PROG6212 Check List

Overview:

- No database (If a database is used for this part, then the script of the DB should be submitted with the documentation)
- No roles/identities (No login or register)
- Use of attributes for validation
- MVC only
- Store in memory / JSON / files / encryption and decryption
- Focus on design / UI
- YouTube video
- Documentation updated

Lecturer Claim View:

1. Layout, colours, user flow, simple and intuitive for lecturers to input their claims.
2. Input for hours worked and hourly rate as well as additional notes.
3. Uploading of documents here as well in lecturer claim.
    a. Uploading of supporting documents should have the following:
        i. Linked to the corresponding claim.
        ii. Securely stored (if you are not using a database, which you do not need to in part 2 - you need to use encryption / decryption).
        iii. Upload button.
        iv. Once uploaded, file name should be displayed on the form.
        v. Need to limit file size here as well as restrict file types to .pdf, .docx, and .xlsx.
            1. Uploading of the document needs appropriate error handling and error messages.
4. Submit button.

Lecturer Viewing Claims:

1. Lecturers can track their claims and see updates of the status as it moves through the approval process.
2. Simple text label or progress bar.
3. As coordinator or manage approves or rejects the claim should update on lecturer side.

<u>X2 Admin Views (separate):</u>

1. *Programme Coordinators View:*
   a. View all pending claims with necessary information on each claim including the documentation to view / download.
   b. Verify and reject buttons.
2. *Academic Managers View:*
   a. View all pending claims with necessary information on each claim including the documentation to view / download.
   b. Approve and reject buttons.

<u>Unit Testing and Error Handling:</u>

Minimum of 5-unit tests to test the main functionality of the application. Error handling should be implemented throughout the whole application with meaningful error messages.

<u>Version Control:</u>

Regularly commit and push to GitHub. Minimum 10 times, preferably more, with descriptive commit messages.

<u>Documentation:</u>

The read me file needs to be updated with changes made from part 1 to part 2 based on lecturer feedback.

<u>Submission:</u>

On ARC: ONLY your GitHub link.

Inside the GitHub link:

- **Source code.**
- **Read me file with documentation that has been updated.**
- **YouTube video link.**
  o **Make sure your video is "unlisted", NOT private and NOT public.**