**AutoGallery**

Experience the future of car showroom management with AutoGallery – innovation, efficiency, and elegance combined, built with Java specifically for showcasing OOP concepts.

**Project Structure**

The project consists of the following main components:

* Main class: Contains the primary logic for the application, including the main menu and user input handling.
* Showroom class: Represents a showroom with its details.
* Employees class: Represents an employee associated with a showroom.
* Cars class: Represents a car available in a showroom.
* utility interface: Defines methods for setting and getting details of showrooms, employees, and cars.

**Classes and Methods**

**Main Class**

* **APP\_NAME**: Constant for the application name.
* **input**: Scanner object for user input.
* **main\_menu()**: Displays the main menu to the user.
* **main(String[] args)**: Main method that starts the application and manages user input.
* **take\_input()**: Utility method to take user input.
* **does\_showroom\_exist(Showroom[] showrooms, int showroom\_counter, Object obj)**: Checks if a showroom exists.

**Showroom Class**

* **showroom\_id**: Unique identifier for the showroom.
* **showroom\_name**: Name of the showroom.
* **showroom\_address**: Address of the showroom.
* **manager\_name**: Name of the showroom manager.
* **total\_cars\_in\_stock**: Number of cars in stock.
* **total\_employees**: Number of employees.
* **get\_showroom\_id()**: Returns the showroom ID.
* **get\_details()**: Displays the details of the showroom.

**set\_details(): Sets the details of the showroom.**

**Showroom class**

* **showroom\_id**: Unique identifier for the showroom.
* **emp\_id**: Unique identifier for the employee.
* **emp\_name**: Name of the employee.
* **emp\_age**: Age of the employee.
* **emp\_department**: Department of the employee.
* **set\_showroom\_id(String sid)**: Sets the showroom ID for the employee.
* **get\_details()**: Displays the details of the employee.
* **set\_details()**: Sets the details of the employee.

**Cars class**

* **showroom\_id**: Unique identifier for the showroom.
* **car\_id**: Unique identifier for the car.
* **car\_name**: Name of the car.
* **car\_brand**: Brand of the car.
* **car\_color**: Color of the car.
* **car\_fuel\_type**: Fuel type of the car.
* **car\_price**: Price of the car.
* **car\_type**: Type of the car (sedan/SUV/HatchBack).
* **car\_transmission**: Transmission type of the car.
* **set\_showroom\_id(String sid)**: Sets the showroom ID for the car.
* **get\_details()**: Displays the details of the car.
* **set\_details()**: Sets the details of the car.

**Utility Interface**

* **input**: Scanner object for user input.
* **get\_details()**: Method to get details (to be implemented by classes).
* **set\_details()**: Method to set details (to be implemented by classes).**Add Showrooms**: Allows the user to add a new showroom by entering its details.

**Usage**

1. **Add Employees**: Allows the user to add a new employee by entering their details. Requires at least one showroom to be added first.
2. **Add Cars**: Allows the user to add a new car by entering its details. Requires at least one showroom to be added first.
3. **Get Showrooms**: Displays the details of all showrooms added.
4. **Get Employees**: Displays the details of all employees added.
5. **Get Cars**: Displays the details of all cars added.
6. **Exit**: Exits the application.

**Conclusion:**

The AutoGallery project provides a simple and intuitive interface for managing showrooms, employees, and cars. By following the structured menu options, users can easily add and view details of the entities within the system. This project not only demonstrates practical applications of Java but also highlights key Object-Oriented Programming (OOP) concepts in a real-world scenario.