

**WM7005 Assessment Cover Sheet**

* Complete and attach this cover sheet to your assessment before submitting

**Course No. and Title: Assessment Title:**

*Database-Driven – WEBSITE APPLICATION - Individual Project*

*WM7005-Web Programming 1*

**Learning Outcomes:**

* [LO1] Create server-side scripts that add advanced backend database-driven systems for websites on the most widely used Web servers.
* [LO2] Plan, organise and implement programming processes following best practices of professional ethics and industry standards in diagram design, programming and documentation conventions.

**Student ID:**

**Student Name:**

**Tutor:**

**Due Date:** Moodle: **15th December 2022, by 11:55pm**

**Date submitted:**

**Late Rule:**

***Late submissions will be graded according to the rules outlined on p.2***

## By submitting this assessment for marking, either electronically or as hard copy, I confirm the following:

* This assignment is **my own work.**
* Any information used has been properly referenced.
* I understand that a copy of my work may be used for moderation.
* I have kept a copy of this assignment.

Do not write below this line. For Polytechnic use only*.*

**Assessor:**

**Date of Marking:**

**Grade:**

**Comments:**

BACHELOR OF WEB MEDIA

WM7005 WEB PROGRAMMING 1

PHP AND MYSQL WEB APPLICATION

Website demonstration video due on Moodle: **20th December 2022, by 11:55pm**

Website Application due on Moodle: **20th December 2022, by 11:55pm**

Grade Value: 30% of the Course (Must Pass: no)

# LATE SUBMISSIONS RULE

**Late submissions are subject to the following rules:**

* 1. If an assessment is submitted late the maximum result the student can achieve is 60% in that assessment unless decided otherwise by the Head of School or delegate
  2. The cut off time for submitting an assessment will be 3 working days after the assessment is due. A student submitting after 3 working days will get 0%
  3. Extensions are not available for controlled assessments or other forms of assessment where scheduling concerns make it impracticable (e.g. workplace assessments).
  4. On application, course coordinators may approve extensions to deadlines for other forms of assessment, (such as projects and assignments,) up to a maximum of 2 working days. Applications must be made before the stated deadline.
  5. If a student misses an exam and is able to produce a medical report or certificate then the student is allowed to retake the assessment at another time. If the student misses a mid semester exam they would be allowed to retake at the end of the semester, if a student misses the exam at the end of the semester then the students grade would be put on hold and they would retake the assessment at the beginning of the next semester and it would be processed by Programme Committee, Faculty Board and Academic Board then.

# INTRODUCTION

Task: to create a database-driven website that uses PHP and MySQL database. through implementing the identified scenario and completing the required tasks

Scenario: you will create a simple movie booking system admin interface that will allow the system administrator to manage the required system data. The system administrator will perform the basic CRUD operations to manage the Cinemas, Movies and Shows data.

# Creating the database

Diagram

Description automatically generated

1. Use phpMyAdmin to create a database called “Poly\_Cinema”.
2. The database should match the structure in the above picture.
3. You are required to identify the relationships type and the required fields.
4. Create the required tables to manage the system’s Cinemas’, Movies’, and Shows’ data.
5. To eliminate many to many relationships, consider that each movie can only belong to one Cinema.
6. Each table must at least have 4 logical columns.
7. Each table must at least have two different constraints other than the PK and FK constraints (consider scenarios where for instance the Cinema details should only be added once and duplicated data should not exist).
8. You can add additional tables if needed considering the logical structure of cinemas from your experience.
9. You are required to submit a txt document that includes all your database creation code.

Note: Before building your webpage, it is recommended to perform the CRUD operations using phpMyAdmin to ensure the database is correctly constructed and to avoid error in your PHP pages.

# Creating the Website

* Create a signup page where system administrators will be able to register to your website
* After signing in, redirect the users to the login page.
* After validating the login credentials, store the required details in PHP session variables
* Users should be able to perform CRUD operations on Cinemas, Movies and Shows
* The website should have a page that displays all movies for all cinemas
* The website should have a page that displays all movies per cinema
* You should have a page that displays all shows for all movies per cinema
* Each movie page should have a functionality that displays all shows for all cinemas
* The website should enable the system administrator to add new shows, update and delete shows
* The website should enable the system administrator to add new cinemas, view all cinemas, view one cinema, update and delete cinemas
* The website should enable the system administrator to add new movies, view all movies, view one movie, update and delete movies
* All movies should have images and the system administrator must be able to upload image files
* Use proper navigation across all pages
* All forms must be validated
* All pages must be coded as PHP templates with product details loaded dynamically from a back-end database
* Your code should include comments for every substantial block of code, explaining the role of each function.
* Your website design must follow User Experience standards. You have the option to use CSS from scratch or to use bootstrap templates (must be customizable).
* If bootstrap templates were used, the student must submit a word document that includes the source of the original template, and the changes made to customize it.
* If the templates was not customized efficiently, it will affect the design effort considered for this project.
* In the view pages, the users should have the option, to update or delete data
* In the update pages, the users should have the option to delete or view data
* In the movies view pages whether it is viewing all movies or view one specific movie details, the user should have the option to add or view movies shows.
* Users should also be able to view their profiles where their details are presented and should have the option to update or delete their accounts.
* Pre-populate your database:
  + At least 5 users
  + At least 3 cinemas
  + At least 5 movies per cinema
  + At least 3 shows per movie
* Record a video where you demonstrate your website functionalities.

# PHP functionalities summary

|  |  |
| --- | --- |
| User Profile | Create(signup), Login, Update, Delete and View |
| Movies | Create, Update, Delete and View pages |
| Cinema | Create, Update, Delete and View pages |
| Shows | Create, Update, Delete and View pages |

# LEARNING OUTCOMES ASSESSED

* [LO1] Create server-side scripts that add advanced backend database-driven systems for websites on the most widely used Web servers.
* [LO2] Plan, organise and implement programming processes following best practices of professional ethics and industry standards in diagram design, programming and documentation conventions.

# REQUIREMENTS

This assignment will be marked according to the proper working of PHP, MySQL, JavaScript (if used), HTML, and CSS, as well as the quality, suitability, and **originality of the website’s graphic design and usability.**

**Your submission should consist of the zipped project files, MySQL commands file, a recorded video.**

The recorded video should demonstrate all working website functionalities (for instance, access the signup page, fill the fields, and show the results). Please ensure that the recorded video is in the following format (MP3, VLP , MP4, VLC)

You must upload the site files and documents to Moodle as a single compressed file (.zip). The file should be named as: studentID\_studentName\_polyCinema

# MARKING

|  |  |
| --- | --- |
| Section | Weighting |
| Database Implementation   * Creating tables * Using suitable datatypes * Constraints * database pre-population | 20% |
| Quality and originality of the graphic design (use of colors, fonts, images in making the product as appealing as possible) | 20% |
| Backend coding | 60% |
| Total | 100 |