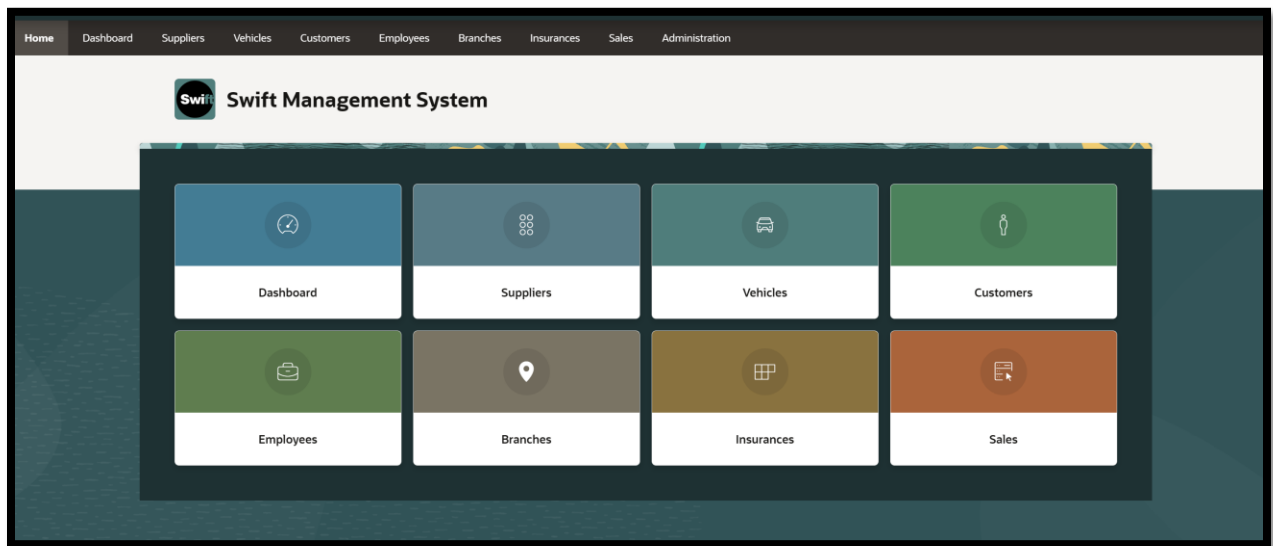


DATABASE - (PROJECT)

Automotive Dealership Management System



Submitted By:

Aqsa Nazakat - 232391

Ali Hassan – 232406

Submitted To:

Ma'am Kainat

Background:

Swift Motor is an automotive dealership having a network of automotive dealerships with several locations in various areas of the countries specifically dealing in selling a large selection of new and used vehicles. In order to improve the way of organizing data not in the form of files, they have been looking out for an application solution that caters their needs of having their entire data stored on a virtual database that is accessible across all their locations. The database would be having different user types, which would increase overall management efficiency, and streamline administrative operations.

1) Introduction

a. Overview of the Application

The database application is not just an ordinary locally hosted single user database but is actually live on the web and that too for no cost. However, this project is very valuable in terms of the efforts to bring it into what it has come across from the initial phases of database designing to the entire development.

This application is solely for the use by administrative or employees of the dealership; however, customers can get privileges to view the inventory of vehicles.

Users of the application

[Stakeholders or the employees]

- **Dealership Owners:** Concerned with the profitability and performance of their business.
- **Branch / Sales / Inventory Managers:** Assigned in charge of running each branch's daily operations on the database like searching up vehicles, and catering with the suppliers.
- **Sales Team:** The sales team whom is responsible of establishing deals among clients.
- **IT Staff:** handling System integrity and database maintenance.

Application Pages/Features:

Home: The database system's home page has instant access to every section ie. Dashboard, Suppliers, Vehicles, Customers, Employees, Branches, Insurances and Sales Page.

Dashboard: Provides visuals of each data from the tables in the database. Different charts and graphs visualize the data from the tables of customers, sales, category of vehicles in the inventory and the types of transmission of each vehicles.

