins** is one of the widely used open source tool to build CI/CD pipelines



**4. Containerization deployment**

Container deployment is the act of pushing (or deploying) containers to their target environment, such as a cloud or on-premises server.



**5. Centralized Logging**

**Elasticsearch, fluentd/fluentbit, Logstash and Kibana** can be used to monitor the application logs from the Kubernetes, cloud or on premises environments.

**6. Monitoring**

**Prometheus** and **Grafana** can used to monitor the infra and container deployments in the Kubernetes environments

**Github Link**: [https://github.com/sudhansagar/eytodolist](https://github.com/rakesh4u/TodoSpringBootApp)

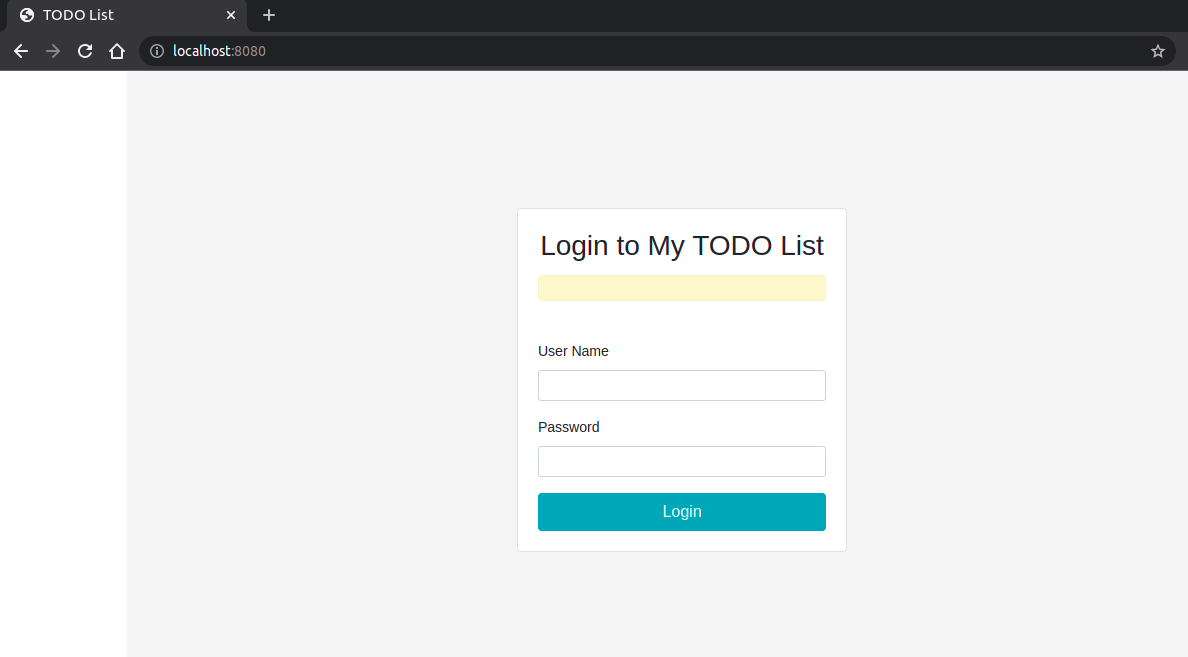
**How to Run the application:**

1. java -jar target/todolist-0.0.1-SNAPSHOT.war

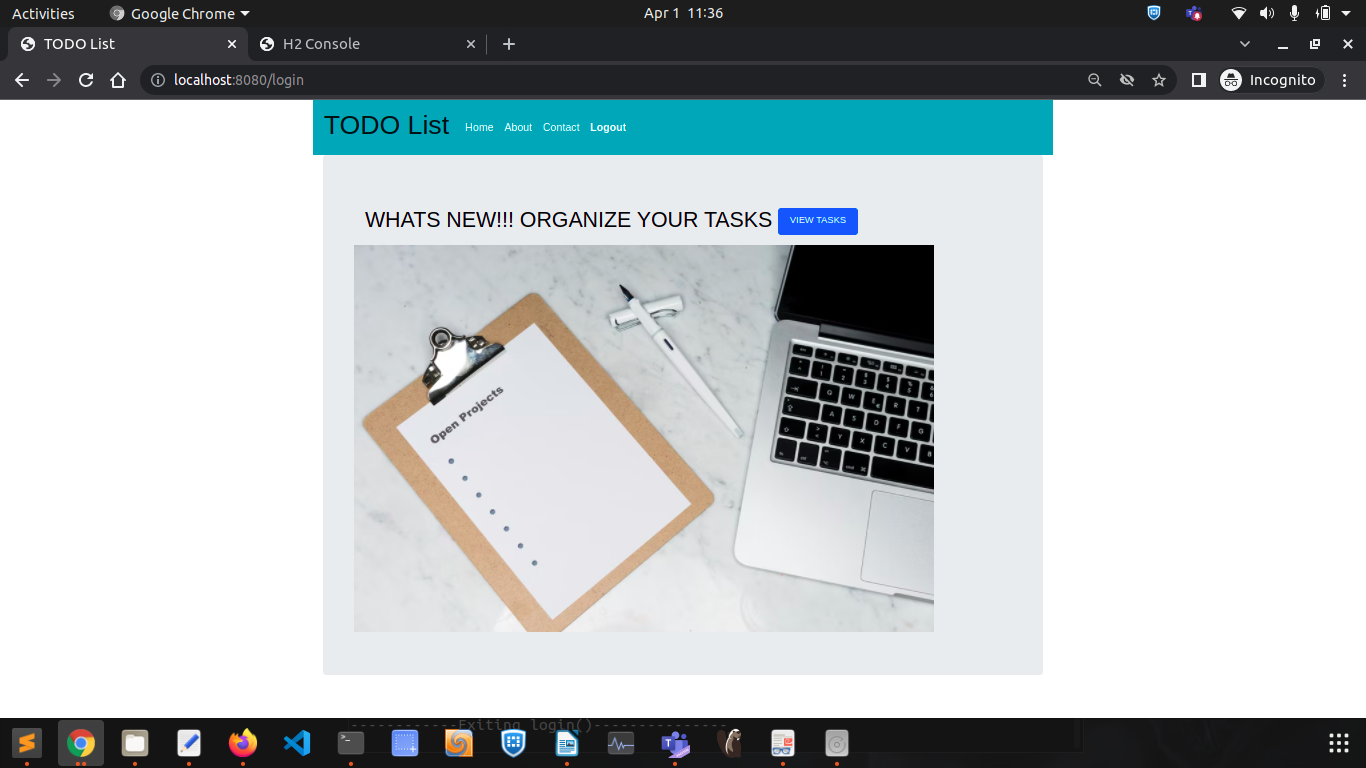
2. Launch the URL: <http://localhost:8080/>

