**Task-01**

Create Java Cmd application where users can create an account by entering details such as userID, full name, email, password, confirm password, and date of birth. The system will automatically assign a default role of "User" during account creation and record the account creation time without any user input. These details, along with the role and account creation time, will be stored in the Oracle database but will not be displayed to the user.

Real-time validation is a key feature of the application. As the user enters their userID and email, the system will immediately check the database to ensure both are unique. If a duplicate is found, an error message will be displayed, prompting the user to correct the information before proceeding. This ensures that each user account has a unique userID and email.

Additionally, the application will provide CRUD functionality:

* Create new user records with unique userID and email.
* View all stored user records, including the role and account creation time.
* Update user details based on the userID.
* Delete user records from the database using the userID.

This application ensures a smooth user experience with real-time validation and automated role assignment, making user account management secure and efficient.

**Task-02**

In this Java Swing GUI application, users will perform tasks similar to the command-line version but with a graphical interface. Before starting, a default entry will be added to the database with the role of "Super Admin." When the application starts, users will be presented with two main options:

1. **Create Account**: This option allows users to create an account. Users will input details such as userID, email, full name, password, confirm password, and date of birth. During account creation, the system will check in real-time whether the userID and email are unique. If either value already exists, an error message will prompt the user to enter a different userID or email before submitting the form. As before, the default role of "User" will be automatically assigned, and the account creation time will be recorded in the database but not displayed.
2. **Manage Users**: This option will redirect users to a login page. Users will log in using their email and password. Only users with the role of **Super Admin** or **Admin** can access the user management features. After logging in, the menus will differ based on the user's role:

* **Super Admin**: Will see two options—**Manage Users** and **Manage Admins**. The Super Admin can create new users, assign roles (Admin or User), delete users, and edit both regular users and Admins.
* **Admin**: Will see only the **Manage Users** option. Admins can manage regular users by editing or deleting them but cannot manage Admins. Additionally, Admins cannot assign the Admin role when creating new accounts, limiting them to assigning the "User" role.

This structure provides role-based access control, where only authorized users (Super Admin or Admin) can manage user accounts. The Super Admin has full control, with separate menus for managing both users and Admins, while Admins have restricted access to managing only regular users. The application ensures security and flexibility with a user-friendly Swing interface, real-time validation, and clearly defined role-based restrictions.

**Task-03**

In this Java Swing GUI application, after logging in, both Admins and Super Admins will have access to a third option: **Manage Restaurant Items**. This feature allows users to manage both categories and individual items within the restaurant system.

**Features of the Manage Restaurant Items Option:**

1. **Manage Categories**:
   * Users can add new categories, which will be stored in the database with a **Category ID** and **Category Name**.
   * This section will also allow for editing and deleting existing categories, ensuring that the category structure can be maintained and updated as needed.
2. **Manage Items**:
   * Users can add new items to the restaurant's menu. For each item, details will be stored in the database, including **Item ID**, **Item Name**, **Category ID** (which must be selected from a dropdown that lists all available categories), **Price**, and **Stock**.
   * The application will ensure that the **Category ID** is stored as a foreign key in the item table, linking each item to its corresponding category.
   * Similar to categories, this section will also provide options to edit or delete existing items, allowing for easy management of the restaurant's offerings.

This design enables both Admins and Super Admins to effectively manage restaurant items and categories, with the ability to perform CRUD operations. It ensures that the application remains flexible and user-friendly while maintaining the integrity of the database relationships between categories and items.

**Task-04**

In this fourth task, the Java Swing application allows regular users to log in, view available items, and place orders with a simple checkout process. Users can log in using their email and password, and upon successful authentication, they’re presented with the complete catalog of restaurant items. Each item displays relevant information, including name, category, price, stock, and an optional image for a more visual experience.

Once logged in, users can browse the catalog and use the **Order** option to place an order. When an order is made, an entry is automatically created in the **Orders** table of the database. This table records essential details, such as a unique Order ID, User ID, order timestamp (capturing the exact date and time the order was placed), and the initial status of the order (set to "Pending" by default). The system also includes **Cash on Delivery** as the default payment method for all orders, simplifying the payment process for users and ensuring consistency.

This task ensures users can navigate a user-friendly, secure interface to view items and make purchases, while the database efficiently manages order data with critical tracking information for future updates and reference.

**Task-05**

n this fifth task, the Java Swing application introduces advanced order management features for Admins and added functionality for regular users to monitor and manage their orders and profiles.

**Admin Features:**

Admins can view all orders placed by users and update the order statuses. They have three status options—"Pending," "Processed," and "Completed"—to reflect the order’s progress. When an Admin marks an order as "Completed," the system automatically reduces the stock count for each ordered item, ensuring the inventory remains accurate and up-to-date in the database. This feature provides seamless inventory tracking aligned with order fulfillment, giving Admins better control over stock levels.

**User Features:**

For regular users, the application provides tools to manage their orders and personal profiles. Users can track the status of each order by entering a unique Order ID in the **Track Order** section, allowing them to follow their order’s journey from placement to completion. Additionally, users have access to a comprehensive **Order History**, where they can view details of past orders, including order dates, items, and statuses.

The application also includes a **Profile Management** feature, where users can update personal information like their password and name. This allows users to maintain current contact details and secure access to their accounts, creating a more user-centered experience.

With these features, the application offers Admins robust tools for efficient order and inventory management while providing regular users greater control and transparency over their account and order status.