THE UNIVERSITY OF SYDNEY

School of Computer Science

COMP9120 Database Management Systems

Assignment 2: Database Application Development

Group Assignment (16%)

Introduction

The objectives of this assignment are to gain practical experience in interacting with a relational database management system using an Application Programming Interface (API) (Python DB-API). This assignment additionally provides an opportunity to use more advanced features of a database such as functions.

This is a group assignment for teams of about 3 members, and it is assumed that you will continue in your Assignment 1 group. You should inform the unit coordinator as soon as possible if you wish to change groups.

Please also keep an eye on your email and any announcements that may be made on Ed.

Submission Details

The final submission of your database application is due at 11:59pm on Sunday 22th October (Week 11). You should submit the *items for submission* (detailed below) via Canvas.

Items for submission

Please submit your solution to Assignment 2 in the 'Assignment' section of the unit's Canvas site by the deadline, including EXACTLY THREE files:

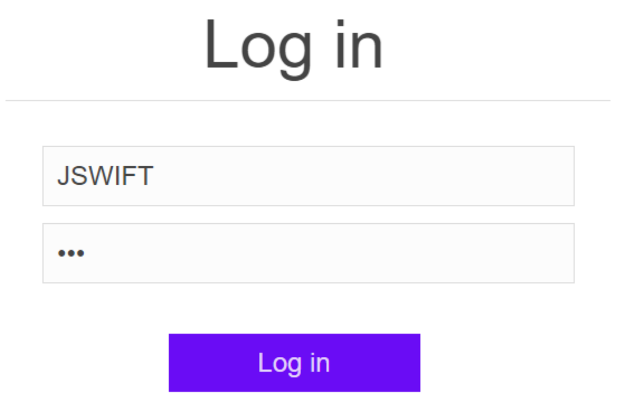
- An assignment coversheet as a PDF document (.pdf file suffix) which is available for download from this link on Canvas.
- A SQL file (**`RACASchema.sql`**) containing the SQL statements necessary to generate the database schema and sample data. This must contain the original schema and insert SQL statements, and any new records and stored procedures you may have added.
- A Python file (**`database.py`**) containing the Python code you have written to access the database.

Task 1: The Rent-A-Car Australia (RACA) System

In this assignment, you will be working with the RACA system which is currently under development. The system still requires work in numerous areas, including interaction with the database. Your main task in this assignment is to handle requests for reads and writes to the database coming from the user interface (UI). We first describe the main features that the RACA system should include from a UI perspective, and then discuss where the majority of your database code needs to be implemented.

Logging In

The first form a user is presented with, when starting the RACA system is the Login, as shown in Figure 1. This feature is still under development and currently requires that an employee enters their username and password to be validated prior to being successfully logged in to the system. Security features such as password encryption/hashing will be implemented at a later stage (and is out of scope for this assignment). Once logged in, the user is taken to the cars list page to view their associated hire cars.



