

Homework 3

Development of 'Carmax' Inventory Management Program

Objective:

Create a Python-based GUI application, named "Carmax," for managing a car inventory system. This application will interface with a MySQL database hosted on Google Cloud SQL, providing functionalities such as displaying, searching, updating, deleting, and adding car inventory records.

	VIN	Brand	Model	Year	Mileage	Price	Color
1	392XT	Honda	Civic	2019	3938	17000	Black
2	DS213	Toyota	Camry	2020	2372	24000	Silver
3	K34W2	Ford	F150	2018	19322	41000	Red
4	UE32L	Hyundai	Sonata	2017	25832	18000	White
5	2M3G8	BMW	325	2019	8732	35000	Blue

Hello, Dr. Kim!

Search Add Remove Update

VIN

Brand Model Year

Mileage Price Color

Program Requirements:

Startup Display:

Upon launching, the application should automatically display all entries from the car inventory table.

Display the developer's name prominently at the top of the application window.

The displayed inventory table should be scrollable. It must accommodate a variable number of rows, becoming scrollable when exceeding 10 rows.

Search Functionality:

Implement a 'Search' feature allowing users to query the inventory using one out of eight attributes: VIN, Brand, Model, Year, Mileage, Price, Color, and Condition (Used/New).

The search must be executed based on a single attribute selection; compound searches across multiple attributes are not permitted.

If the search yields results, display all matching entries. If no matches are found, present an empty table alongside a message indicating the absence of records matching the search criteria.

A scenario where no search criteria are provided should result in the display of the entire inventory.

Include validation to alert users when attempting a search with multiple criteria filled, indicating that only one attribute should be used for searching.

Deletion of Records:

Enable users to delete inventory records via a 'Delete' button. This function should only be accessible through the provision of a valid VIN number.

If an attempt is made to delete without specifying a VIN, display a warning message.

Updating Records:

Facilitate the updating of any car's information through an 'Update' function. Users must provide both the VIN number and the new value for a single field they wish to update.

If the 'Update' button is pressed without a VIN, notify the user with a warning message.

Ensure that only one field can be updated per operation.

Adding New Records:

Incorporate an 'Add' function allowing users to insert new car records into the database.

Completion of all eight fields is required for this operation to proceed.

Display a warning message if the user attempts to add a new record without filling all required fields.

Technical Specifications:

The database connectivity must be established using Google Cloud MySQL to facilitate remote access and grading.

The GUI layout should include a message bar situated between the inventory table and the operation buttons (Search, Delete, Update, Add) to relay alerts and instructions to the user.

Submission Requirements:

Submit the source code file(s) for the application.

Include a screenshot of the application's interface in action, capturing the full window.

Evaluation Rubric for 'Carmax' Inventory Management Program

Your submission will be evaluated based on the following criteria. Ensure your application meets these requirements for a comprehensive assessment.

Functionality (40 points):

Startup Display (5 points): The application correctly displays all rows from the car inventory table upon startup, including a scrollable feature for tables with more than 10 rows.

Search Feature (10 points): Implements a functional search feature allowing queries based on one of eight fields (VIN, brand, model, year, mileage, price, color, used/new) and displays relevant results accurately. Includes handling for empty or invalid searches.

Delete Function (5 points): Enables deletion of car records using the VIN, with appropriate messaging for cases without VIN input.

Update Function (10 points): Allows for updating a specific field of a car's profile using its VIN. Correctly updates the database and limits updates to one field at a time.

Add Function (10 points): Supports adding new car records to the database with all required fields filled. Properly handles cases with missing information.

User Interface Design (20 points):

Layout and Accessibility (10 points): The application features a user-friendly interface, including a clearly visible display of the user's name, a well-positioned message bar, and intuitive button placement for primary functions.

Visual Appeal and Usability (10 points): The design is visually appealing and facilitates easy navigation and usability across all application features.

Database Integration (20 points):

Google Cloud SQL Server Usage (10 points): Demonstrates successful integration with a Google Cloud MySQL database, ensuring the application can query, update, and manage data effectively.

Data Handling and Security (10 points): Efficiently manages data transactions with attention to security practices, preventing data leaks and ensuring data integrity.

Code Quality and Documentation (20 points):

Code Efficiency and Readability (10 points): The source code is well-organized, efficiently written, and easy to read, with clear variable naming and logical structuring.

Documentation and Comments (10 points): Includes comprehensive documentation and comments within the code, explaining the functionality of major sections and complex logic for ease of understanding.