

PYTHON PROGRAMMING

Project 1

|  |  |  |  |
| --- | --- | --- | --- |
| Faculty | Information Technology | | |
| Module Name | Python Programming | Module Code | ITPYA0 |
| Project Number | 1 | Copy Editor | Ms Nicole Stern |
| Student Name | Sopumelela | Student Surname | Sandekela |
| Student Number | CON-1475940-C5L6 | Date Submitted | 09/02/2024 |
| Marker Name |  | Date Marked |  |
| Mark | /150 | Percentage | % |
| Moderator Name |  | Date Moderated |  |
| Mark | /150 | Percentage | % |

|  |  |  |  |
| --- | --- | --- | --- |
| Question Number | | Total | Mark Achieved |
| Question 1 | | 150 |  |
| Lecturer |  | Total Mark |  |

# ITPYA0 – **Project 1** – PYTHON PROGRAMMING 2024 | V3.0 Page 1 of 3

Eduvos (Pty) Ltd. (formerly Pearson Institute of Higher Education) is registered with the Department of Higher Education and Training as a private

Question 1 150 Marks

Deduct marks where applicable.

Refer to the model answer in the memorandum.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirements | | Possible | | Achieved |
| Program Design | Source code printed out | 5 | 10 |  |
| Documentation printed out | 5 |
| Source Code | Naming standard followed | 2 |  |  |
| Indentation followed | 5 | 17 |  |
| Prologue at top of each file | 5 |  |
| Comments provided | 5 |  |  |
| Documentation | Index page | 3 |  |  |
| Description of application | 3 |  |  |
| Installation and setup | 3 |  |  |
| Directions on how to use the application | 3 | 18 |  |
| Directions on how to handle error  messages | 3 |  |  |
| Screenshots of the entire application | 3 |  |  |
| Design | How well does the application interact with the user? Good screen layout (including use of components and  events), user-friendly, clear instructions | 5 | |  |
| Database | MySQL database correctly created | 5 | 15 |  |
| Tables correctly created | 2 |  |
| Columns contain the correct types | 3 |  |
| Statements to populate and manipulate  the database correctly created | 5 |  |
| Registration | Registration function correctly | 5 | 22 |  |
| Indication of required fields | 2 |  |
| Validation of form functions correctly | 5 |  |
| The client retrieves data from the  database using the server | 10 |  |
| Application | Check to ensure the user exists | 5 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Check to ensure a video exists | 5 | 63 |  |
| New customers are added correctly | 5 |  |
| New videos are added correctly | 5 |  |
| The custId is incremented correctly | 5 |  |
| The videoId is incremented correctly | 5 |  |
| The videoVer is incremented correctly | 5 |  |
| Videos are marked as hired out | 5 |  |
| Videos are returned correctly | 5 |  |
| System’s current date is used | 5 |  |
| Outstanding movies marked within the  hire table | 5 |  |
| dateReturn is **server** system’s current  date | 5 |  |
| Try and catch statements are used | 3 |  |  |
| Penalties | Project returned and resubmitted (-10%  on next submission) | -10% | |  |
| Project resubmitted due to plagiarism  (−20% on next submission) | -20% | |  |
| Program causes unhandled errors  during execution or user input | -10% | |  |
| Program does not compile | -10% | |  |
| Redundant code | -10% | |  |
| Total | | 150 | |  |

Table of Contents

Description Of Application5

Table: customers 5

Table: video 5

Table: hire 5

Installation And Setup6

Unzip the folder to desired file directory/path. 6

Create an empty schema/database6

Run the Setup\_Database.py file 6

How To Use7

Choosing the "Register Customer" 7

Choosing the "Register Movie" 7

Choosing the "Hire Out Movie" 8

Choosing the "Return Movie" 8

How To Handle Error Messages9

Mistyped your number while registering 9

Special cases of when registering movie 9

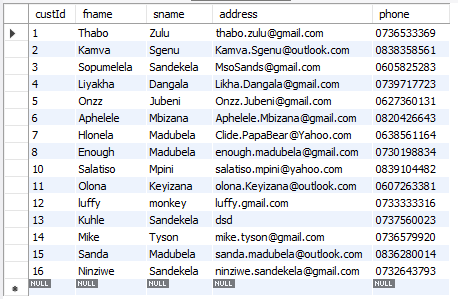
Validating for NULL values 9

## Description of Application

This application is called Video Store, it is developed to allow the business to keep record of all their customers as well as their videos. To achieve this the application is connected with a DBMS (Database Management System). For this application MySQL was used as the DBMS.

