SOMANATHAN SUBRAMANIYAN

STUDENT ID: G00364742

Data Representation

Big Project

2019

Table of Contents

[Q1: Big Project 1](#_Toc27314474)

[Q1.1. About the project 1](#_Toc27314475)

[Q2: Technical Details 1](#_Toc27314476)

[Q3: Results 2](#_Toc27314477)

[Q4: Software Requirements 4](#_Toc27314478)

# Q1: Big Project

## Q1.1. About the project

The “Vehicle Accident Register” website allows the authenticated (google) users to retrieve, create, update and delete accidents from the Accidents database, which is hosted on the “PythonAnywhere” cloud services.

# Q2: Technical Details

The “Vehicle Accident Register” source code is managed through GIT repository

**GIT Repository:** <https://github.com/SomanathanSubramaniyan/project-pythonanywhere.git>

**Web-Application**: <http://somanathan.eu.pythonanywhere.com/index.html>

**Database host address**: somanathan.mysql.eu.pythonanywhere-services.com

**Source Code details**:

Accidents.py : REST API source which get, post, update and delete details from database

AccidentsDAO.py : python CLASS file to manage database connections

Config.py : Configuration file which contains the Database connection details

Index.html : HTML page which captures the vehicle details and consumes RESTAPI

**Database Schema:**

CREATE TABLE Accidents (

  id INT NOT NULL AUTO\_INCREMENT,

  province varchar(20) NOT NULL,

  VehicleType varchar(20) NOT NULL,

  DriverAge varchar(20) NOT NULL,

  DriverSex varchar(20) NOT NULL,

  MonthYear varchar(20) NOT NULL,

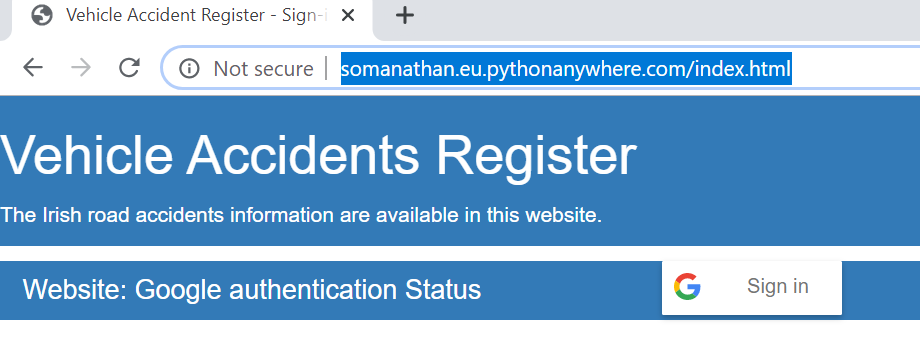
  primary key(id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

