**Deloitte Read Me for To Do List**

1. ***Approach Used***:

* **Spring Boot** framework is used for backend as it helps in developing the web applications in the fast and easy manner. The advantage is the auto-configuration which is really helpful in developing application. It also provides inbuilt tomcat for running the application.
* **Angular** framework is used for frontend development as it helps in creating light-weight single page applications.
* **H2** database is used as the in-memory database.
* **Session based authentication** is used for security. In this approach user credentials (username and password) are authenticated during login and a UUID session is generated on the server side and stored in the database. Every time, user creates a new task, edit an existing task or delete a task, the session is checked for authorization.
* **Cascade delete** is implemented so that if a user decides to delete his account. Then his relevant tasks will also be deleted from the task table.
* **Spring data JPA is** used for CRUD operation with h2 database.
* **Custom** Exceptions are implemented in exceptions package and status code is implemented using Enum class in utils package.
* The packages created in the spring boot are **main, controller, models, exceptions, services, utils, repository, mail, test**.
* PostMan used for testing the endpoints locally.
* **Login component, Register, to-do-task-list, forgot passwords** are the components created in Angular. The version used is Angular 11.
* **Maven** used for managing dependencies.

Below are the endpoints

User component endpoints:-

<http://localhost:8080/Deloitte/createUser> (creating a new user)

<http://localhost:8080/Deloitte/loginUser> (login endpoint for existing user)

<http://localhost:8080/Deloitte/forgotPassword> (new password setting for existing user)

<http://localhost:8080/Deloitte/deleteUserData> (user can delete his/her account)

Task component endpoints:-

<http://localhost:8080/Deloitte/createTask> (user can create a new task after login)

<http://localhost:8080/Deloitte/getUserAllTasks> (user sees all tasks on dashboard after login)

<http://localhost:8080/Deloitte/editTask> (user can change details of an existing task)

<http://localhost:8080/Deloitte/deleteTask> (user can delete a task)

1. ***Architecture pattern*** used is MVC which brings modularity in the code. View being the frontend in Angular. Controller package used for handling incoming request containing all the endpoints and mapping it to models in backend. Backend also contains service layer for business logics, repository layer containing spring data JPA for CRUD operation and a H2 in memory database.
2. ***If I could have been given more time***:

* I implemented the SMTP in the project but I was getting SSL handshake Exception. I tried various workarounds such as adding trust properties in application.properties it was not working. I removed the SMTP call from the service and starting developing frontend due to shortage of time. I could have checked more into this exception if I had more time.
* I could have completed the unit testing for all the controller endpoints, service logic methods, models etc.
* I could have implemented swagger UI for creating endpoint document.
* I could have also implemented logging in backend.
* I could have worked more in frontend such as improving design.

1. ***Anything else***: I could have worked on completing the J-unit testing first before SMTP implementation. As due to time-constraint I couldn’t do unit testing properly. I also faced issue in persistence storage of in-memory h2 database as data was getting lost after application restart but I was able to resolve it with the help of the internet.