INTRODUCING THE RESTAURANT REVIEW SYSTEM

- **Why I Made This Project**
- Easy for Beginners
- I wanted to build a food review system that feels simple and easy to use, even for someone who's not very tech-savvy.
- Too Many Complex Apps
- Big review platforms have too many features — my goal was to keep things focused and clean.
- Fun and Interactive
- I added stars, emojis, flames, and colorful usernames to make the app feel lively and more fun to interact with.
- Built from Scratch with Java
- I chose to build everything from scratch using Java and Swing, so I could learn how real-world GUI systems work.
- Inspired by Real Apps
- Apps like Yelp and Google Reviews inspired the layout, but I made a smaller desktop version with basic but useful features.

Classes and OOP Structure Overview

Understanding the Key Components of the System

- **Reviewer Class**

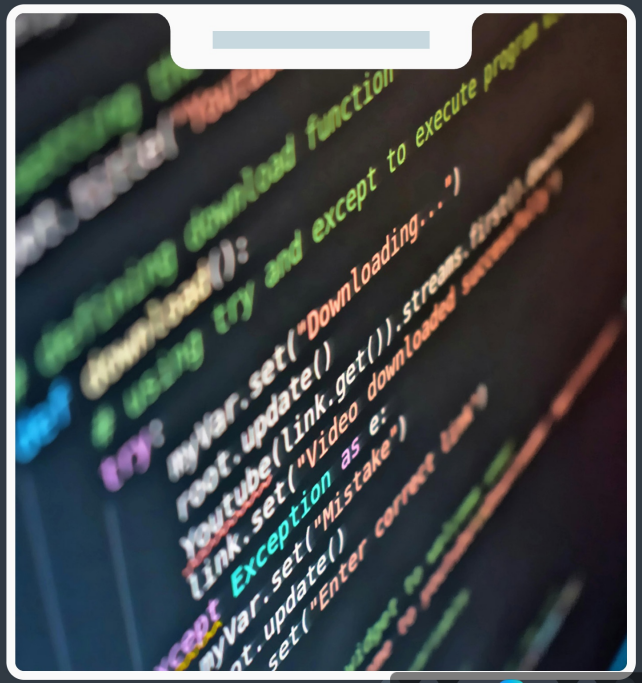
  Extends User class to manage and store reviews submitted by users.

- **Review Class**

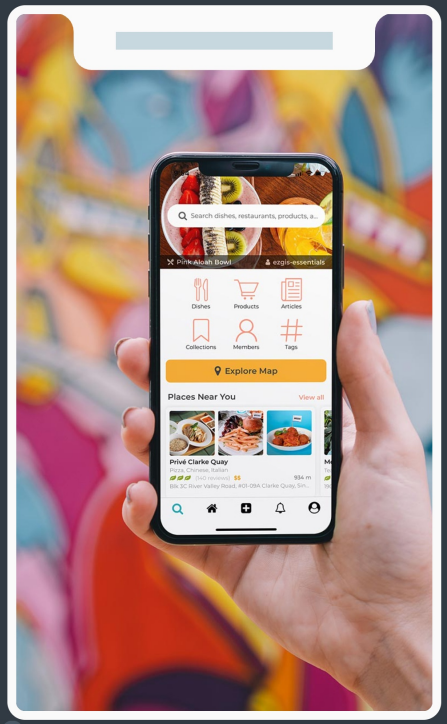
  Contains essential details such as restaurant, rating, and user comments.

- **RestaurantReviewSystem Class**

  Manages the user interface and overall program flow for the application.



🛠️ Features of My Review System



01 🪄 **Features of My Review System**

This slide highlights the key features that make the review system user-friendly and engaging.

02 🍴 **Restaurant cards with star ratings, names, and emojis**

Each restaurant is presented with a card that includes its name, star rating, and fun emojis to enhance visual appeal.

03 ✏️ **Edit button opens a popup window to write reviews**

Users can easily edit their reviews through a convenient popup interface.

04 💬 **Users can type their review and choose a star rating (1–5)**

The system allows users to provide detailed feedback along with a star rating.

05 😃 **Emoji changes based on the rating selected**

The emoji displayed will change according to the star rating chosen by the user, adding a fun element.

06 🔥 **Menu item ratings using flame icons (top 5 items)**

Highlighting the top 5 menu items with flame icons to indicate popularity.

07 💲 **Menu items also show prices**

Users can see the prices of menu items directly on the restaurant cards.