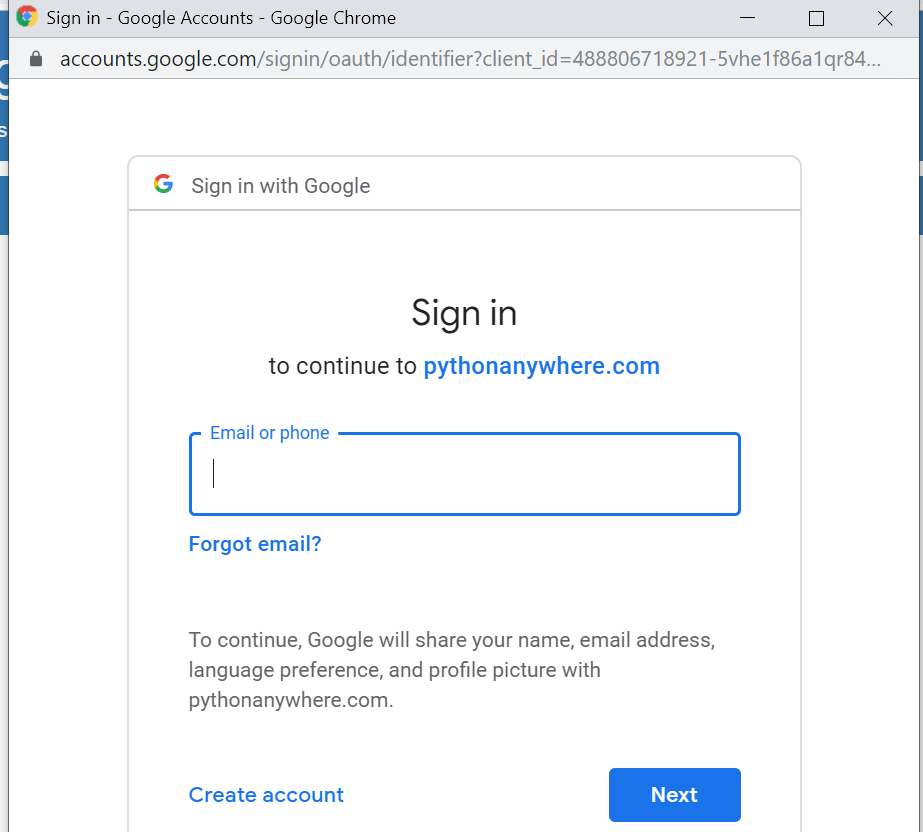
# Q3: Results

The browser used for the testing is google Chrome

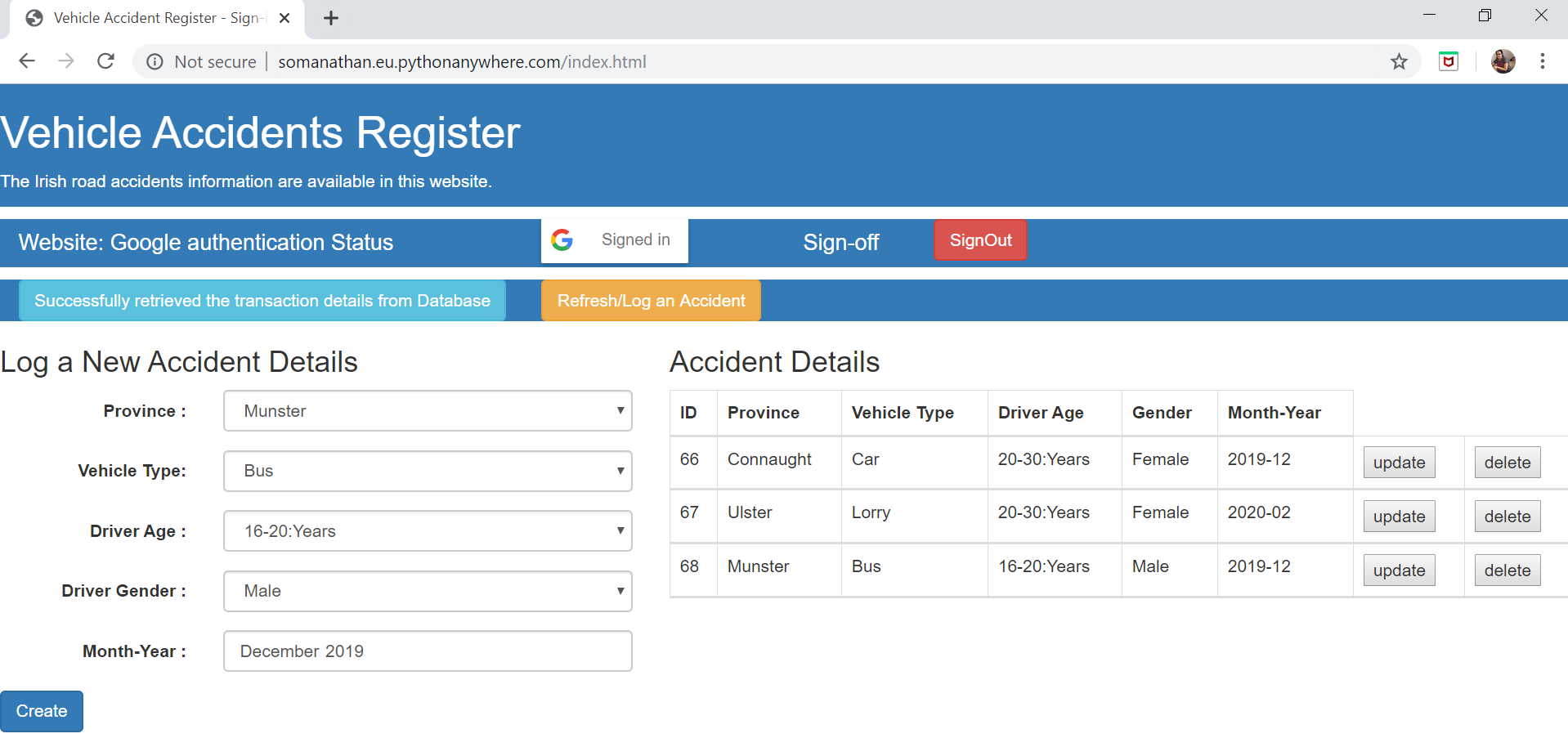
Step1: <http://somanathan.eu.pythonanywhere.com/index.html>



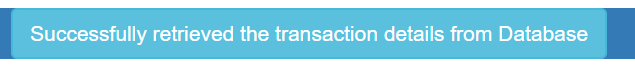
**Step2:** Sign-in using your google ID. On sign-in the accident details, will be displayed



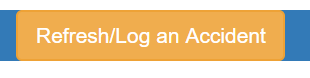
**Step3**: On successful google login, the accident details are shown as per below