Log in

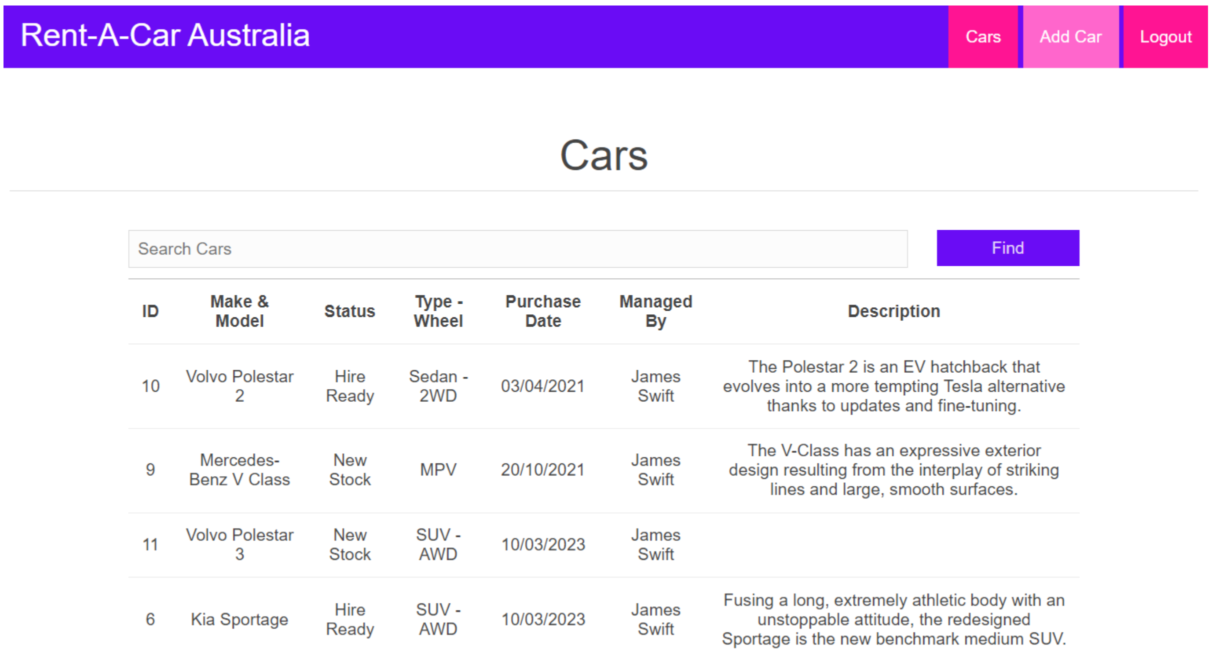
JSWIFT

•••

Figure 1 – Login form

Viewing Cars List

Once a user is logged in, they are shown a list of all their associated hire cars, as shown below in Figure 2. This list must be ordered such that the most recently purchased cars appear at the bottom. The list is also sorted by description in ascending order, and status in descending order. Each car has a make, model, status, type, wheel, purchase date, managing employee, and description. A car is associated with an employee if the employee is allocated as the person in charge of managing all aspects related to the hire car.

