Read me file for Todo Manager application

# Software Requirements:

1. It should be a responsive web page.
2. It should allow a customer to list the active items in the todo list.
3. Allow adding new simple one line item
4. Allow deleting the items
5. The delete option should have a web based confirmation dialog asking the user to confirm they really wish to delete the selected item.
6. Summary of the todo item should not be more than 100 characters.
7. Description of the todo item should not be more than 500 characters.
8. Due date time for the todo item should be in the format 'dd-MM-yyyy hh:mm:ss' where
   * dd: day
   * MM: month
   * yyyy: year
   * hh: hour
   * mm: minutes
   * ss: seconds

Any technology can be used as per discussion.

# Application deployment instruction:

Application requires a RDBMS, a Java EE application server & web browser. It has been tested with the following software:

* MySQL Cmmunity edition 5.6.19
* Glassfish Open Source edition 4.0
* Mozilla Firefox 44.0.2, Internet Explorer 11

1. In MySQL command prompt, create database by running the following command – create database todoDB.
2. Run the sql script sql\_script.sql to create the tables and to load default data. The script can be run using the command – source <path to script>/sql\_script.sql. SQL script is present in TodoEJB\ejbModule folder.
3. Start the Glassfish server by running 'start-domain' command in the asadmin prompt. 'asadmin' should be run by running 'asadmin' command.
4. In the Glassfish admin console, create a JDBC connection pool for the todoDB database & a XA-compliant driver. Then create a JDBC resource with JNDI name 'jdbc/TodoDB', referring to the JDBC connection pool created earlier.
5. Install the application from the asadmin command prompt using the command – deploy <path to EAR file>/todo.ear.
6. If the application is installed, open the home page by using URL – http://<host>.<port>/TodoWeb-JSP/index.jsp.

# Architecture/Design:

Following technologies are used: JPA, JSP, Servlet, EJB, JNDI, HTML, CSS, Javascript.

Various layers in the application are as follows:

1. Presentation layer comprising of JSPs & TodoServlet.
2. Persistence layer comprising of TodoItemDAO class, its interfaces, & TodoItem & Status entity classes.

Business logic layer is not present. It can be implemented using Stateless Session Beans. Unit tests can be written using Arquillian framework for the business logic layer.

Validation regarding size of summary & description is done by using Bean Validation annotation in the Entity class, and also by setting maximum size in the HTML code.

# Package contents:

The contents of the package are as follows:

1. todo.ear file.
2. TodoProject-JSP.zip file containing the source code. This file can be imported in Eclipse as Existing Projects.
3. This file.