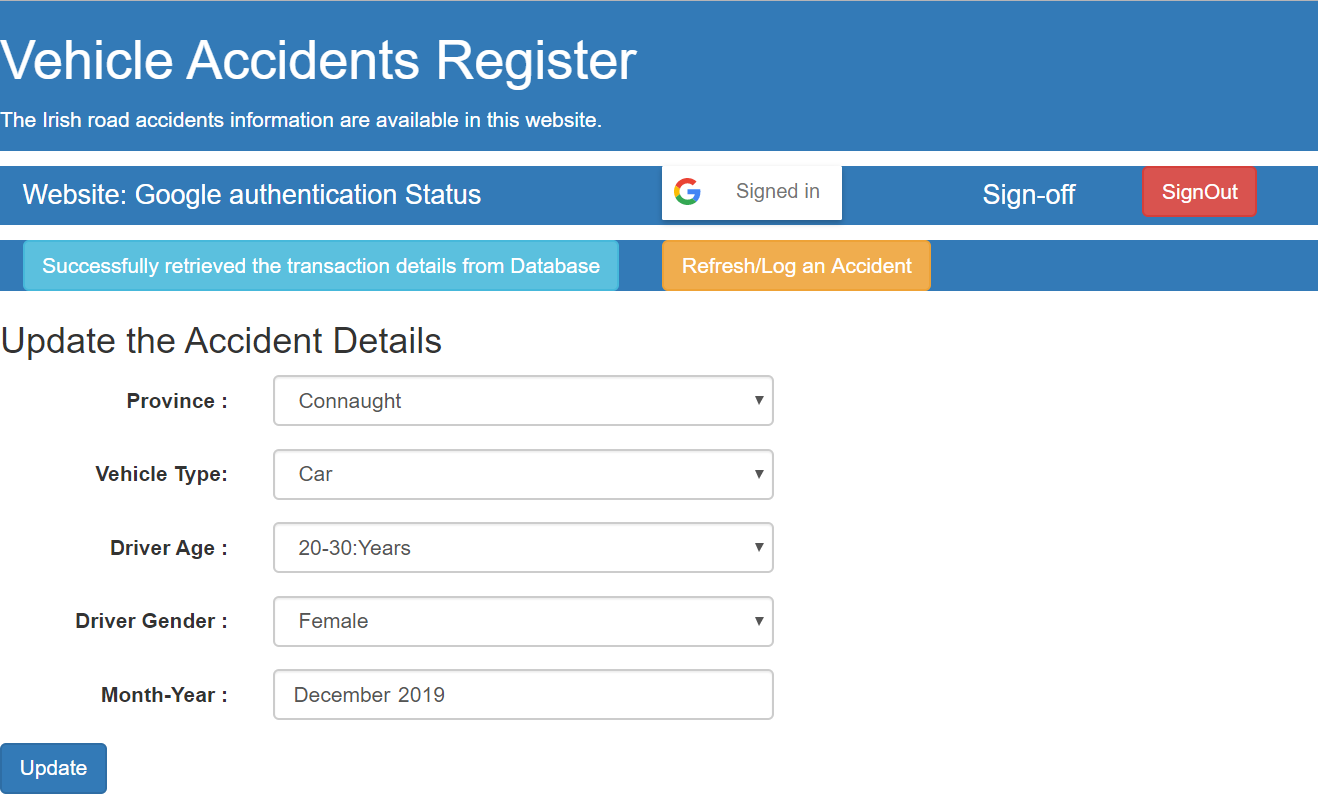
**Step4**: The Last successful API status is displayed in the below bootstrap styled button



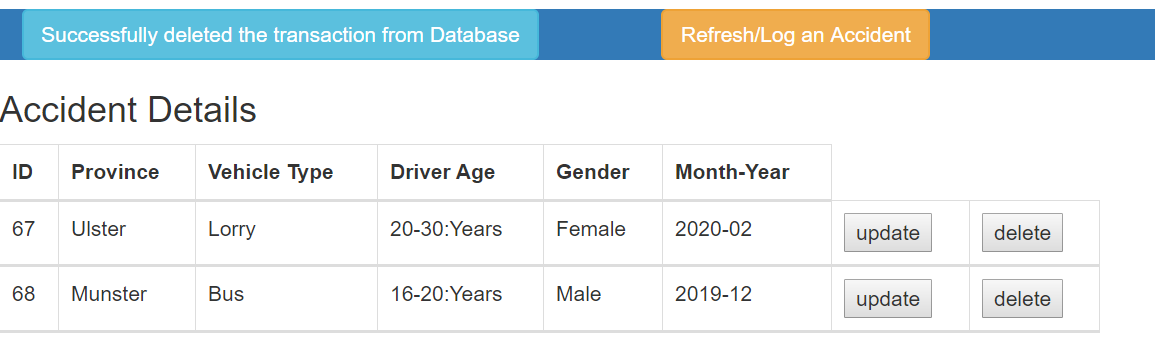
**Step5**: The webpage can be reloaded by clicking the below button



