TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

FACULTY OF AUTOMATICS AND COMPUTER SCIENCE

DEPARTMENT OF AUTOMATION AND APPLIED INFORMATICS

SEMESTER PROJECT

TravelGo – travel assistant

at

Databases

Sem. **II**, Year **2022-2023**

Capătă Andrei-Bogdan

Scientific supervisor:

Cristi Mocan, Ph.D.

Table of contents:

Introduction:

Project requirements:

Build a database corresponding to a field of interest (e.g., e-commerce, hotel management, medical practice) that contains a minimum of 5 tables, as well as the following elements:

- Database diagram (1-n relationships, m-n relationships, recursive relationships) - presentation of the tables individually (structure + populated data)
- Constraints
- Mandatory usage of: JOIN operation, aggregate functions, SELECT within SELECT, NULL, Self-JOIN
- SQL Statements using INSERT UPDATE DELETE
- Order By, Top, Group By
- Views (at least 4)

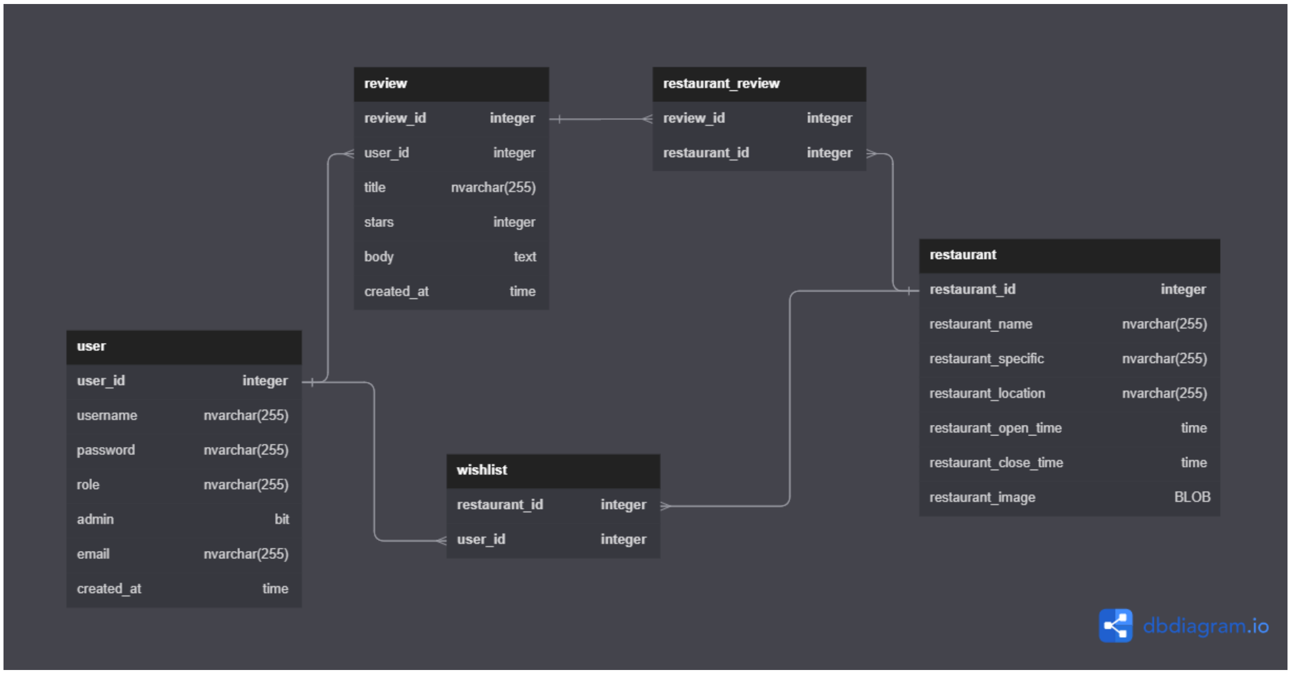
Regarding the WEB interface: Login - username - password (Roles with different rights for Administrator and User) Build the WEB interface from scratch. The requirements are more related to the database being functional, accessible, and operable through the WEB interface. The theme must be accompanied by documentation describing the project domain, database structure (diagram), as well as other implemented elements in the database (in .docx format).

Project theme:

TravelGo is a comprehensive travel companion designed to enhance users' dining experiences by providing a platform for them to share restaurant reviews and add establishments to their personal wishlists. In addition to browsing and contributing reviews, registered users have the ability to save restaurants to their wishlists for future reference. To maintain an up-to-date database, TravelGo's administrators are responsible for adding new restaurants to the platform.