Suppliers: Retrieves the supplier information ie. Their names and their email contact from the suppliers tables.

Vehicles: Gets the information of every car available in all branches, including make, model, year, price, engine size and category from the vehicles table.

Semester Project

Customers: The section of customers keeps track of all the customers that were successful in either buying or selling a vehicle to the dealership. This extracts their first name, last name, email, phone and type of customer they were from the customers table.

Sales: A segment focused on the entire record of the sales which gets data from the union of different tables like the vehicles, customers and the sales table providing information such as the vehicle id, customer id, employee id, and the sale date and price across different tables.

Similarly, Branches, Insurance and Employees table which gets data from different tables from the database.

Administration: This specific section is only in the interface of the admin user which provides ways to customize the interface and handling of data.

b. Objectives and Goals of the Application

Centralized Data Management: To improve data accessibility and management, all dealership-related data are to be centralized into a single application, which allows access, and changes, from anywhere around the world.

User Interfaces: Each user would have a different interface. The database administrator would have a separate administrative section, which works on the access control of the database, the customization of the interface etc that the other users will not have. Other users would only be capable of adding, deleting, or modifying the existing data in each section of the table.

Reporting and Analytics: Allowing users to view visualized reporting and analytics of the sales, classification of vehicles, deals, and total sales, so that they may make honorable and successful decisions using data.

Management Operations: Dynamic management like adding, removing or updating of data from the live application from user-end would respectively also update the database table that has been modified on the backend whilst also updating the visual reports they are linked to, on the dashboard page.

User Feedback: To make sure the database is working well, any user recommendations are to be valued a lot which in our case are received on the apex's backend.

c. Significance of the Project

This is a fully functioning and interactive database application deployed live on the web, supporting different types of users whilst each having their respective interfaces.

Semester Project

The application will greatly improve operational efficiency like managing employee, customer, vehicles, sales, insurance, and records as it has respective section for each department involved in the dealership management. Furthermore, interactive visuals like charts, and graphs from the tables in the database allow the stakeholders, and the big players to take thoughtful decisions.

Ultimately, by improving their operational skills, and providing greater customer service, the dealership will have a competitive advantage.

Preservice of data within the users of the database, and the errors of humans would reduce significantly.

2) System Analysis

a. Requirements gathering

To develop an effective database application for an automotive dealership, gathering comprehensive requirements is important. All data entities were identified which were expected to be involved in the database (e.g., vehicles, customers, sales, employees, suppliers, branches, insurances). The attributes for each entity (i.e., for vehicles: make, model, year, price, engine size, category) were designated for all the entities in a similar manner.

b. Functional requirements

User Management:

- Admin functionalities for adding, removing, and modifying user roles and permissions.
- Creating and modifying user interfaces and access levels for dealership owners, branch managers, sales team, and IT staff.

Data Management:

- Centralized database accessible from all locations.
- Capabilities for adding, updating, and deleting records in tables like Vehicles, Customers, Employees, Sales, and Suppliers.

Search and Retrieval:

- Advanced search functionality to search specific vehicles, customers, sales records by also implying filters to narrow down search results based.

Reporting and Analytics:

- Dashboards displaying data from respective tables using charts and graphs.
- Reports on sales performance, inventory status, total sales, etc.

Inventory Management:

- Real-time tracking of vehicle inventory across all branches from anywhere.
- Features for categorizing vehicles based on make, model, year, price, and other attributes.

Semester Project

Sales Management:

- Recording of all sales transactions, including details like vehicle ID, customer ID, employee ID, sale date, and price.
- Integration with inventory to update vehicle status upon sale.

Supplier Management:

- Database of supplier information, including names, contact details, and associated vehicles.

Customer Management:

- Tracking of customers, their purchase history, and they were either a buyer or a seller.

c. Non Functional requirements

Security:

Maybe a two factor authentication would further secure the database's access from any external party user or attacker who has no relation with the dealership or its software management.