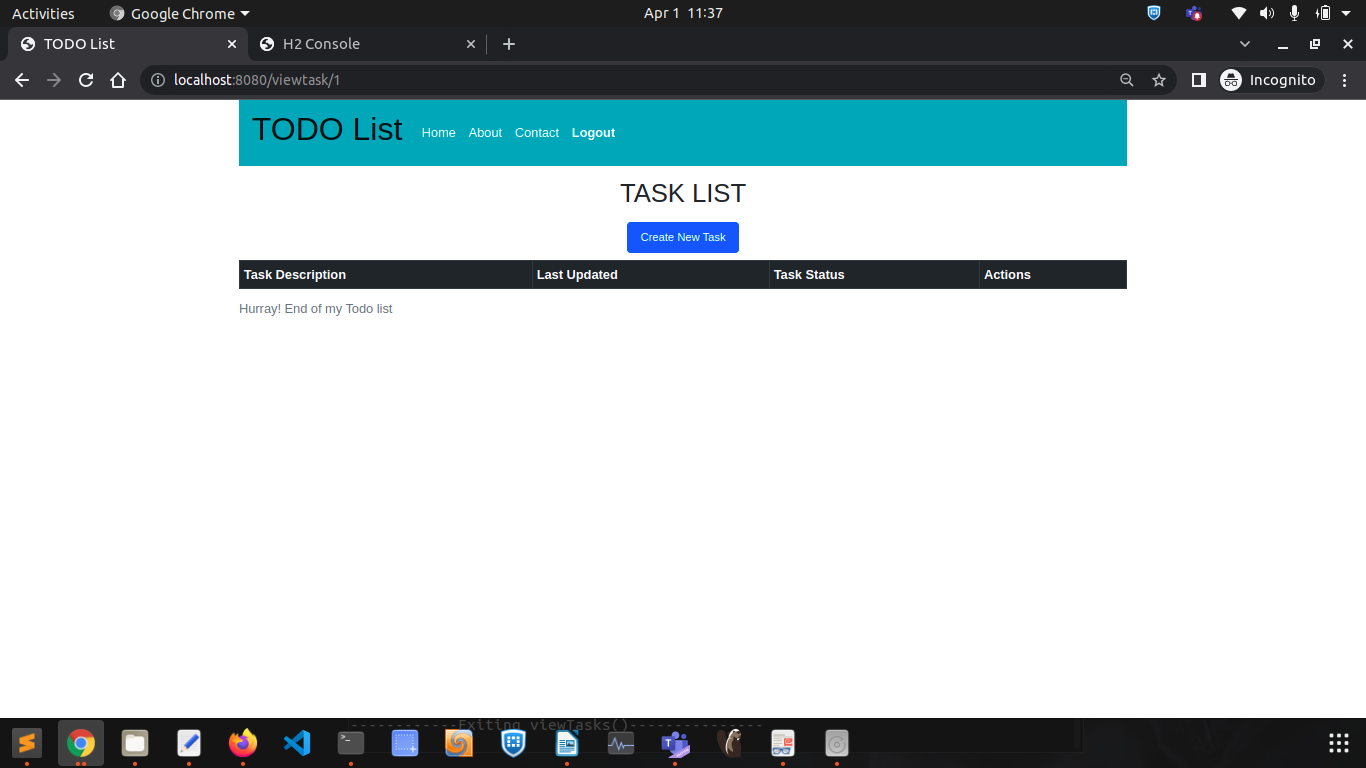
3. Follow the below workflow from the screenshots