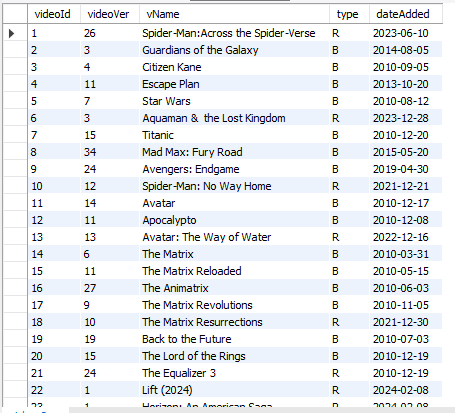
Each customer must register before they can hire out videos. This information is stored in a table called customers.

Table: customers



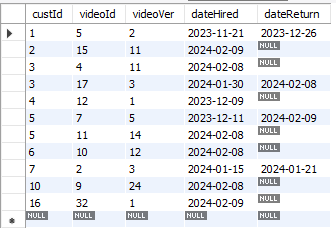
All available videos that customers can hire must be stored in a table called videos.

Table: video



The third table is added to keep the record of all transactions. This table stores all videos that have been hired out.

Table: hire



## Installation and setup

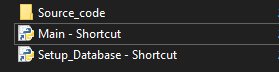
To install and set up the application properly.

* Unzip the folder to desired file directory/path.



* After accessing the folder, the important folders / files to notice are **Source\_code** folder,

**Setup\_Database.py** and **Main.py** shortcut files from the files inside the Source code folder.



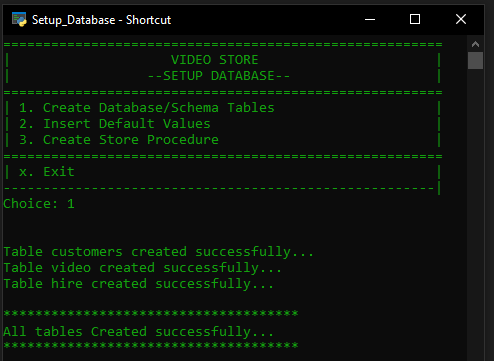
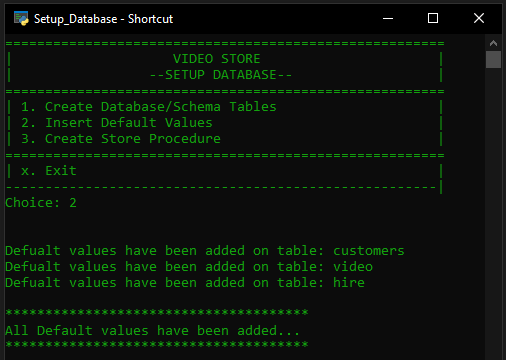
* Next step is creating an empty schema / database on MySQL workbench or MySQL cmd Client

After logging in to one of these interfaces you can create a empty schema/database by:

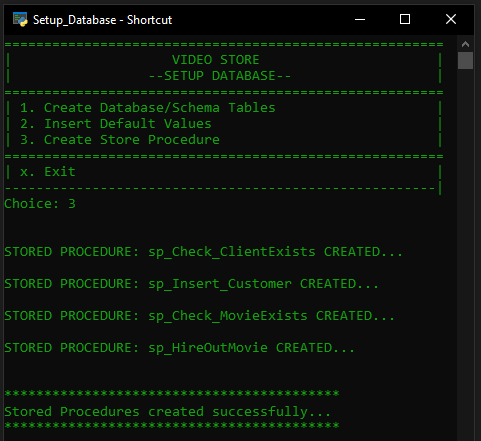
|  |  |
| --- | --- |
| MySQL command line Client | MySQL Work bench |
| Application appearance | |
|  |  |
| Creating Schema / Database command | |
|  |  |
| Run command | |
| Press “ENTER” | Click on the lightning icon. |

* Next step is setting up the database, run the Setup\_Database.py file and choose every option one by one.