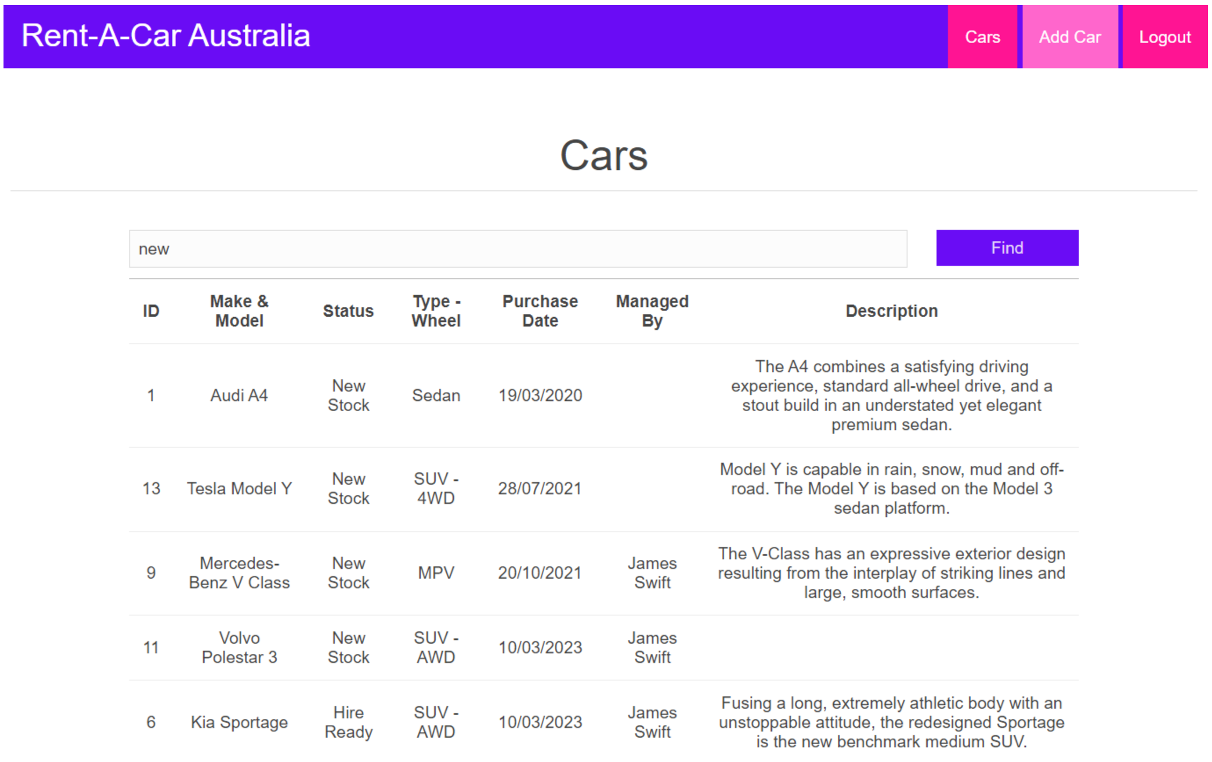


ID	Make & Model	Status	Type - Wheel	Purchase Date	Managed By	Description
10	Volvo Polestar 2	Hire Ready	Sedan - 2WD	03/04/2021	James Swift	The Polestar 2 is an EV hatchback that evolves into a more tempting Tesla alternative thanks to updates and fine-tuning.
9	Mercedes-Benz V Class	New Stock	MPV	20/10/2021	James Swift	The V-Class has an expressive exterior design resulting from the interplay of striking lines and large, smooth surfaces.
11	Volvo Polestar 3	New Stock	SUV - AWD	10/03/2023	James Swift	
6	Kia Sportage	Hire Ready	SUV - AWD	10/03/2023	James Swift	Fusing a long, extremely athletic body with an unstoppable attitude, the redesigned Sportage is the new benchmark medium SUV.

Figure 2 – Viewing Cars List

Finding Cars

A user can search through all hire cars by entering a word or phrase (a ‘keyword’) in the field next to the Find button, as shown in Figure 3, and then clicking on Find. When such a keyword is specified, then the retrieved cars and shown on the list are those that include this word or phrase in the make, model, status, car type, wheel system, managing employee’s full name, or description. The search is case insensitive. For example, given the search keyword ‘new’, Find will return all cars that include the keyword ‘new’ in the make, model, status, car type, wheel system, managing employee’s full name, or description. Searching with a blank/empty keyword field will show all of the logged in user’s associated cars. Any search results returned must be ordered such that hire cars without a managing employee would appear at the top, and then by purchase date in ascending order. The search results must exclude those cars whose purchase dates are older than 15 years (from today's date).