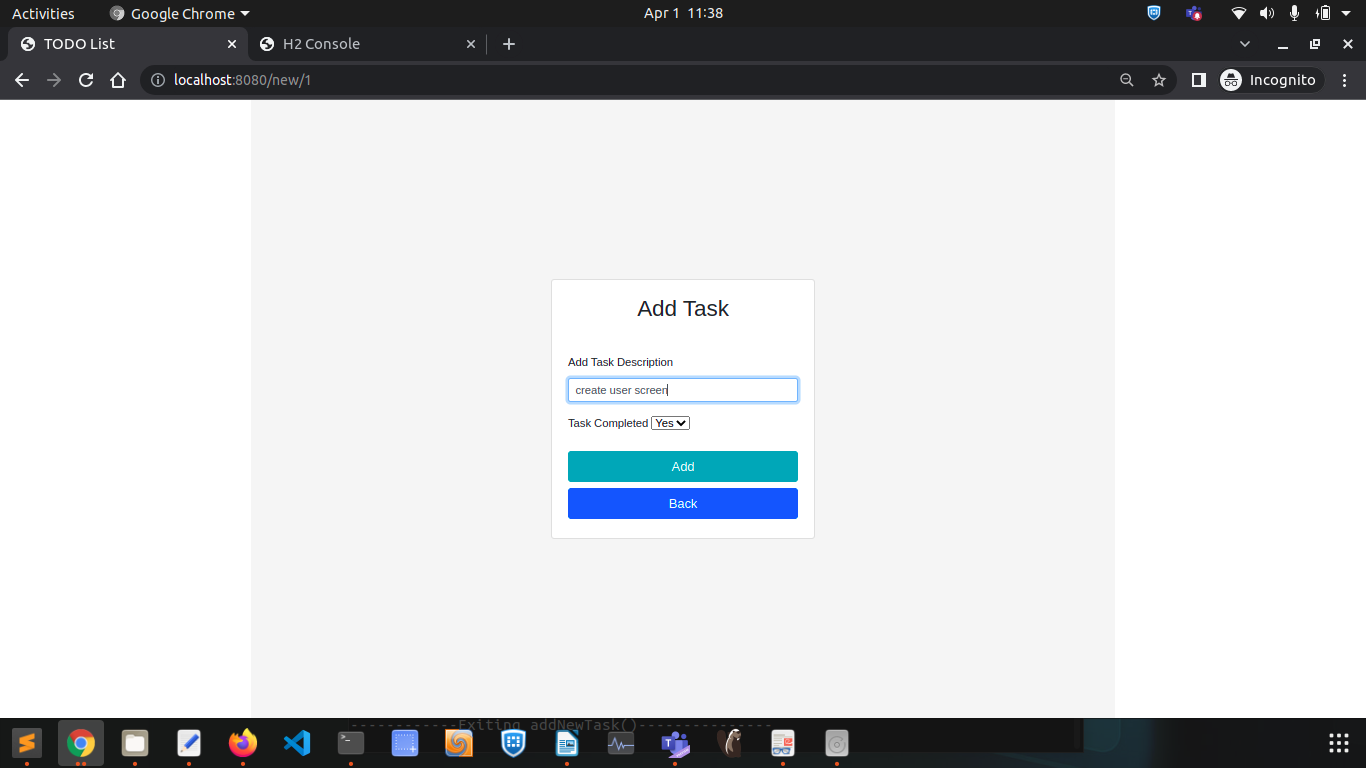
**Screen Shots of Todo web Application**