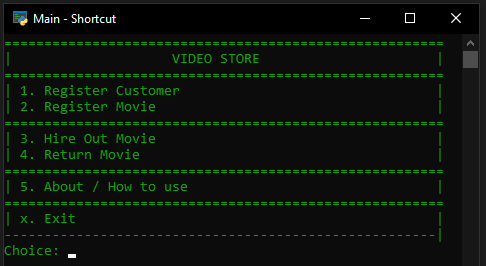
Option 1 results: Option 2 results:

Option 3 results:



## How To Use

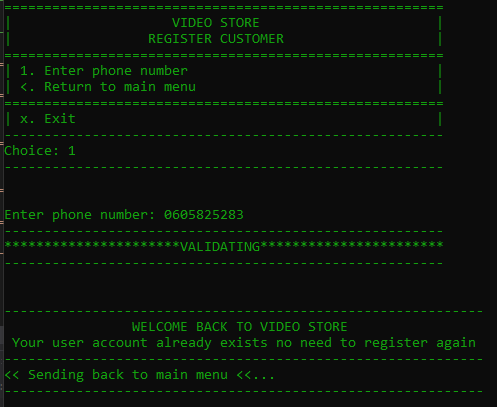
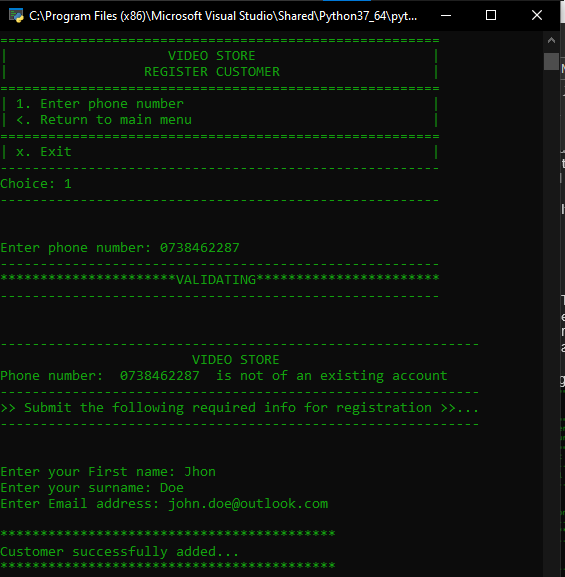


This Video Store application is basically used for hiring out movies, returning movies and registering movies.

To use the application the user needs to be registered using the application. Using this application is straight forward as it is user friendly just follow the instructions.

* If you are a new user:
  + Before choosing any other options besides "About" you should be registered into the system, so first step is choosing the "Register Customer" option. (Remember: respond with option number e.g(“1”) instead of writing "Register Customer".
* To register, as the customer enter your phone number then the program will check whether you exist on the database. If you already exist a message reflecting that will appear. If the customer is not contained within the database, the user must enter the customer’s name, surname, and address.

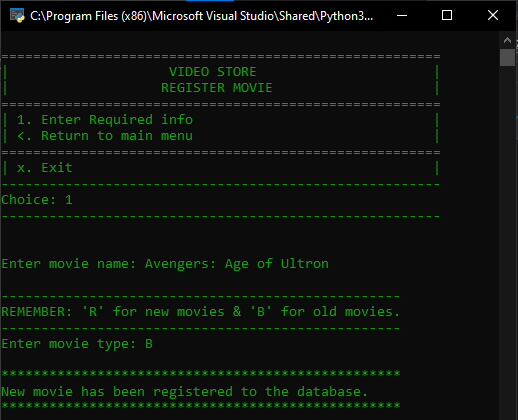
Existing customers on database: Not existing on database:

* When the user chooses the **Register Movie** option, then the user must enter the movie name and type.

