/50

1. Term Test 3 Submission (10 marks)

a. You are to create two (2) new PHP files named **`termtest3out.php`** and **`termtest3in.php`**

b. Be sure that your new **`termtest3*.php`** pages incorporates your **`header.php`** and **`footer.php`** files

c. Your **`header.php`** file should be changed to include a link in your navigation bar to the new **`termtest3*.php`** files (to make them part of your site).

NOTE: your pages should incorporate your CSS (this should be already done), and look like your existing pages.

d. Your `header.php` should be given appropriate variables that make the page's <title> element, the page's banner and HTML comments (including filename, date created, a brief file description) dynamic (similar to your lab assignments).

NOTE: the <title> should be "**YOUR_NAME's Test 3 – Club Applications**"

e. Include in the **`termtest3*.php`** files PHP comments (i.e. **`/* C-style comments*/`**) that include your name, the date and the course code.

Your pages should include description paragraphs, identical to the ones on the example pages provided (see links in the requirements)

f. Ensure sure that your **`footer.php`** file includes the clickable XHTML validation logo (to be used for marking).

g. Verify that your page validates to the XHTML 1.0 Strict standard, using the validator found at **https://validator.w3.org/** maximum 3 marks penalty PER PAGE.

h. For full marks you should publish your modified website out onto the course webserver, into your working directory (i.e. **`/var/www/html/inft1206/lastnamefirstinitial`**)

BE SURE TO PUBLISH ALL YOUR **`termtest3in.php`**, **`termtest3out.php`**, **`club_applications.sql`**, **`header.php`** and **`footer.php`** files

i. The **`club_applications.sql`** files created as part of the test should be published to your **sql** sub-directory

j. For backup purposes, also compress/zip your "**`lastnamefirstinitial`**" folder (your whole website with all of your test work included), attach the file into the "**Term Test 3 – Practical Portion**" drop box in DC Connect.

k. Place a clickable link in the comment section of DC Connect to your **`termtest3out.php`** file on **opentech**

NOTE: you may only submit once in DC Connect. Ensure there is a clickable link and attached *.zip file before submitting. Email or MS Teams Submissions will not be accepted.

2. Database Table `club_applications.sql` (15 marks)

a. Create a SQL script file named **`club_applications.sql`**
   - i. include SQL comments near the top of the file that have your name, the file name, the course code and the date the file was created.
   - ii. include a statement that removes the **`club_applications`** table from the database, but only if the **`club_applications`** table exists

b. Then include a statement that will `CREATE` a table named `club_applications`, the `club_applications` table should include:
   - i. a field named **club_code** that holds 4 characters (and ONLY 4 characters)
     - this field should be used as the table's unique identifier
   - ii. a field named **club_name** that can hold strings up to 60 characters (but only uses as many as it needs)
   - iii. a field named **member_count** that can hold integers
   - iv. a field named **contact_email** that can hold strings up to 255 characters (but only uses as many as it needs)
   - v. a field named **date_submitted** that holds a DATE

c. All privileges should be granted to the `faculty` role

d. All fields are mandatory

e. Creates two (2) records for the following:

Club Code	**Club Name**	**Initial Member Count**	**Contact Email**	**Date Submitted**
CODE	Code Club	14	Nadia.Goralski@durhamcollege.ca	2021-01-14
WEBD	Web Development Club	31	You DC Mail email	2023-01-09

f. Publish **`club_applications.sql`** onto the opentech server in your **`/var/www/html/webd2201/lastnamefirstinitial/sql`** folder

g. **`club_applications.sql`** should be ran against your `lastnamefirstinitial_db` on the opentech server through PuTTy

3. termtest3in.php page (20 marks)

a. Create a new **termtest3out.php** that looks identical (including banner and description paragraph) to the file found at:

   http://opentech.durhamcollege.org/inft1206/nadia/termtest3v1in.php

   NOTE: you are allowed to use the form and page content from the version provided on the opentech server.
   Right-mouse click- > View Page Source and copy/paste the description paragraph and the form.

b. The termtest3in.php should:
   - define constants for the following (to be used in the description paragraph, data validation and error messaging):
     i. **CLUB_CODE_LENGTH** with a value of 4
     ii. **MAX_CLUB_NAME_LENGTH** with a value of 60
     iii. **MIN_MEMBER_COUNT** with a value of 10
     iv. **MAXIMUM_EMAIL_LENGTH** with a value of 255

   NOTE: the last constant might already be declared in your functions if you completed Lab 10. If this is the case, you DO NOT need to re-declared it in your test file
   - have a self-referring and "sticky" form that has contains appropriately named (i.e. not $input1, $input2, $input3 etc.) text input boxes for: a club code, the club name, the member count, and an email address
   - have a submit button that contains an appropriate value
   - determine when the form is **POST**'ed to itself and retrieves the club code, the club name, the member count, and the email address from the **$_POST[]** array.
   - The inputted data should have leading and trailing white-spaced removed
   - Validates all fields (see below for validation rules)
   - INSERT a new club application record in the **club_applications** table if there were no errors (todays date can go into the date_submitted field
   - after a successful INSERT the user should be automatically redirected to the **termtest3v1out** page, flushing the buffer

c. Appropriate validation should occur for all fields:

- Use the pre-defined PHP `strlen()` function to determine the string length of the user inputted data.
- verify that a club code was entered (i.e. not an empty string), that the club code is exactly **`CLUB_CODE_LENGTH`** (i.e. 4) characters.
- verify that a name was entered (i.e. not an empty string), the name entered does not exceed **`MAX_CLUB_NAME_LENGTH`** (i.e. 60) characters long, creates a detailed message including the invalid data (where appropriate) otherwise.
- verify that an email address was entered (i.e. not an empty string), that the email address entered is no greater than **`MAXIMUM_EMAIL_LENGTH`** (i.e. 255) characters long, Creates detailed messages (including the invalid data where appropriate) otherwise.
- retrieve the member count from input, verify that there was something entered, that what was entered (i.e. not an empty string) was an integer (using the **`filter_var($member_count, FILTER_VALIDATE_INT))`** `and the` integer entered was at least **`MIN_MEMBER_COUNT`** (i.e. 10). Each input gets its own specific error
- When all validation has passed, verify that the club code entered is not already currently a record. Display an appropriate error message.
- Any error messages should be displayed above the form

NOTES:

Remember your page may visually differ in terms of colours, fonts, font sizes, etc. depending on your customized CSS. It is not necessary to make the pages identical, but your form should work similar.

4. termtest3out.php page (10 marks)

a. Create a new **termtest3out.php** that looks identical (including banner and description paragraph) to the file found at:

   http://opentech.durhamcollege.org/inft1206/nadia/termtest3v1out.php

b. The file should connect to your database using your **db_connect()** function from Lab 9

c. Query the **club_applications** table, using a SQL SELECT statement, to retrieve the **club_code, club_name, contact_email,** and **date_submitted** from all the records in the **club_applications** table in your database.

d. Sort by the club applications by date submitted ascending

   - For full marks you must specify the order in your SQL statement

e. In a for loop your page should create an XHTML table that displays all of the club application records in your database (as demonstrated above)

NOTES:

Use the **pg_connect()** function if you did not create a **db_connect()** function, there will be a one (1) mark penalty.

If your PostgreSQL database password on the opentech server is still your student number there will be a one (1) mark penalty.

NOTES:

Remember your page may visually differ in terms of colours, fonts, font sizes, etc. depending on your customized CSS. It is not necessary to make the pages identical, but your form should work similar.