Cars

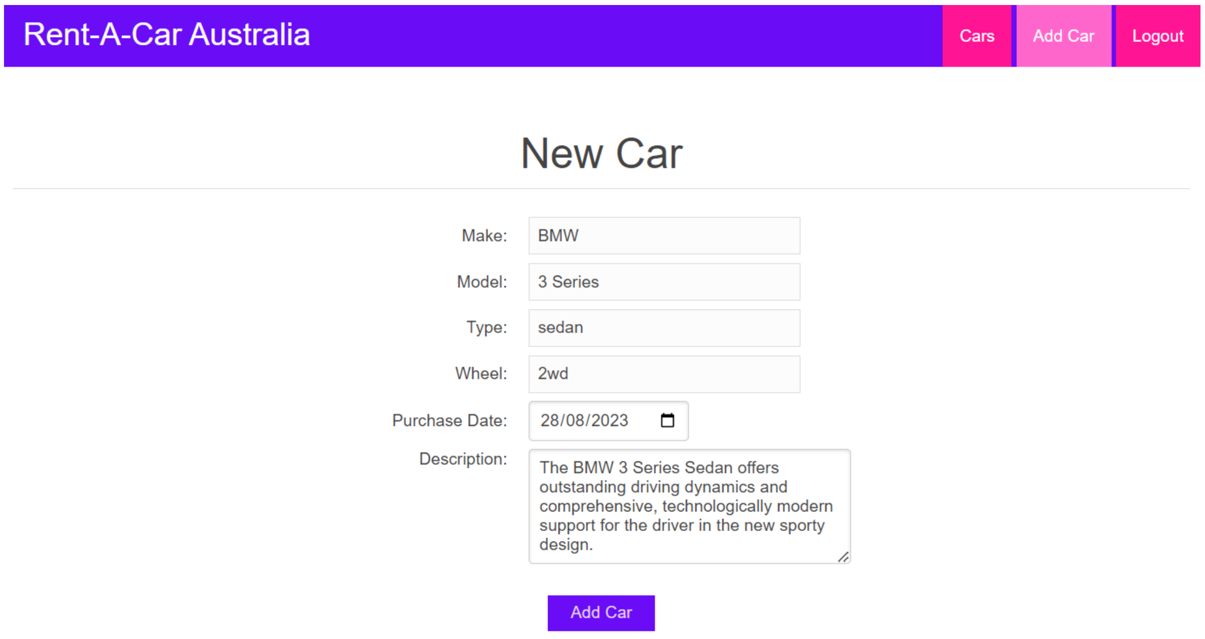
new

ID	Make & Model	Status	Type - Wheel	Purchase Date	Managed By	Description
1	Audi A4	New Stock	Sedan	19/03/2020		The A4 combines a satisfying driving experience, standard all-wheel drive, and a stout build in an understated yet elegant premium sedan.
13	Tesla Model Y	New Stock	SUV - 4WD	28/07/2021		Model Y is capable in rain, snow, mud and off-road. The Model Y is based on the Model 3 sedan platform.
9	Mercedes-Benz V Class	New Stock	MPV	20/10/2021	James Swift	The V-Class has an expressive exterior design resulting from the interplay of striking lines and large, smooth surfaces.
11	Volvo Polestar 3	New Stock	SUV - AWD	10/03/2023	James Swift	
6	Kia Sportage	Hire Ready	SUV - AWD	10/03/2023	James Swift	Fusing a long, extremely athletic body with an unstoppable attitude, the redesigned Sportage is the new benchmark medium SUV.

Figure 3 – Finding Cars

Adding a Car

Users may also add a new hire car by clicking on the *Add Car* tab in the title bar. They may enter car details such as make, model, type, wheel system, purchase date, and description, in the popup dialog. Once the popup dialog appears, click on Add Car button, as shown in Figure 4. A new car has a default status of "New Stock" and will not be assigned to an employee to manage.