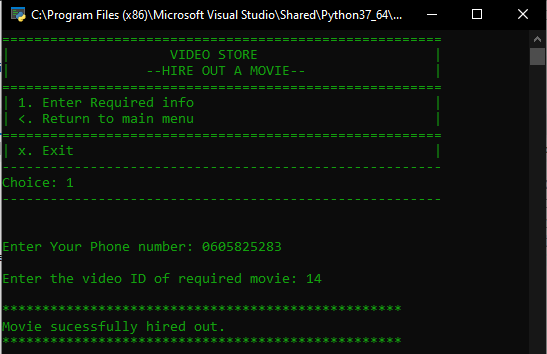
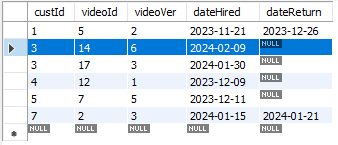
RE: The movie can only be one of two types:

* **Red box** – new movies, this is indicated by a ‘R’
* **Black box** – old movies, this is indicated by a ‘B’



* When the **Hire Out Movie** option is chosen, the application then prompts for the customer’s phone number. The user then enters the videoId of the requested movie.

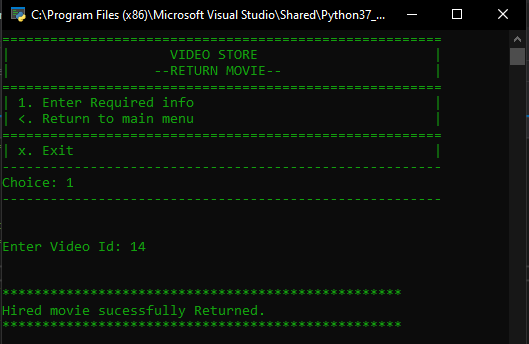
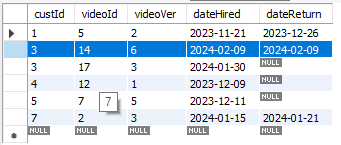
Movie gets added to the hire table:

* The **Return Movie** option should request that the user enters the videoId.

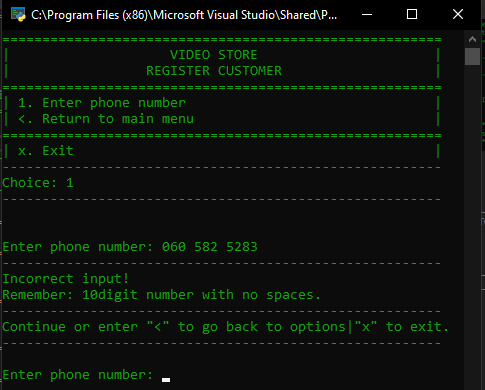
(in example lets return the same hired movie)

The date return is updated to indicate:

## How To Handle error messages

To handle error messages is simple. Let’s say you mistyped your number while registering, an error message pops out reminding you your input should be in range of 10digit number and with no spaces in between the gives you another chance to retry, go back to main menu or exit the program.

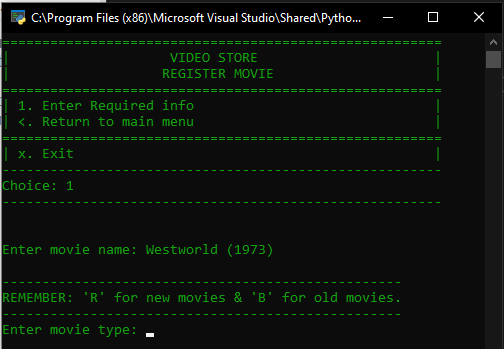


In special cases of when registering movie there is movie types distinguished with:

“R” = For new movies

“B” = For old movies

A message pops up before the user input the movie type reminding him/her the required info.



* Validating for NULL values: If user enters nothing the program releases error message and offer retrying input.

