ITRW 324

vERSION cONTROL WEBSITE

27108945

Mabwe BP

Table of Contents

[1.1 LIST OF FIGURES 2](#_Toc530369378)

[2 Project overview 3](#_Toc530369379)

[3 Description of MVC created in codebase 3](#_Toc530369380)

[3.1 Models 3](#_Toc530369381)

[3.2 Views 3](#_Toc530369382)

[3.3 Controllers 4](#_Toc530369383)

[4 Home Page 4](#_Toc530369384)

[5 Private Repository 5](#_Toc530369385)

[6 Public Repository 6](#_Toc530369386)

[REFERENCE LIST 7](#_Toc530369387)

## LIST OF FIGURES

[Figure 1- MVC (http://3.bp.blogspot.com/-KyWNL6EHcic/UUXzK3ZXRqI/AAAAAAAAGrE/RaXXJdEmpOw/s1600/Slide2.PNG ) 3](#_Toc530370877)

[Figure 2- Home Page 4](#_Toc530370878)

[Figure 4 - Public Repository form 6](#_Toc530370879)

# Project overview

The objective of this project was to develop a Version Control website for documents whilst adhering to the Model, View and Controller framework which is basically in summary a helpful way of designing interactive systems. The main purpose of each component is illustrated below by Figure.

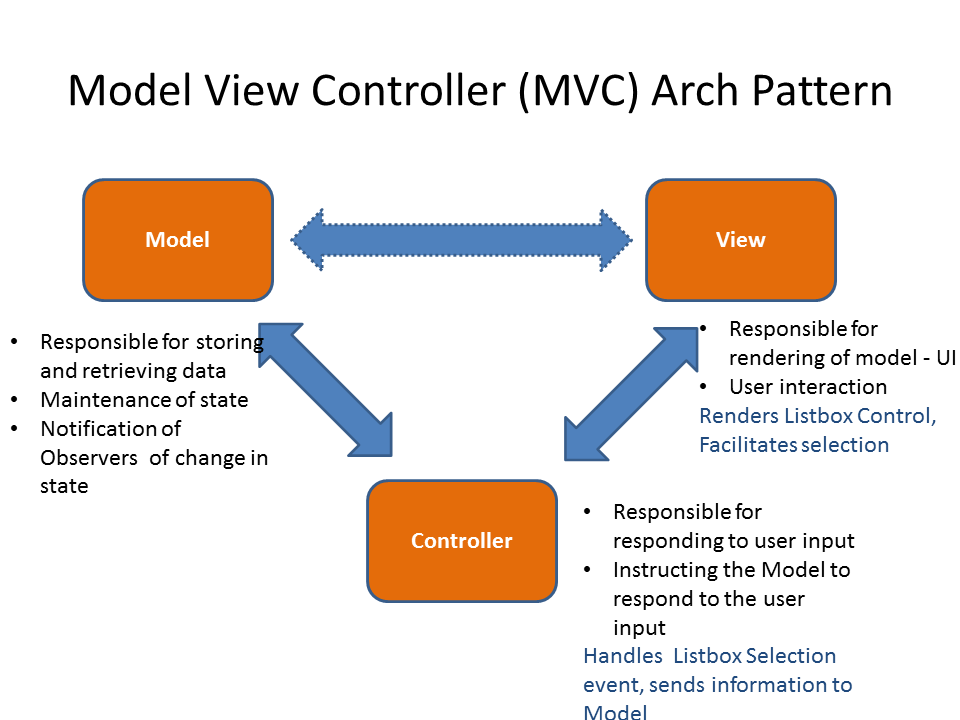


Figure 1- MVC (<http://3.bp.blogspot.com/-KyWNL6EHcic/UUXzK3ZXRqI/AAAAAAAAGrE/RaXXJdEmpOw/s1600/Slide2.PNG> )

# Description of MVC created in codebase

## Models

I decided to use a SQLite database in my project due to its easy set-up procedure and its compatibility with the program I used to develop the system –Web2py mainly developed by Massimo DiPierro. The database has various tables that are created upon installation but I mainly used db.auth\_user, db.auth\_group, db.Uploaded\_Documents and db.auth\_membership. The author tables are mainly associated with the user and attributes relating to the uploads made by the user.

Furthermore, I implemented and modified the Menu model which basically is the main application menu and maintains a consistent responsive state menu with various links to controllers and views such as the private\_doc and public\_doc views.

## Views

Web2py contains various built-in helpers with regards to views and I mostly used the BEAUTIFY component which returns a presentable XML-like object. I have a default/index html view form which basically displays the main page thus linking up with the menu model. Furthermore, I have the public\_doc view which displays all files in the public repository as well as the uploader and the time the file was uploaded.

