JAVA BACKEND LLM ASSESMENT

SUCCESSFUL REFERRAL

Problem Description

 a. Create an API for user signup with referral code tracking, where a referral is considered successful only after the user completes their profile.
 For instance, if user A signs up using referral code of user-B then successful referral is considered only after user-A completes the profile.

Expected Output -

- 1. User Signup Endpoint
 - Allow signup with/without referral code
 - Generate unique referral code for each user
 - Validate referral code if provided
- 2. Referral Tracking
 - Track referral status
 - Link referrer and referred user
 - Mark referral as complete upon profile completion
- 3. API Endpoints
 - Signup API
 - Profile Completion API
 - Get Referrals API
 - Referral Report API (Bonus)

Solution:

- I am using the MVC (Model View Controller) Code Structure, for managing the Code in proper and adequate manner.
- This problem code consists of different packages
 - 1. entity
 - 2. db (Database)
 - 3. DAO (Data Access Object)
 - 4. Servlets
- Their exists the near around the 7-8 API endpoints for the given problem

Details:

- entity package:
 - User: This consists of different attributes such as userID, userName, userEmail, phone, userReferralCode, loginReferralCode, password, referralStatus.
 - 2. Referral package: This consists of different attributes such as referralID, referralCode, userID.

NOTE: Relation between the user and referral (referralStatus) is has-a relation.

- db (Database):
 - 1. DRIVER FOR THE DATABASE USING:
 - i. MySQL used in this Project
 - ii. DRIVER STRING FOR MySQL: com.mysql.cj.jdbc.Driver
 - 2. DATABASE URI STRING
 - i. jdbc:mysql://localhost:3306/DATABASE_NAME_FOR_PROJECT
 ", "YOUR_ROOT", "YOUR_PASSWORD"
 - 3. USERNAME [only if Set while creation of the System-DB]
 - 4. PASSWORD [SECURITY]
- <u>DAO (Data Access Object)</u>: The <u>Data Access Object</u> Pattern, sometimes known as the <u>DAO</u> pattern, is a method of structuring code to manage the interaction between a computer and a database

In this Project, I Have used two DAO model:

- UserDAO
- ReferralDAO

NOTE: ReferralStatus is Enum-type to update the CURRENT-REFERRAL STATUS.

KEY: - Interaction Points

♣ UserDAO

- a. userRegister: Used to register a User
- b. loginUser: used to login user once register
- c. **completeProfile**: Used to fill in the Data required to update the Profile and helps to use the referrals
- d. changePassword: used to change the password

- e. **updateReferralStatus**: Applied once the Profile Details is Completely filled and User have register using ReferralCode
- f. **getUser**: fetching data from the server based on the **useremail** and **password**
- g. **getAllUser**: fetches all the user from the server register
 - Endpoint for the ADMIN

ReferralDAO

- a. addReferral: Used to add Referrals to Database once user is Register
- b. **matchReferralUsed**: Used for validating the ReferralCode that a newUser is using while register themselves
 - a. Here we get the userID from the **Server (form Referral Table)**
 - validate (referralCode, userID): used to validate the referralCode across the User Table form the server -- THIS IS INTERNAL IMPLEMENTATION
- c. changeReferralCode: Used to customize the userReferralCode if needed
- d. **getAllReferral**: fetches all the Referrals from the Server
- e. getAllReferral: fetches all the Referrals based on the userID
- f. **getReferrals**: fetches all the Referrals **based on the ReferralCode**
- g. **getReferralData**: fetches all the Referrals form the **Server Based on Unique userID**
- **ReferralStatus:** It is the package of the Enum-types for changing the referralStatus of the user-referral relation

This consists of different values:

- a. **PENDING** -> User just registered using the referral code but has not completed their profile details.
 By default, the referral code status is **PENDING**.
- b. **CLAIMED** -> User registered and completed their profile details and claimed the referralCode
- c. **NOT_USED** -> User registered without any referral Code

• **Servlet:** Servlet is a technology which is used to create a web application. It is an API that provides many interfaces and classes including documentation.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

NOTE: Endpoints

♣ UserAPI

- i. **signup**
 - a. https://localhost:8080/simplifyMoneyAssignment/signup: used for making the user register with or without referral Code
 - b. Also used for the generating the Referral Code across the User
 - c. Each ReferralCode generated is Uniquely
- ii. **signin**
 - a. https://localhost:8080/simplifyMoneyAssignment/signing is used for making user login to the site or page
- iii. completeProfile
 - a. https://localhost:8080/simplifyMoneyAssignment/com pleteProfile: used to complete the profile details
 - b. Responsible for changing the **ReferralStatus**, if used while Registering (signup)
- **♣ ReferralAPI:** This extraction is only possible if the user is login
 - i. getAllReferral
 - a. https://localhost:8080/simplifyMoneyAssignment/adm in/getAllReferrals: Extracts all the Referrals from the Server
 - b. This is the ADMIN-API
 - c. CUSTOMIZE
 - > Email: admin@gmail.com and Password: password
 - d. **HARDCODED**, Present Scenarios

ii. getReferralsUsingReferralCode

a. https://localhost:8080/simplifyMoneyAssignment/find
ReferralCode Extracts all the Referrals from the Server based on the ReferralCode

iii. getReferralUsingUser:

a. https://localhost:8080/simplifyMoneyAssignment/getR
 eferral: Extracts all the Referrals from the Server based on the userID from current login user (Specifically from the HTTP-HEADER)

KEY

→ COMMA-SEPERATED VALUES (CSV): A Comma-Separated Values (CSV) file is just a normal plain-text file, store data in column by column, and split it by a separator.

CHALLENGES:

- getAllReferral: Based on the unique userID and
 - https://localhost:8080/simplifyMoneyAssignment/admin/referralsData: Creates a .csv file for all the Register user with or without referral and the status of the referrals used
 - o Referral Status is based on the Profile Details Completion

Brownie Point:

- https://localhost:8080/simplifyMoneyAssignment/brownie/reportCSV:
 Creates a .csv file for all the Register user with or without referral and the status of the referrals used
 - o Referral Status is based on Profile **Details Completion**