**Topkapi Hazinesi**

**Group Members:** Barış Ava – 23040102030, Abdulrahim Manafikhi – 23040102007

**GitHub Repository Link:** https://github.com/AbdulrahimMnf/Topkap-Hazinesi

**Project Description**

Topkapi Hazinesi is a desktop expense tracking application developed in Java using JavaFX. It allows users to register and log in to manage expenses by category. Users can add new expenses, delete them, and view them in a categorized list. The application stores data in a MySQL database via JDBC and provides dynamic, date-filterable expense reports. The interface is designed for usability with modern UI controls.

**Key Features**

* User registration, login , logout
* Categorized expense entry and deletion (e.g., Food, Market, Shopping categories)
* Total expense tracking and filtering by date range to generate reports
* Splash screen at startup with branding elements
* Persistent data storage in MySQL using JDBC for all user/expense data

**Application Flow**

1. **Launch & Splash:**  
   The app displays a splash screen for branding and loading dependencies.
2. **Authentication:**  
   Users can register or log in , log out.
3. **Dashboard:**  
   After login, users access the main dashboard where expenses are managed by category.
4. **Expense Entry & Tracking:**  
   Users can add, edit, or delete expenses. Total values update automatically.
5. **Reports:**  
   Users filter records by date and category to generate expense summaries.
6. **Logout/Exit**:  
   Users can log out securely or close the application.

**Technologies Used**

The application is implemented in Java (Java SE) using JavaFX for the graphical user interface. We used JavaFX Scene Builder to design the FXML layouts and JFoenix for modern material-design styled components. Maven manages the project dependencies and build process. Data persistence is handled with a MySQL database accessed via JDBC for secure storage and retrieval of user and expense data.

**Challenges and Solutions**

During development, we encountered challenges with database connectivity, UI consistency, and implementing date filtering. Establishing a reliable MySQL connection required careful handling of JDBC connections and SQL exceptions; this was solved by creating a dedicated DatabaseConnection utility class. Ensuring a consistent and responsive interface was addressed by using JavaFX property bindings and separating UI logic into controller classes following an MVC pattern. Implementing date-based report filtering involved parsing date inputs and querying the database accordingly, with error handling to ensure the application remains stable on invalid input.

**Conclusion**

In conclusion, this project provided practical experience in JavaFX development and database integration. Topkapi Hazinesi implements the required functionality with a user-friendly interface. We gained important experience in structuring a desktop application, handling user authentication, and managing persistent data. The final software meets the course objectives by offering a functional expense tracker with dynamic reporting. Future enhancements could include additional report types or multi-user support.

**Feature Checklist**

| **Feature** | **Successfully realized (Yes/No)** | **Source code file names** |
| --- | --- | --- |
| Basic functionality | Yes | Main.java, HomeController.java, Expense.java |
| Authentication | Yes | AuthController.java |
| File processing | Yes | DatabaseConnection.java |
| Splash screen | Yes | SplashScreen.java |
| Guest login | Yes | AuthController.java |