**Reff\_System**

**Major Module:**

1. Login & Registration
   1. Authentication
   2. Token Generation and Tracking
   3. Appy Hashing to Password
2. Referral Profile
   1. Unique Link Generation
   2. Tack the User is Register using Referral Link or Not
3. Wallet Update
4. Withdraw Request
5. User Profile
   1. Update Profile
   2. Change Password
6. Admin Login
7. Admin Panel
   1. Withdraw Request
   2. Modify Request
   3. Approved / Reject Request

**Tech Stack:**

HTML, CSS, JavaScript, Bootstrap, MySQL, Nodejs

**User Flow**

1. User Registration and Login Flow:
   1. Accesses the registration page.
   2. Enters name, email, password, and confirms password.
   3. Submits the registration form.
   4. Upon successful registration, user is directed to the login page.
   5. User logs in with email and password.
   6. If credentials are valid, user is authenticated and directed to the dashboard.
2. Referral Flow:
   1. User shares their unique referral link with others.
   2. New users click on the referral link and register.
   3. Referrer and new user each receive a $1 reward upon successful registration.
3. Dashboard Flow:
   1. Upon logging in, user is directed to the dashboard.
   2. Dashboard displays user profile information and wallet balance.
   3. User can navigate to the profile page to update information or change password.
4. Withdrawal Request Flow:
   1. User navigates to the withdrawal page from the dashboard.
   2. Enters withdrawal amount, account number, and IFSC code.
   3. Submits withdrawal request.
   4. System validates the request against wallet balance and threshold amount.
   5. If valid, request is added to the withdrawal list with status "pending".
5. Admin Panel Flow:
   1. Admin logs in to the admin panel.
   2. Admin views the list of withdrawal requests with status filters.
   3. Admin has the option to modify withdrawal requests if needed.
   4. Admin can approve or reject withdrawal requests, updating their status accordingly.

**Module 1: Login and Registration Module**

In the registration process, users are prompted to provide their name, email, password, and confirm password. This information is then securely stored in the user table within the "reff\_system" database. The password provided by the user is securely hashed using the bcrypt Node.js module before being stored in the user table within the "reff\_system" database.

In the login process, users input their email and password. The system checks whether the email exists in the database. If the email is found, the system verifies the password by comparing it with the hashed password stored in the database. Upon successful authentication, the user is logged in.

To track the user session, JSON Web Tokens (JWT) are utilized. These tokens are generated using a secret key. Once a user logs in, a JWT token is issued, containing relevant user information. This token is then securely transmitted to the client.

Additionally, an "isLoggedIn" middleware function is employed post-login to monitor user activity using the token. This middleware function validates the token upon each subsequent request, ensuring that only authenticated users can access protected routes.

**Module 2: Referral Module**

In the registration process, users are prompted

In the referral module, a unique six-digit key is generated for each new user during the registration process using nanoid. This key is then stored in the user database.

Subsequently, a referral link is created using the generated key. For example, if the unique key is "bsgHrA," the referral link would be structured as follows: http://localhost:3000/register?ref=bsgHrA. When accessed, this referral link directs new users to the registration page.

**Module 3: Wallet Update**

By utilizing unique keys and referral links, the system facilitates seamless tracking and attribution of referrals, enhancing user engagement and incentivizing referrals within the platform.

Upon decoding the referral URL, the system extracts the referral ID embedded within the URL. This referral ID is then checked against the database to verify its validity. If a match is found, indicating that the user arrived via a referral link, both the new user and the referrer are rewarded with $1 each.

It's important to note that only users who register through a referral link are eligible for the reward. Direct registrations do not qualify for any rewards. All wallet balances, including the rewards, are stored within the user table in the database. This ensures easy access and management of user funds. By implementing this referral reward system, the platform incentivizes user referrals, fostering user engagement and growth while providing tangible benefits to both new users and referrers.

**Module 4: Withdraw Request**

Withdrawal Request Form: Users access a dedicated page containing a form to submit withdrawal requests. The form typically includes fields for the withdrawal amount, account number, and IFSC code.

Validation: Upon submission, the system validates the withdrawal request. Firstly, it checks if the requested withdrawal amount is within the user's available wallet balance. Additionally, the system may impose a threshold amount (e.g., $5 or $10) to ensure withdrawal requests are of significant value.

Withdrawal Request List: Users can view a list of their withdrawal requests, including details such as the withdrawal amount, account number, IFSC code, and the status of the request (pending, approved, or rejected). This provides users with transparency and visibility into the status of their withdrawal requests.

**Module 5: User Profile**

This module allows users to manage their profile information and change their password securely. The functionality includes:

Profile Data Update: Users can update their profile information, such as their name and email address. An interface is provided where users can input new information, and upon submission, the system updates their profile accordingly.

Password Change: Users have the option to change their password. To ensure security, the system requires users to provide their old password along with the new password. The system verifies the old password before allowing the change to occur.

Middleware for Authentication: To access the profile management and password change functionalities, users must be logged in. A middleware function, "isLoggedIn," is implemented to verify the user's authentication status before allowing access to these features. This enhances security by ensuring that only authenticated users can modify their profile data and passwords.

**Module 6: Admin Login**

Facilitates secure access for administrators to the system's administrative dashboard.

**Module 7: Admin Panel**

Offers comprehensive visibility of all withdrawal requests, equipped with status filtering options for efficient management. Empowers administrators with the capability to modify withdrawal requests, ensuring flexibility in addressing any necessary changes or updates.

Facilitates the seamless approval or rejection of withdrawal requests, enabling administrators to efficiently manage and control the financial transactions within the system.