3) System and Database Design

a. Database Design

Tables and Relationships:

- **Users Table:** Stores information about all users (e.g., ID, name, role, email, password).
- **Vehicles Table:** Contains details of all vehicles (e.g., ID, make, model, year, price, engine size, category, branch ID).
- **Customers Table:** Stores customer information (e.g., ID, first name, last name, email, phone, customer type).
- **Sales Table:** Records sales transactions (e.g., sale ID, vehicle ID, customer ID, employee ID, sale date, price).
- **Suppliers Table:** Contains supplier details (e.g., ID, name, email).
- **Branches Table:** Stores information about different dealership branches (e.g., ID, location, manager ID).
- **Employees Table:** Contains details of employees (e.g., ID, name, role, branch ID, contact info).
- **Insurance Table:** Stores insurance-related information (e.g., policy ID, vehicle ID, provider, coverage details).

Relationships:

- Vehicles are linked to branches, suppliers, and sales.

Semester Project

- Sales link vehicles, customers, and employees.
- Branches are linked to managers and employees.

b. Application Architecture

Frontend:

- **User Interfaces:** Separate interfaces for different user roles.
- **Dashboard:** Real-time data visualization using charts and graphs.
- **Forms:** For data entry and updates (e.g., adding new vehicles, employees, recording sales etc).

Backend:

- **API Layer:** Handles requests from the frontend and communicates with the database.
- **Database Management System (DBMS):** Apex Centralized database storing all dealership data.

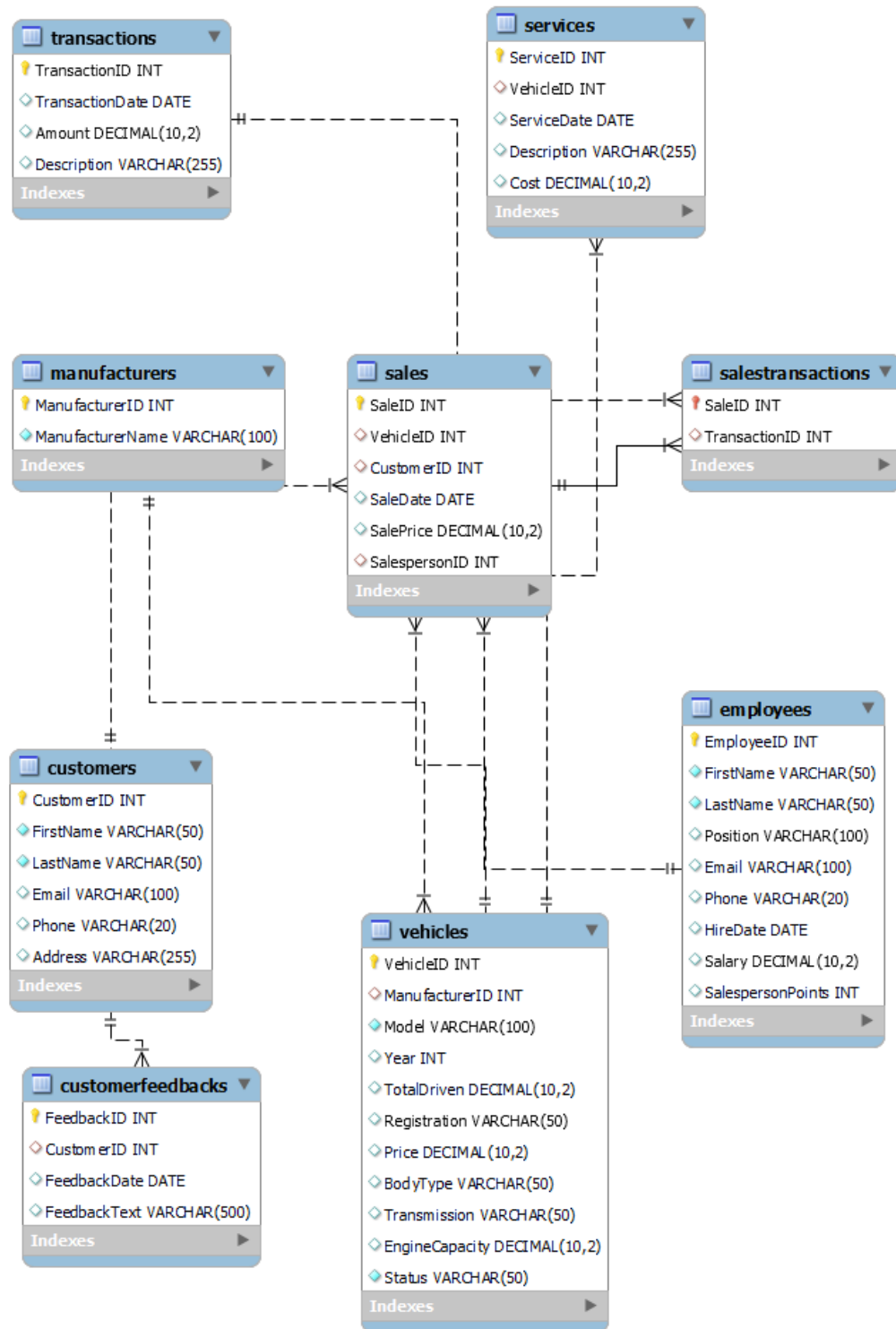
Security:

- **Authentication:** User login system to verify identity.
- **Authorization:** Role-based access control to allocate access to different users.

c. Entity-Relation Diagram

On the next page.

Semester Project



4) Implementation

Technologies and Tools Used for the implementation were as below

1. Oracle APEX Release 23.0.
2. Oracle SQL Developer Web

The SQL including all the DDL, DML, and DCL was implemented through the SQL workshop within the Oracle APEX DBMS.

```
---- suppliers table ----

CREATE TABLE "SUPPLIERS"
(
  "SUPPLIERID" NUMBER(*,0),
  "NAME" VARCHAR2(255),
  "CONTACTINFO" VARCHAR2(255),
  PRIMARY KEY ("SUPPLIERID")
  USING INDEX ENABLE
) ;

---- vehicles table ----

CREATE TABLE "VEHICLES"
(
  "VEHICLEID" NUMBER(*,0),
  "MAKE" VARCHAR2(255),
  "MODEL" VARCHAR2(255),
  "YEAR" NUMBER(*,0),
  "PRICE" NUMBER(10,2),
  "COLOR" VARCHAR2(50),
  "MILEAGE" NUMBER(*,0),
  "VIN" VARCHAR2(17),
  "CATEGORY" VARCHAR2(50),
  "ENGINECAPACITY" NUMBER(*,0),
  "TRANSMISSION" VARCHAR2(50),
  "BRANCHID" NUMBER(*,0),
  PRIMARY KEY ("VEHICLEID")
  USING INDEX ENABLE
) ;

ALTER TABLE "VEHICLES" ADD FOREIGN KEY ("BRANCHID")
  REFERENCES "BRANCHES" ("BRANCHID") ENABLE;

---- sales table ----

CREATE TABLE "SALES"
(
  "SALEID" NUMBER(*,0),
```