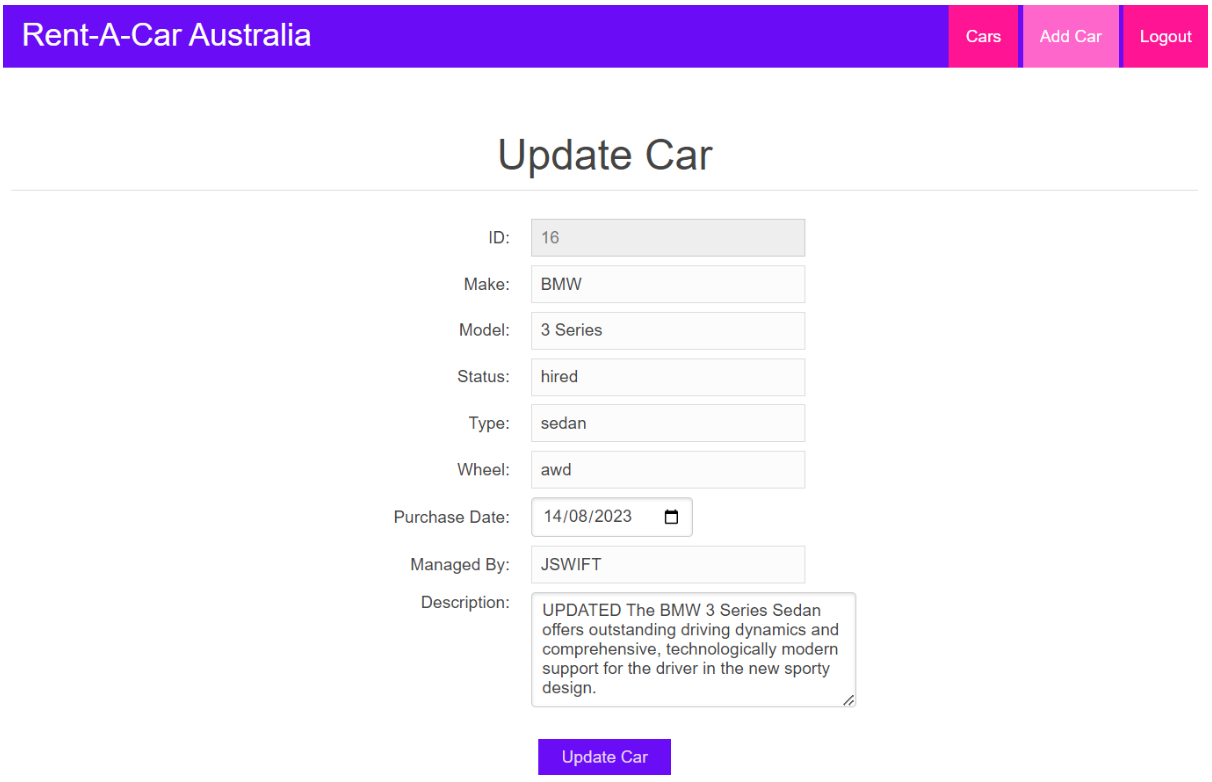
New Car

Make:	BMW
Model:	3 Series
Type:	sedan
Wheel:	2wd
Purchase Date:	28/08/2023
Description:	The BMW 3 Series Sedan offers outstanding driving dynamics and comprehensive, technologically modern support for the driver in the new sporty design.

Figure 4 – Adding a Car

Updating a Car

Users can also update a car by modifying data in the car details screen as shown in Figure 5, and clicking on Update Car button. You can access this update screen by clicking on a car from the list of cars in the Cars tab.



ID:	16
Make:	BMW
Model:	3 Series
Status:	hired
Type:	sedan
Wheel:	awd
Purchase Date:	14/08/2023
Managed By:	JSWIFT
Description:	UPDATED The BMW 3 Series Sedan offers outstanding driving dynamics and comprehensive, technologically modern support for the driver in the new sporty design.

Figure 5 – Updating a Car

Database Interaction Code

The files that are needed for the Python version of assignment are as follows:

1. **`RACASchema.sql:`** a file which contains SQL statements you need to run to create and initialise the RACA system database, before starting the application https://canvas.sydney.edu.au/files/33303631/download?download_frd=1
2. **`Assignment2_PythonSkeleton.zip`**: a zip file encapsulating the Python project for the RACA system https://canvas.sydney.edu.au/files/33381402/download?download_frd=1

To inspect the RACA system code, you need to unzip the ZIP archive first, which will create a folder that includes the name **`Assignment2_PythonSkeleton.`** If you experience any difficulties installing and exploring the project, ask your tutor or lecturer for assistance.

The skeleton code uses a number of Python modules to implement a simple browser-based GUI for the RACA system. The main modules are the Flask framework for the GUI and the psycopg2 module for the PostgreSQL database access. Similar to tutorial 6, **you will need to install the Psycopg2 module and the Flask module**. The skeleton code follows the structure described below:

- The main program starts in the **`main.py`** file. You need to use the correct username/password details as specified in tutorial 6, and then implement the missing database access functions – including any necessary SQL code statements required – in the data layer **`database.py`**.
- The presentation layer is done via a simple HTML interface that can be accessed from a web browser.