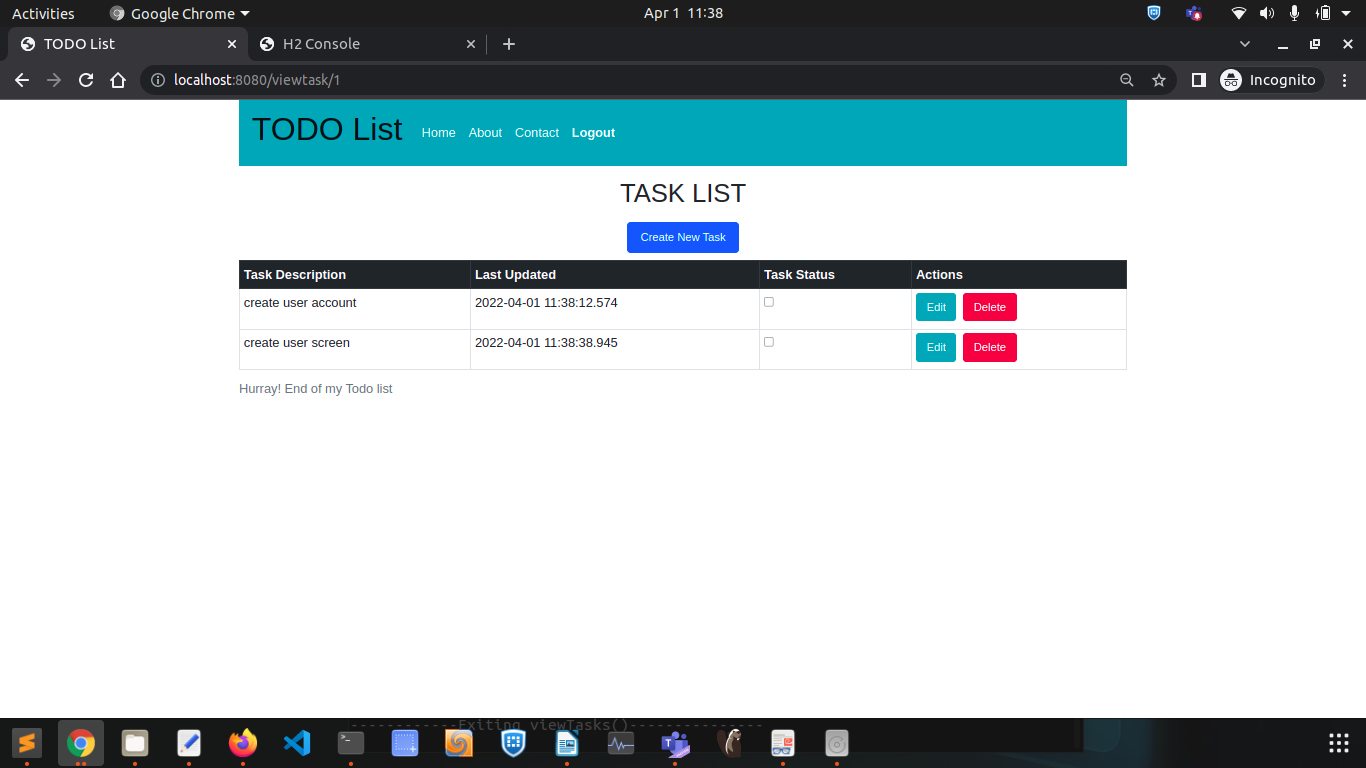


***fig : Login Page***

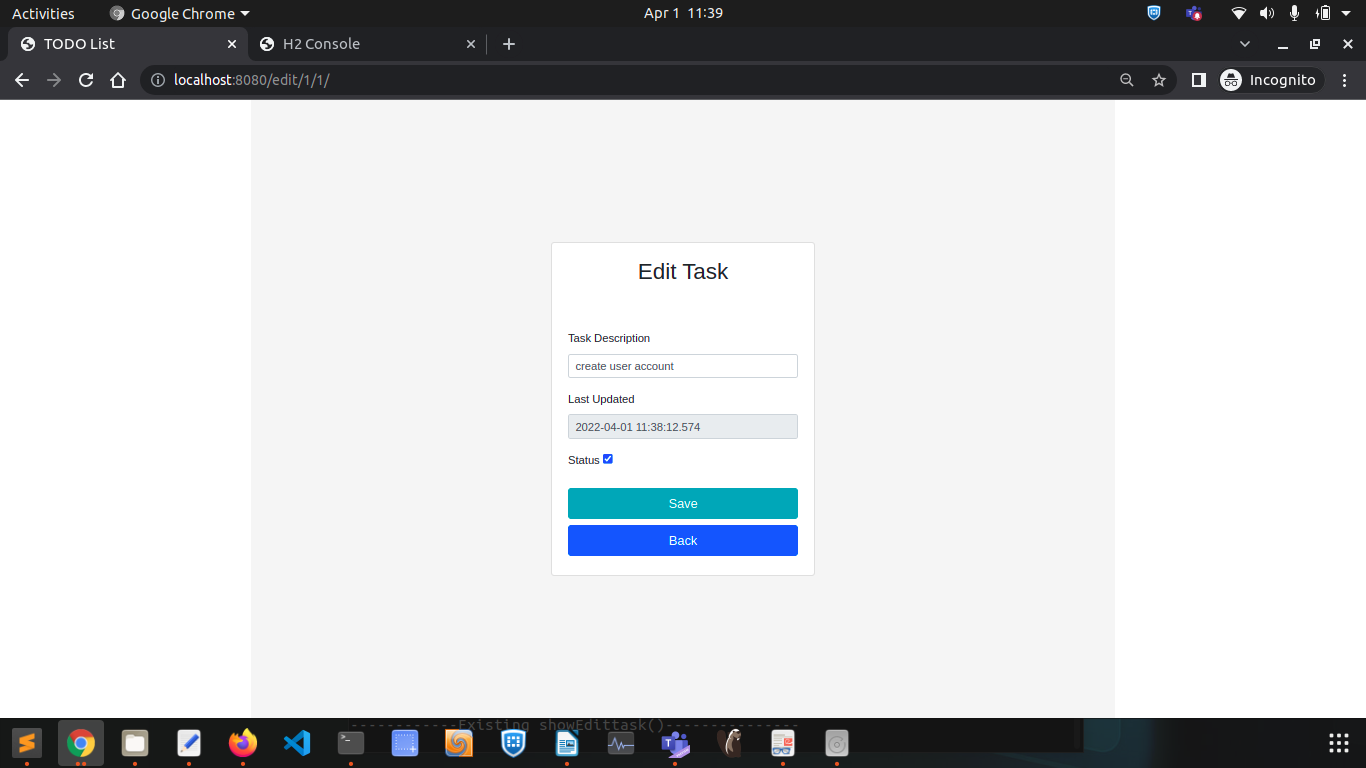
***fig:HomePage***

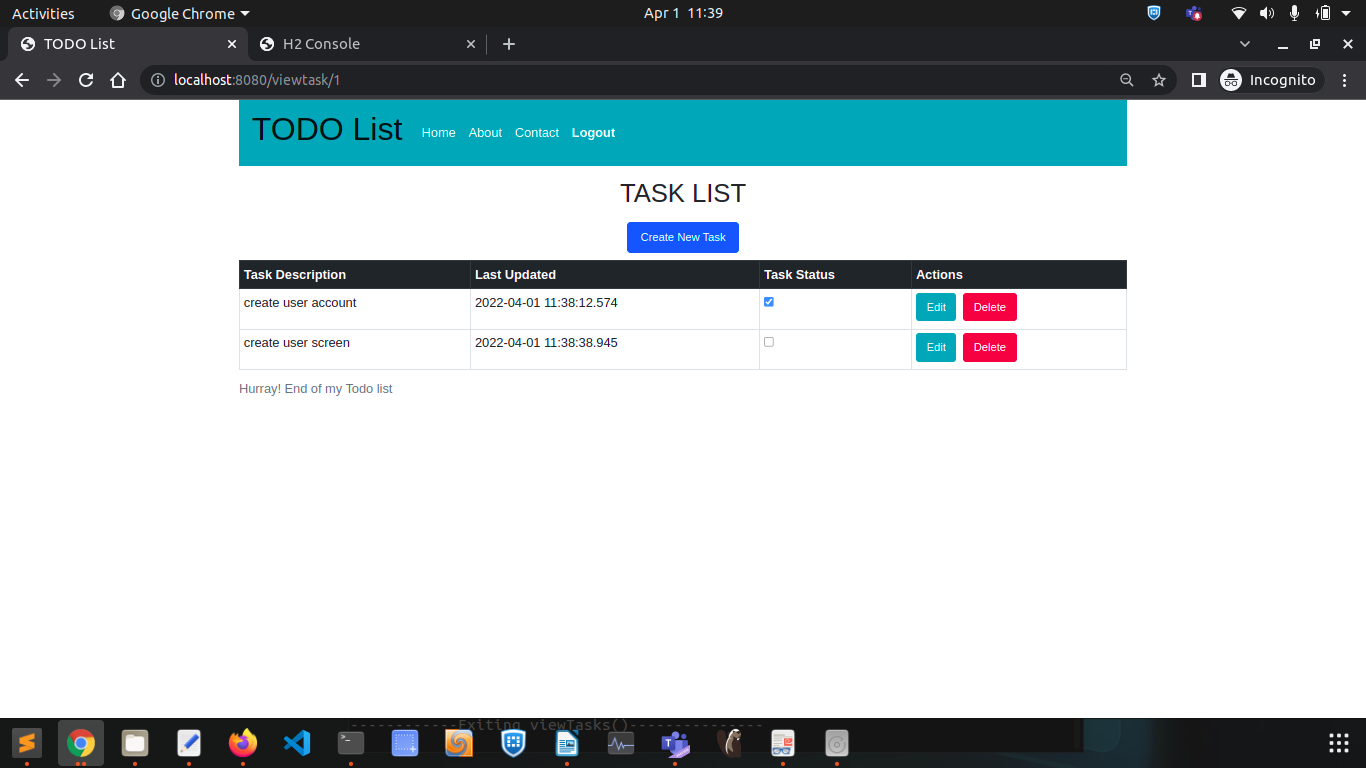
***fig : ViewTaskPage***

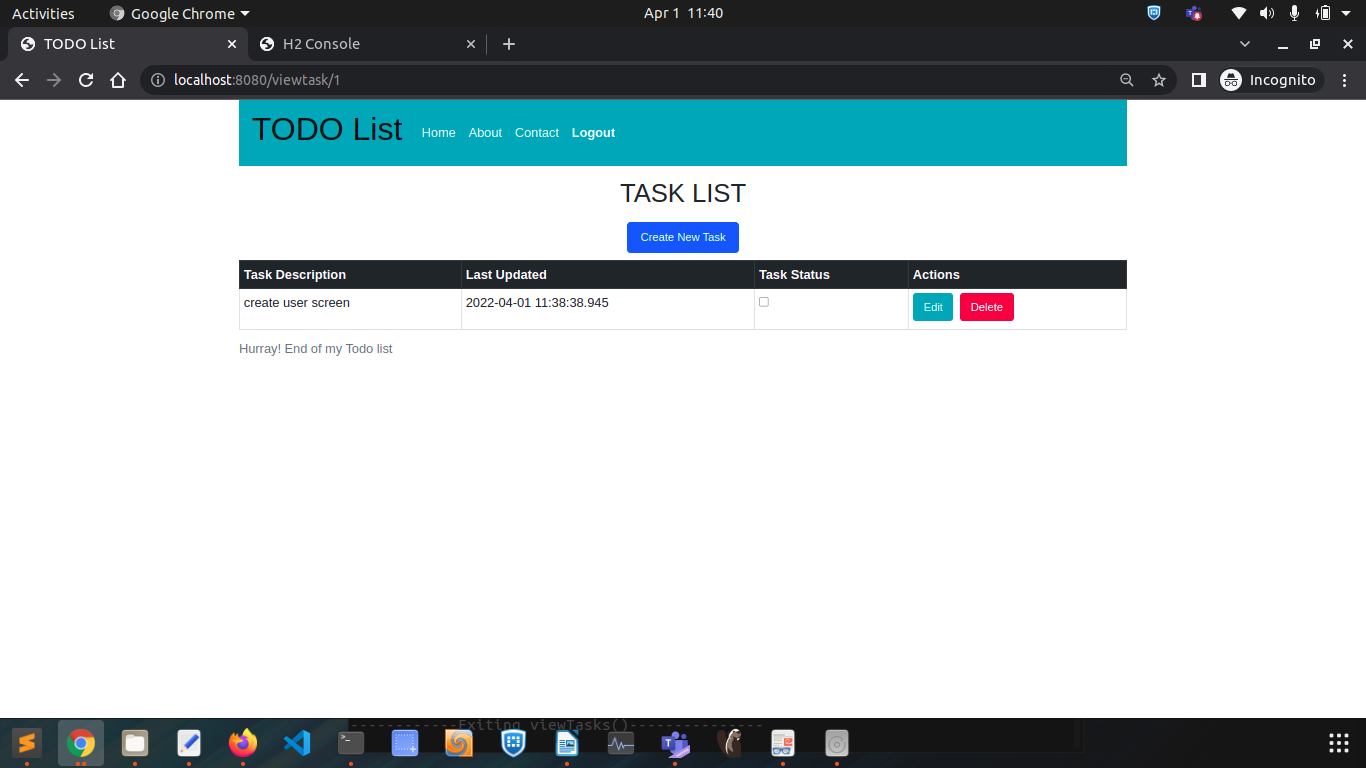
***fig:AddTaskPage***



***figNewlyAddedTask***

***fig : EditTask***

***fig : EditedTask***

***fig :DeletedTask***

**Troubleshooting:**

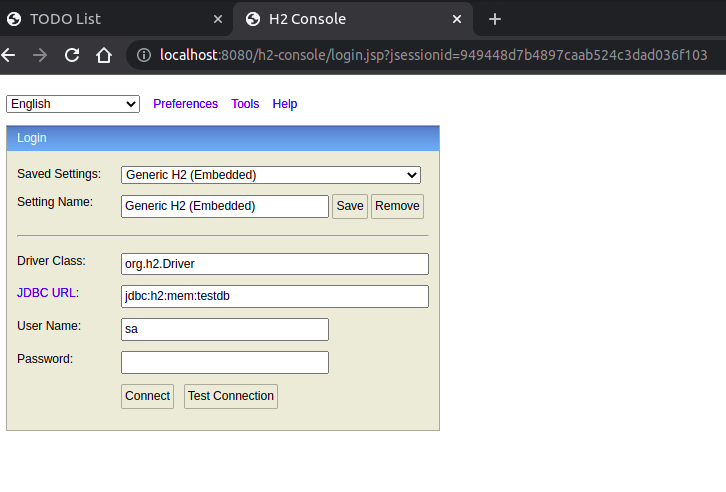
1. In case war file doesn’t work you can run the application from command line

java -jar target/TodoManager-0.0.1-SNAPSHOT.war

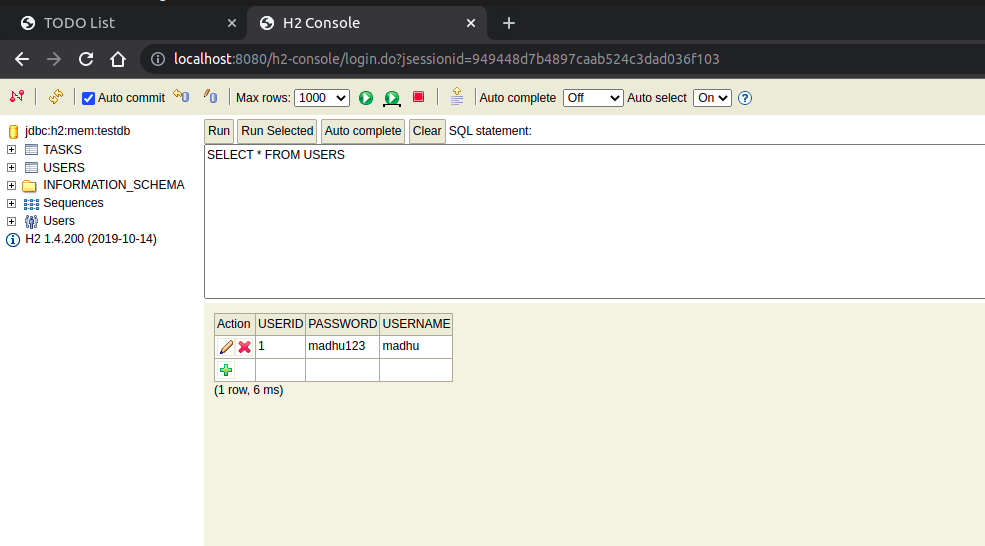
2. After on launch of the URL, we have already user inserts available in the Users table of In-memory database

In case if you are not able to login to the application, please make an entry in the database following the steps :

1. Launch the URL : [http://localhost:8080/h2-console](http://localhodt:8080/h2-console) and Connect to the database



2. Select User as shown below and insert a row with details as below shown screenshot



3. Try to login to the application now with the new user detail, you should be able to navigate to the home page of the application and you can continue.