Semester Project

```
"VEHICLEID" NUMBER(*,0),
"CUSTOMERID" NUMBER(*,0),
"EMPLOYEEID" NUMBER(*,0),
"SALEDATE" DATE,
"SALEPRICE" NUMBER(10,2),
    PRIMARY KEY ("SALEID")
USING INDEX ENABLE
) ;

ALTER TABLE "SALES" ADD FOREIGN KEY ("VEHICLEID")
    REFERENCES "VEHICLES" ("VEHICLEID") ENABLE;
ALTER TABLE "SALES" ADD FOREIGN KEY ("CUSTOMERID")
    REFERENCES "CUSTOMERS" ("CUSTOMERID") ENABLE;
ALTER TABLE "SALES" ADD FOREIGN KEY ("EMPLOYEEID")
    REFERENCES "EMPLOYEES" ("EMPLOYEEID") ENABLE;

---- leasing table ----

CREATE TABLE "LEASING"
(
    "LEASEID" NUMBER(*,0),
    "LEASENUMBER" VARCHAR2(50),
    "VEHICLEID" NUMBER(*,0),
    "LESSEENAME" VARCHAR2(255),
    "LEASESTARTDATE" DATE,
    "LEASEENDDATE" DATE,
    "MONTHLYPAYMENT" NUMBER(10,2),
    PRIMARY KEY ("LEASEID")
USING INDEX ENABLE
) ;

ALTER TABLE "LEASING" ADD FOREIGN KEY ("VEHICLEID")
    REFERENCES "VEHICLES" ("VEHICLEID") ENABLE;

---- inventory table ----
CREATE TABLE "INVENTORY"
(
    "INVENTORYID" NUMBER(*,0),
    "BRANCHID" NUMBER(*,0),
    "VEHICLEID" NUMBER(*,0),
    "QUANTITY" NUMBER(*,0),
    "DATERECEIVED" DATE,
    "NOTES" VARCHAR2(255),
    "SUPPLIERID" NUMBER(*,0),
    PRIMARY KEY ("INVENTORYID")
USING INDEX ENABLE
```

Semester Project

```
) ;

ALTER TABLE "INVENTORY" ADD FOREIGN KEY ("BRANCHID")
    REFERENCES "BRANCHES" ("BRANCHID") ENABLE;
ALTER TABLE "INVENTORY" ADD FOREIGN KEY ("VEHICLEID")
    REFERENCES "VEHICLES" ("VEHICLEID") ENABLE;
ALTER TABLE "INVENTORY" ADD FOREIGN KEY ("SUPPLIERID")
    REFERENCES "SUPPLIERS" ("SUPPLIERID") ENABLE;

---- insurance table ----

CREATE TABLE "INSURANCE"
(
    "INSURANCEID" NUMBER(*,0),
    "POLICYNUMBER" VARCHAR2(50),
    "VEHICLEID" NUMBER(*,0),
    "INSURANCECOMPANY" VARCHAR2(255),
    "COVERAGEDetails" VARCHAR2(255),
    "EXPIRATIONDATE" DATE,
    PRIMARY KEY ("INSURANCEID")
    USING INDEX ENABLE
) ;

ALTER TABLE "INSURANCE" ADD FOREIGN KEY ("VEHICLEID")
    REFERENCES "VEHICLES" ("VEHICLEID") ENABLE;

---- employees table ----

CREATE TABLE "EMPLOYEES"
(
    "EMPLOYEEID" NUMBER(*,0),
    "FIRSTNAME" VARCHAR2(50),
    "LASTNAME" VARCHAR2(50),
    "POSITION" VARCHAR2(50),
    "CONTACTINFO" VARCHAR2(255),
    PRIMARY KEY ("EMPLOYEEID")
    USING INDEX ENABLE
) ;

---- customers table ----

CREATE TABLE "CUSTOMERS"
(
    "CUSTOMERID" NUMBER(*,0),
    "FIRSTNAME" VARCHAR2(50),
    "LASTNAME" VARCHAR2(50),
    "EMAIL" VARCHAR2(255),
    "PHONE" VARCHAR2(20),
```