The corresponding page templates are located in the **`templates/`** subdirectory, their CSS style file is **`static/css`**.

- The transition between the different GUI pages and the initialisation of the Flask framework is done in the **`routes.py`** file. It currently just invokes the pages, but there is no further business logic implemented yet.

You can run the code by running "**`python main.py`**". This starts a local web server and prints out some debug messages in the terminal; the GUI can then be accessed with any web browser on the same computer via the local URL `http://127.0.0.1:5000/` (If that doesn't work you can also try `http://0.0.0.0:5000/`). Please note that, to terminate the application, you will need to stop the local web server which is running in the background.

Task 2: Functions Implementation

Core Functionality

In this assignment, you are provided with a Python skeleton project that must serve as the starting point for your assignment. Your task is to provide a complete implementation for the file **`database.py`**, as well as make any modifications necessary to the database schema (i.e., **`RACASchema.sql`**). Specifically, you need to modify and complete these five functions:

1. **`checkEmployeeCredentials`** (for login)
2. **`findCarsByEmployee`** (for viewing cars list)
3. **`findCarsByCriteria`** (for finding cars)
4. **`addCar`** (for adding car)
5. **`updateCar`** (for updating car)

Note that, for each function, the corresponding action and outcome should be implemented by issuing SQL queries to the database management system. If you directly output the result set, pre-process, manipulate and/or make changes to the input or output datasets using Python code i.e. without issuing SQL queries, you are considered as cheating, and you will get penalised or zero point for the assignment. No additional Python modules or libraries should be imported.

Marking

This assignment is worth 16% of your final grade for the unit of study. Your group's submission will be marked in line with the rubric that follows.

Group member participation

If members of your group do not contribute sufficiently, you should alert the unit coordinator as soon as possible. The course instructor has the discretion to scale the group's mark for each member as follows:

Percentage of contribution	Proportion of final grade received
< 5% contribution	0%
5 - 10% contribution	20%
11 - 15% contribution	40%

16 - 20% contribution	50%
21 - 24% contribution	60%
25 - 28% contribution	80%
29 - 30% contribution	90%
> 30% contribution	100%

Note: The above table assumes that each group will have 3 members, so, on average, around 33% contribution is expected from each member of the group. In special case, if the group has less than 3 members then the contribution percentage will be adjusted accordingly. You must justify your contribution percentage by providing detailed explanation of your contribution in the coversheet mentioned before. You should also maintain a diary of your group meetings and discussions on Canvas. Furthermore, we will run random face-to-face interviews to understand your contribution, if needed.

Marking Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of 16 points.

	Part Marks (0 – 1.5 pts)	**Full Marks (2 – 2.5 pts)**
Login	Can correctly login the user 'JSWIFT' and validate his username and password.	All valid users can be logged in successfully, and unsuccessful user logins should be rejected.
View Cars List	Correctly list all cars associated with user 'jswift' in the correct order (see Figure 2).	Correctly list all cars associated with a user, in the correct order, for all possible username input from Figure 1.
Find Car	Correctly list cars for keyword "new" (see Figure 3).	Correctly list cars for all possible keywords.
Add Car	Can correctly add a car to the database.	Can correctly add all valid cars to the database. Cars entered with invalid details should be rejected.
Update Car	Can correctly update the status of a car as shown in Figure 5.	Can correctly update details of all cars, ensuring the updated details for a car are valid.
Stored Procedure	A couple of stored procedures (functions) are correctly created in the submitted SQL file.	A couple of stored procedures (functions) are correctly created in the submitted SQL file, and correctly called in two of the five specified functions.

	No Marks (0 pt)	**Full Marks (1 pt)**
Record Keeping of Group Discussion	One or more issues reported or found with group member participation or maintaining records of group discussions.	No issue reported or found with group member participation. All group members participate and regularly maintain a diary of group meetings and discussions on Canvas.