

## IS218 Project #3: MVC Pattern

### Overview:

This project builds off of what was done in Projects 1&2 and expects that you have all the requirements for Projects 1&2 already completed.

Take the existing application you created in Project 2 and rewrite it to follow the Model-View-Controller (MVC) Design pattern. You should have separate files for the controller, models, and views. Controller should be a single index.php file, while you should have several model for each database table, and several view files for each webpage. Use the guitar\_shop application from week 9 in moodle as an example.

Additionally, you should add the ability to both edit and delete and existing question that the user has in their database. Buttons should be present on each questions list item on the display questions page.

Edit should take user to a form much like New Question (can even use same HTML template) upon changing the data and submitting the form, the user is brought back to their list of questions where the new data should be reflected.

Delete should just be a button that removes the current question and refreshes the page to get new data, the deleted question now removed.

### Requirements:

#### 1. Controller File (25 Points)

- a. Contain Actions for everything the user can do within the application. Some actions to support include:
  - i. display\_login
  - ii. login
  - iii. display\_registration
  - iv. register
  - v. display\_questions
  - vi. display\_new\_question
  - vii. create\_new\_question
  - viii. display\_edit\_question
  - ix. edit\_question
  - x. delete\_question
- b. Should only validate submitted data, and transfer control to the model and view when appropriate.

2. Model Files
  - a. Have a *database.php* file that creates a PDO Object **(3 Points)**
  - b. *questions\_db.php* that has functions for: **(10 Points)**
    - i. Get all user's questions
    - ii. Get 1 specific question
    - iii. Create a Question
    - iv. Edit a question
    - v. Delete a question
  - c. *accounts\_db.php* that has functions for: **(7 Points)**
    - i. Getting a user
    - ii. Creating a new user
3. View Files **(15 Points)**
  - a. All existing webpages should be converted into views that follow the MVC pattern.
  - b. Use reusable templates wherever possible (Extra Credit) (EX. Header/footer from guitar shop)
4. Edit Function
  - a. Edit button on each question in the Question list which takes user to a form with all the question data populated **(10 Points)**
  - b. Submitting this form should update the record in the database (using same MVC pattern) **(10 Points)**
  - c. After running Database Query, direct user back to Display Question pages, data should be updated **(5 Points)**
5. Delete Function
  - a. Delete Button on each question in the Question list which should delete the question it's attached to. **(5 Points)**
  - b. Send request to controller a follow MVC pattern **(5 Points)**
  - c. After running database query, refresh Display Question pages to reflect new data. **(5 Points)**

Important:

1. Track changes to your code using GitHub
2. Submit your AFS link and GitHub link to the app on Moodle Both Links are required, if you missed one of them, you get 20% off from full credit.
3. Projects can be submitted late for a 10% penalty a day