Database:

Database diagram:



review	
review_id	integer
user_id	integer
title	nvarchar(255)
stars	integer
body	text
created_at	time

restaurant_review	
review_id	integer
restaurant_id	integer

restaurant	
restaurant_id	integer
restaurant_name	nvarchar(255)
restaurant_specific	nvarchar(255)
restaurant_location	nvarchar(255)
restaurant_open_time	time
restaurant_close_time	time
restaurant_image	BLOB

user	
user_id	integer
username	nvarchar(255)
password	nvarchar(255)
role	nvarchar(255)
admin	bit
email	nvarchar(255)
created_at	time

wishlist	
restaurant_id	integer
user_id	integer

Database explanation:

The database schema provided has a logical structure and appropriate relationship mapping between tables.

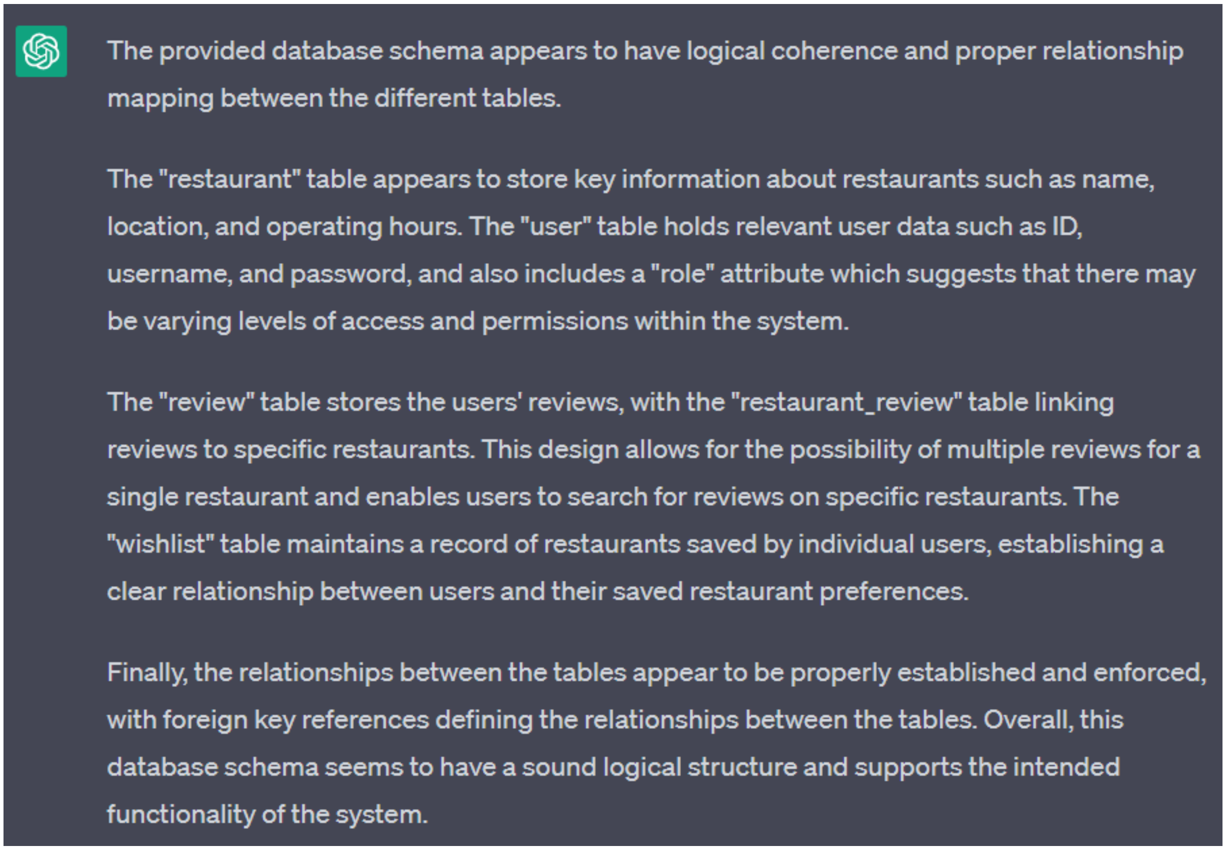
The "restaurant" table stores essential information such as restaurant name, location, operating hours and photos. The "user" table holds user-related data such as ID, username, password, email, role or the admin condition. The "review" table stores user reviews, with the "restaurant_review" table linking reviews to specific restaurants. This design allows for multiple reviews for a single restaurant, and users can search for reviews of specific restaurants. The "wishlist" table stores information about restaurants that individual users have saved, linking users to their saved restaurant preferences.