Semester Project

```
"CUSTOMERTYPE" VARCHAR2(50),
    PRIMARY KEY ("CUSTOMERID")
    USING INDEX ENABLE
) ;

---- branches table ----

CREATE TABLE "BRANCHES"
(
    "BRANCHID" NUMBER(*,0),
    "LOCATION" VARCHAR2(255),
    "ADDRESS" VARCHAR2(255),
    "CONTACTINFO" VARCHAR2(255),
    "MANAGERID" NUMBER(*,0),
    PRIMARY KEY ("BRANCHID")
    USING INDEX ENABLE
) ;

ALTER TABLE "BRANCHES" ADD FOREIGN KEY ("MANAGERID")
    REFERENCES "EMPLOYEES" ("EMPLOYEEID") ENABLE;

---- triggers----

CREATE OR REPLACE TRIGGER vehicle_status_trigger
BEFORE INSERT OR UPDATE ON Vehicles
FOR EACH ROW
BEGIN
    IF :NEW.Mileage < :OLD.Mileage THEN
        :NEW.Status := 'Used';
    ELSIF :NEW.Mileage = :OLD.Mileage THEN
        :NEW.Status := 'Unchanged';
    ELSE
        :NEW.Status := 'New';
    END IF;
END;
/

CREATE OR REPLACE TRIGGER vehicle_status_trigger
AFTER INSERT OR UPDATE ON Vehicles
FOR EACH ROW
BEGIN
    IF :NEW.Mileage < :OLD.Mileage THEN
        UPDATE Vehicles
        SET Status = 'Used'
        WHERE VehicleID = :NEW.VehicleID;
```

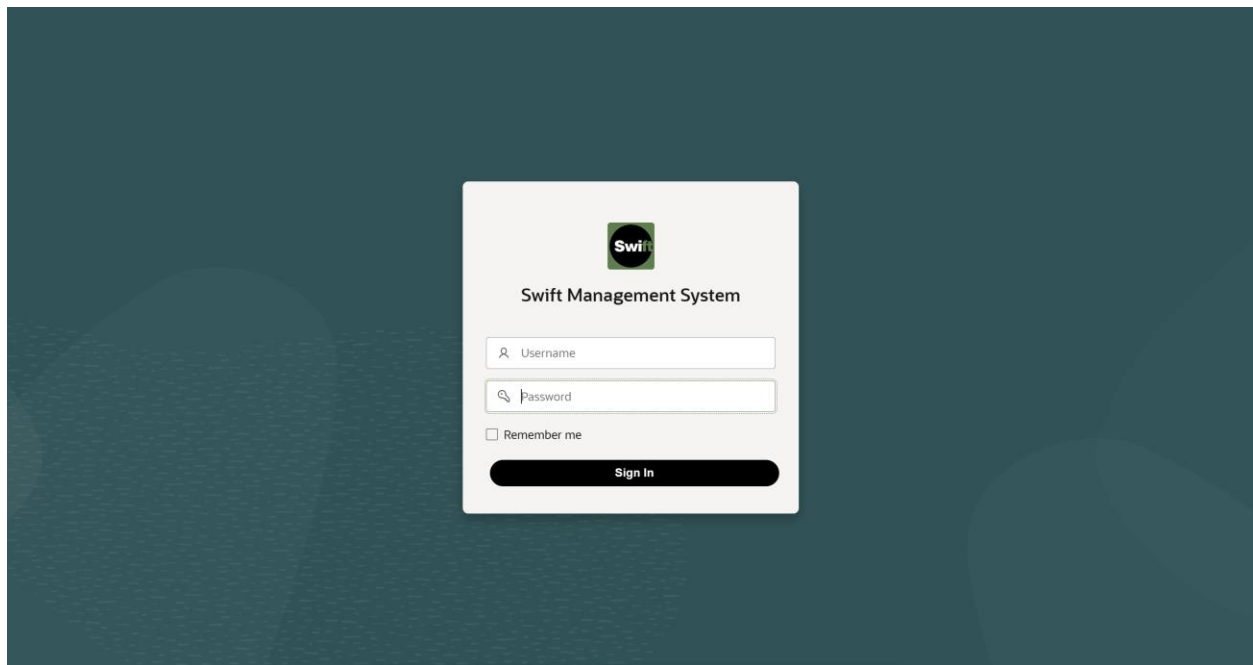
Semester Project

```
ELSIF :NEW.Mileage = :OLD.Mileage THEN
    UPDATE Vehicles
    SET Status = 'Unchanged'
    WHERE VehicleID = :NEW.VehicleID;
ELSE
    UPDATE Vehicles
    SET Status = 'New'
    WHERE VehicleID = :NEW.VehicleID;
END IF;
END;
/
```