08 🎨 **Fake user reviews appear with colorful usernames**

To create a lively atmosphere, user reviews are displayed with vibrant usernames.

09 🖤 **Dark theme with a clean layout**

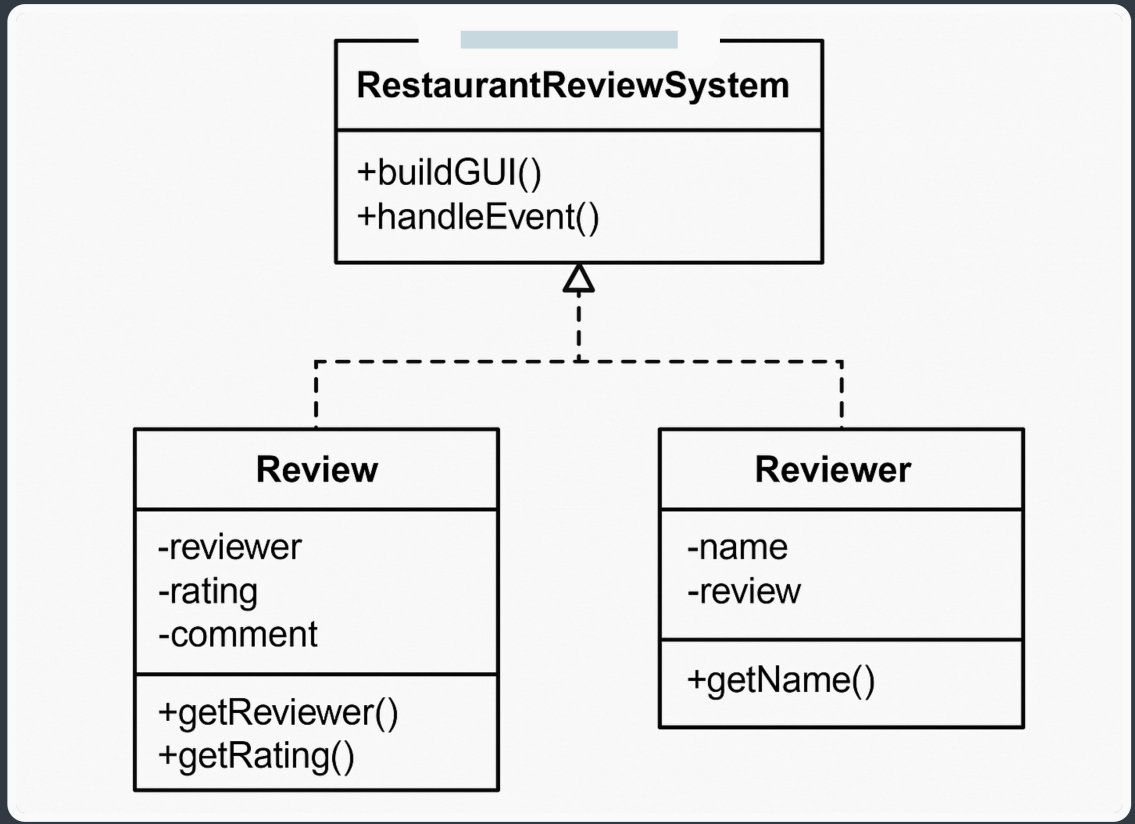
The application features a sleek dark theme that enhances readability and aesthetics.

10 🔄 **Sort dropdown to organize restaurants by name or rating**

Users can easily sort the restaurant list based on their preferences.

11 👷 **Entirely built using Java Swing (AWT + Swing components)**

The application is developed using Java Swing, ensuring a robust and responsive user interface.



UML Class Diagram Overview

Understanding Class Relationships in UML

System Design and UML Classes

- **Understanding Class Relationships in UML:**
- RestaurantReviewSystem → Main class that builds the interface and manages all event
- Review → Stores each review: reviewer name, stars, and comment
- Reviewer → Stores current user info and links to their review
- Other Notes: Data is stored using HashMaps for each restaurant
- Only one editable review per user per restaurant
- Everything is designed using object oriented programming

What I Learned from This Project

01 How to create full GUI apps in Java

02 How to use Java Swing components (buttons, panels, text boxes)

03 How to organize code using OOP (classes and objects)

04 How to use HashMaps and ArrayLists to store and update data

05 How to handle user actions with event listeners

06 How to design a user-friendly interface with good layout and visuals

07 How to make apps feel realistic and fun with emojis, color, and interactivity