Foreign key references are used to establish relationships between tables, which are properly enforced. For example, the "user_id" column in the "user" table references the "user_id" column in the "review" table, linking reviews to the user who created them. Similarly, the "review_id" column in the "review" table references the "review_id" column in the "restaurant_review" table, linking restaurant reviews to specific restaurants. Lastly, the "restaurant_id" column in the "restaurant_review" table references the "restaurant_id" column in the "restaurant" table, linking restaurant reviews to their respective restaurants.

Overall, the schema has a well-defined structure and supports the desired functionality of the system.

Database verification:

I used AI to verify my database structure and to ensure that logic is maintained throughout the relations.



The provided database schema appears to have logical coherence and proper relationship mapping between the different tables.

The "restaurant" table appears to store key information about restaurants such as name, location, and operating hours. The "user" table holds relevant user data such as ID, username, and password, and also includes a "role" attribute which suggests that there may be varying levels of access and permissions within the system.

The "review" table stores the users' reviews, with the "restaurant_review" table linking reviews to specific restaurants. This design allows for the possibility of multiple reviews for a single restaurant and enables users to search for reviews on specific restaurants. The "wishlist" table maintains a record of restaurants saved by individual users, establishing a clear relationship between users and their saved restaurant preferences.

Finally, the relationships between the tables appear to be properly established and enforced, with foreign key references defining the relationships between the tables. Overall, this database schema seems to have a sound logical structure and supports the intended functionality of the system.

Web interface:

Programs used:

The main program used to build the web interface is Adobe Dreamweaver. I used this application because the focus is on the database and not on the design itself, i could code and view the changes in real-time.

Adobe Dreamweaver is a software application used for creating and designing websites and web applications. It offers a visual design interface and a code view, allowing users to create web pages without having to write code manually. Dreamweaver includes pre-built components and CSS styles that make it easier to create visually appealing designs. It also includes features for managing website files, such as FTP support for uploading files to a web server, and supports collaborative web development through its integration with Adobe Creative Cloud.

Dreamweaver provides tools for testing and debugging web pages, including previewing web pages in different web browsers and a built-in debugger that helps identify and fix errors in the code. Overall, Dreamweaver is a powerful web development tool that provides a range of features for creating, designing, and managing websites and web applications, suitable for both novice and experienced web developers.

I also used Visual Studio Code, witch is a powerful and versatile code editor that has become immensely popular among developers. With its sleek and intuitive interface, it offers a seamless coding experience across various programming languages and platforms. Visual Studio Code provides an extensive range of features, including intelligent code completion, syntax highlighting, and debugging capabilities, which enhance productivity and streamline the development process. Its vast ecosystem of extensions allows developers to customize and enhance their coding environment to suit their specific needs. Whether you're working on a small project or a large-scale application, Visual Studio Code is a reliable and efficient tool that empowers developers to write clean, efficient code with ease.

For hosting i opted for MoWeS Portable 2 wich is a portable web server package that includes Apache2 (Version 2.2.11), MySQL5 (Version 5.5.8), and PHP5 (Version 5.3.5). This comprehensive package enables users to create and host dynamic websites or web applications locally without the need for a dedicated server. Apache2 serves as the web server, allowing users to handle HTTP requests and deliver web content efficiently. MySQL5 acts as the database management system, providing a robust and reliable platform for storing and retrieving data. PHP5, a server-side scripting language, empowers developers to write dynamic and interactive web pages. Together, these installed packages in MoWeS Portable 2 offer a complete web development environment, allowing users to build and test their websites or web applications conveniently on their own machines.

Design approach:

In the process of designing the website, a palette of five distinct colors was selected to enhance the overall aesthetic appeal. The selected colors were #0FC60D (lime green), #2A2D34 (gunmetal), #6184D8 (glaucous), #F26430 (giants orange), and #6761A8 (ultraviolet). These colors were chosen with the intention of creating a modern and visually appealing design for the website.

Furthermore, in order to improve the user experience and ensure user-friendliness, a simple UX/UI design was implemented. This approach prioritizes ease-of-use and clarity, with a focus on streamlined navigation and intuitive interactions. By utilizing a modern color palette and a simple UX/UI design, the website aims to create a visually pleasing and user-friendly experience for its visitors.

Database implementation:

To incorporate a database into your Adobe Dreamweaver website, you must start by preparing the database and setting it up on your server. After creating the database, you need to create a user account with appropriate permissions to access the database.

In Dreamweaver, navigate to the "Database" tab within the "Application" panel. Click on the "Connection" option and provide the necessary credentials to connect to the database. Once the connection has been established, you can begin to build queries and incorporate database functionality into your website.

It is essential to make sure the database and Dreamweaver settings are correctly configured to avoid errors when attempting to interact with the database.

After finishing the design I imported it on my local web-server and continued to make final retouches using live editors and debugging in the web-console.

Bibliography:

- Adobe Dreamweaver product
- AI verification tool
- Database diagram design
- Palette picker