## Controllers

The default.py controller was mainly used and this is where I implemented various def. functions and database queries as well as returning certain variables which were then sent to the views component, formatted then displayed to user although in this project I mainly used web2py’s built in helpers which can be modified extensively to meet an individual’s needs. The default controller in the project also contains various “restrictions” such as auth.requires\_login () which basically requires that the user be logged in prior viewing and uploading files.

# Home Page

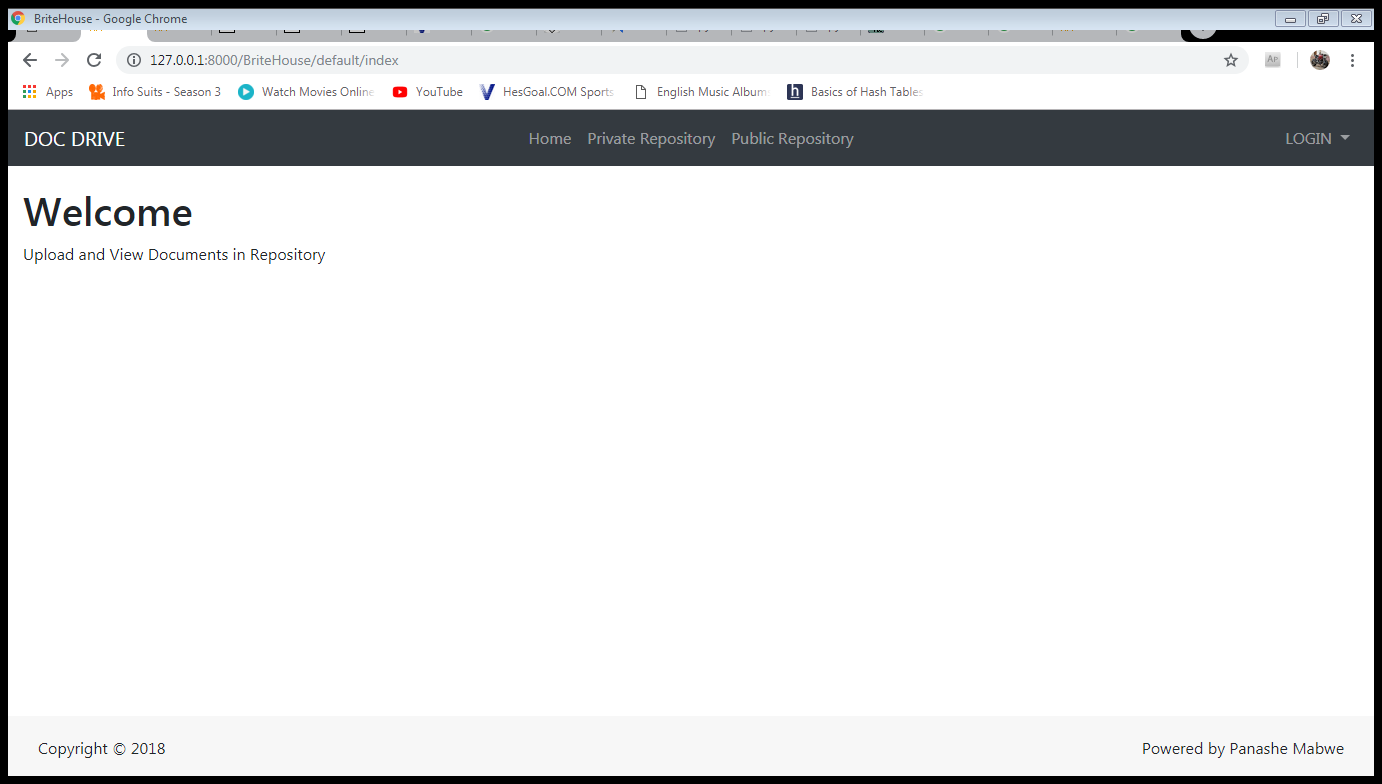


Figure 2- Home Page

# Private Repository

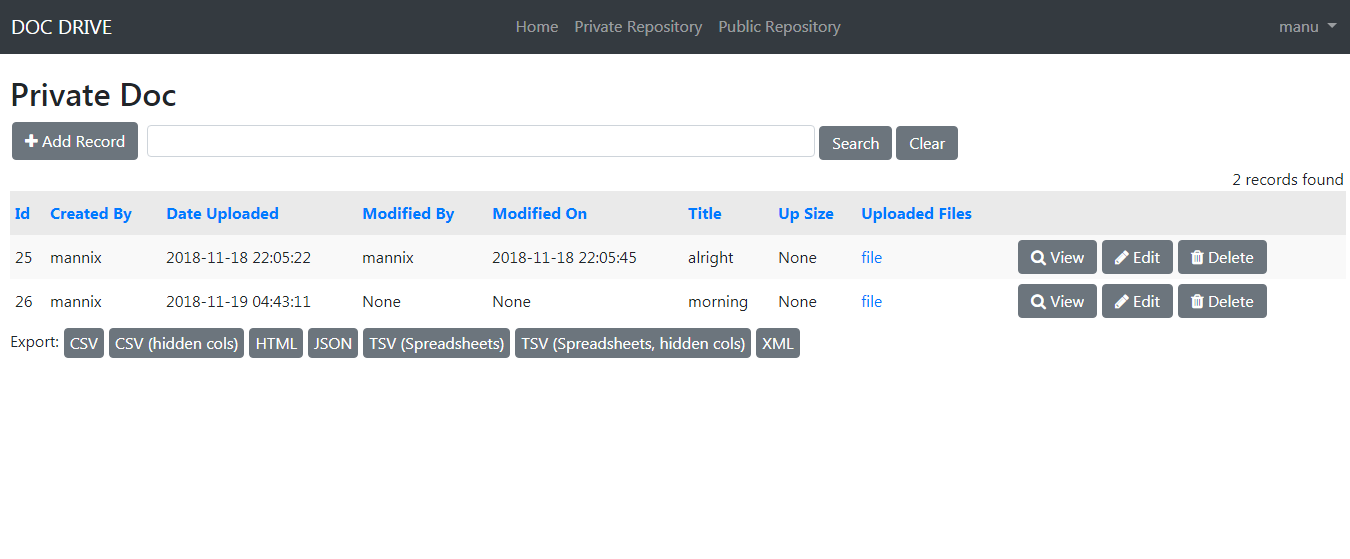


Figure 3 - Private Repository form

# Public Repository

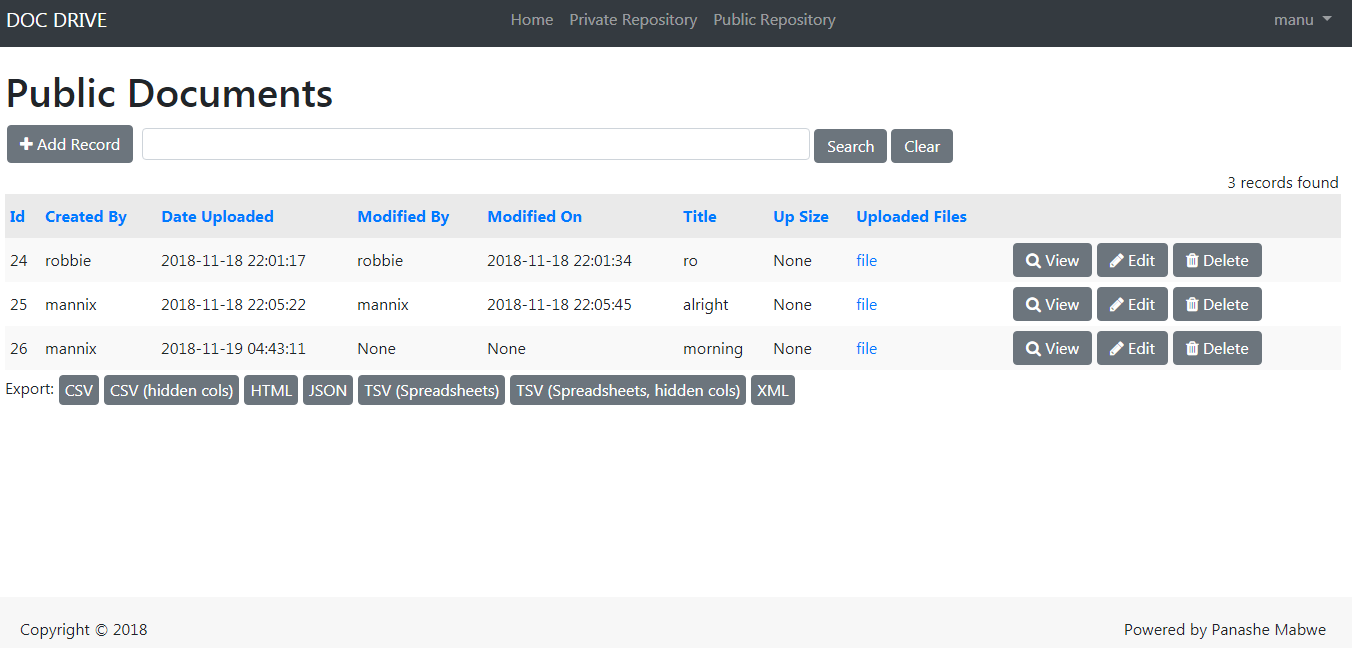


Figure 4 - Public Repository form

# Database tables

# 

# REFERENCE LIST