5) User Manual

As the application is deployed live on the internet, the first step is to visit the following link

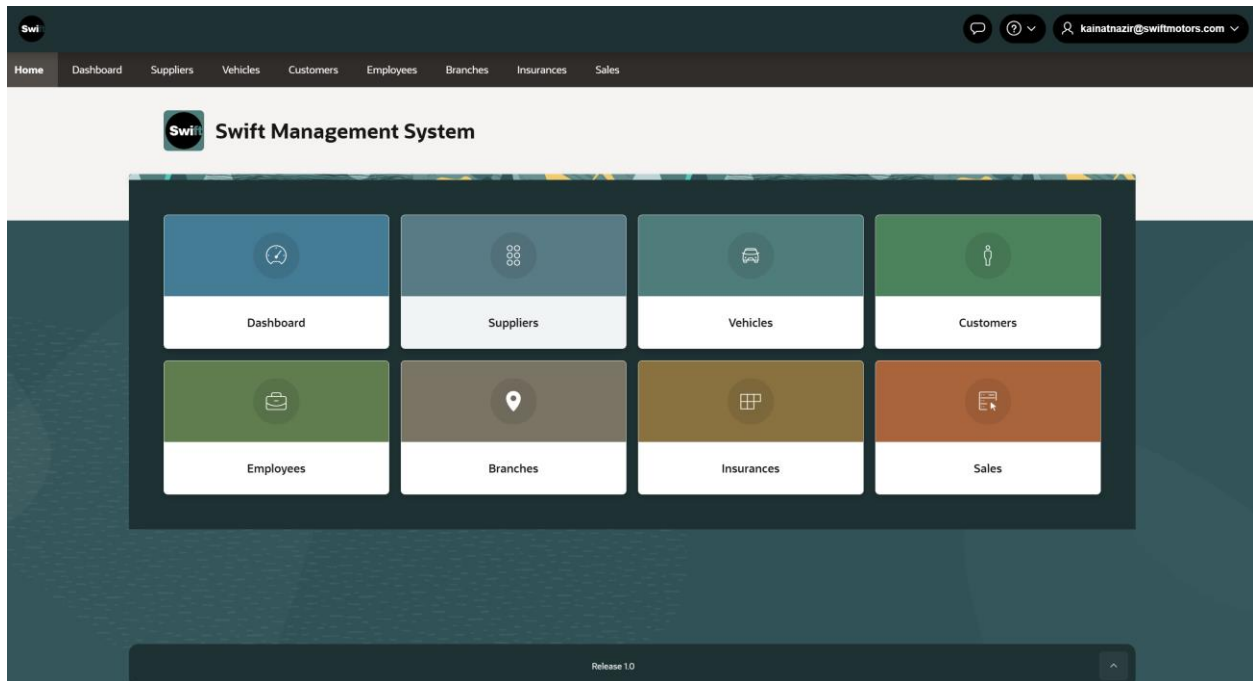
[Swift Management System - Log In \(oracle.com\)](#)



For ma'am a respective account is already created.

The username is kainatnazir@swiftmotors.com and the password is 'swift@2024'. Ma'ams account has a different interface with no access to database administration; however, she can perform every other task in terms of deleting, adding or renaming existing entries of data. An entire record for each user is also recorded on the backend.

Semester Project



Above is the page after first logging in which further has different section, which can be interacted with.

6) Conclusion

The development of a centralized, web-based database application for Swift Motors aims to enhance the efficiency and effectiveness of dealership operations. By merging data management, improving accessibility, and providing powerful reporting tools, this application will support better decision-making and provide a competitive edge. The thorough system analysis, responsive design, and robust implementation plan ensure that the final product will meet the needs of all stakeholders, producing improved operational efficiency and engaging customer service.