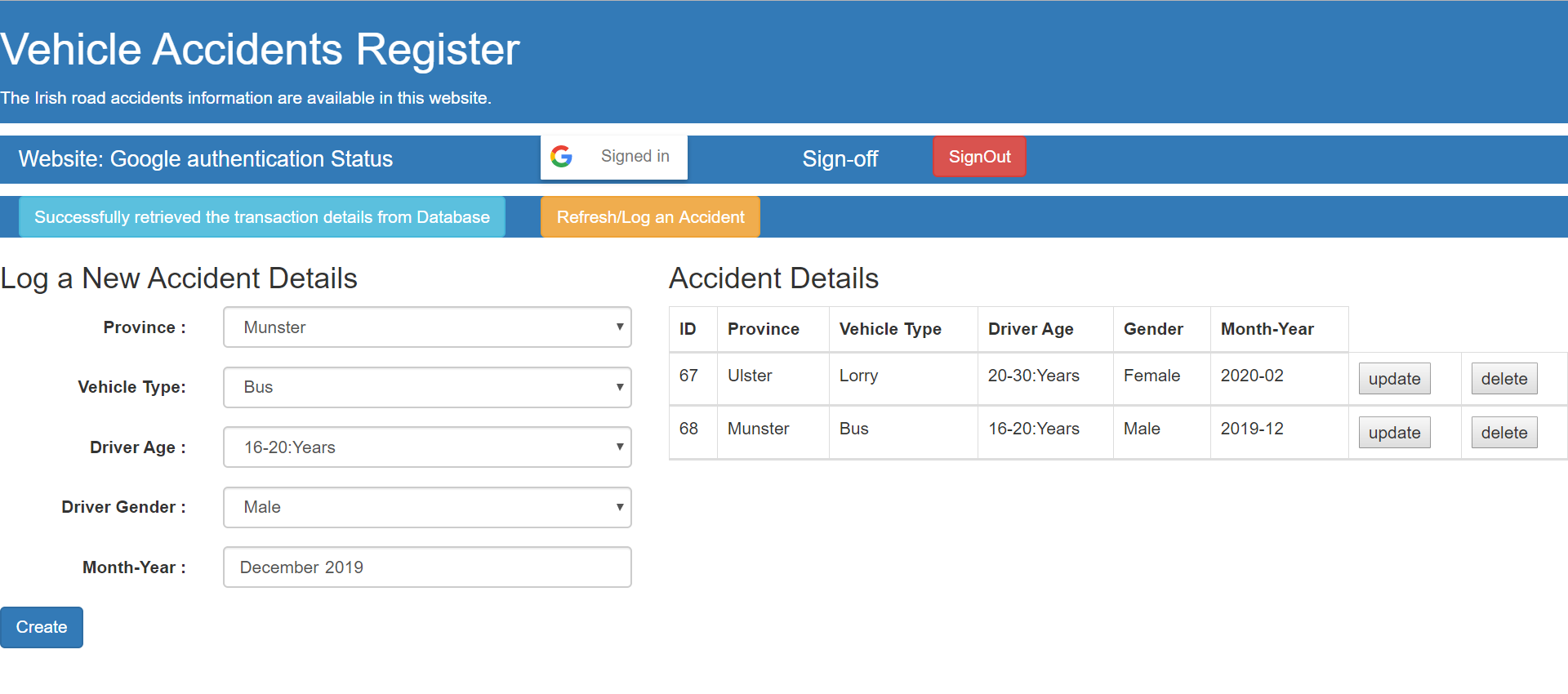
**Step6**: Update Vehicle details; On clicking the individual row in the vehicle detail screen; the details are available for modification in the Accidents form.



**Step6**: Delete Vehicle details; On clicking the individual row in the vehicle detail screen; the individual row/record will be deleted from the database. Status will be displayed in the “status” button



**Step7**: On clicking the refresh/log an incident button; the webpage will be loaded again, and form will be available to log the new incident.



# Q4: Software Requirements

astroid==2.3.3

Click==7.0

colorama==0.4.3

Flask==1.1.1

Flask-Cors==3.0.8

isort==4.3.21

itsdangerous==1.1.0

Jinja2==2.10.3

lazy-object-proxy==1.4.3

MarkupSafe==1.1.1

mccabe==0.6.1

mysql-connector==2.2.9

mysql-connector-python==8.0.18

numpy==1.17.4

pandas==0.25.3

protobuf==3.11.1

pylint==2.4.4

python-dateutil==2.8.1

pytz==2019.3

six==1.13.0

typed-ast==1.4.0

Werkzeug==0.16.0

wrapt==1.11.2