CPSC 304 Project Cover Page

University of British Columbia, Vancouver

Department of Computer Science

Milestone: 03

Date: October 29, 2023

Group Number: 102

Name	Student Number	CS Alias (Userid)	Preferred Email Address
William Chow	93966943	x3c4k	williamchow604@gmail.com
Simrit Nijjar	66234287	s9m3h	simritnijjar@outlook.com
Davis Horton	17572686	i2o5k	hortondavis1@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Table of Contents

Project Summary

Disclaimer

Tasks

- Frontend
  - **Create an input form for adding data to the database.**
  - **Create an interface for viewing the database on the client side**
  - **Create an interface for filtering the database and viewing on the client side**
  - **Create an interface for updating an item in the database**
  - **Create an interface for deleting an item from the database**
- Backend
  - **Handle adding an item to a table in the database**
  - **Return the database on the client side**
  - **Return Filtered database on the client side**
  - **Update an item in the database and return to Client**
  - **Deleting an item from the database and return to Client**

Timeline

Project Summary

A database that holds media made from different studios and offered by streaming services. A user can subscribe to a streaming services and save media to history or to watch later. Users interact with the database through a web GUI.

Disclaimer

We are still in the early stages of learning SQL and PHP, so the task list detailed below is currently at a very conceptual level and may be subject to changes as we continue to deepen our understanding of these technologies.

Tasks

Frontend

Create an input form for adding data to the database.

- Allow selection of entity type when adding new entity to the database
- Create the relevant fields to be inserted into the Database
- Click a “Add” button to handle the request via the backend using Javascript and PHP
- Handle **Success**
  - Show a “Successfully Added” message on the front end
- Handle **Failure**
  - Show a “Failed to Add” message on the front end

Create an interface for viewing the database on the client side

- Allow selection of relevant relation / table to be viewed by the user
- Click a “View” button to handle the request via the backend using Javascript and PHP
- Handle **Success**
  - Show the table using HTML table to the user
- Handle **Failure**
  - Show a “Failed to get [selected table]” message to the user

Create an interface for filtering the database and viewing on the client side

- Continue from the implementation of the above feature to view the database.
- Along with selecting the desired table, create a form with the relevant fields to filter the data based on the given parameter

- Create a button “View” to handle the request via the backend using Javascript and PHP
- Handle **Success**
  - Show the table using an HTML table to the user
- Handle **Failure**
  - Show a “Failed to get [Selected Table]” message to the user

Create an interface for updating an item in the database

- Create an option to select a row from a currently displayed table
- Create a button for “Update” to open a new form
  - Pre-fill the form with the current values of the the selected row
  - Allow the user to update the relevant fields in the row
- Create a button for “Submit” to handle the request via the backend using Javascript and PHP
- Handle **Success**
  - Show a “Success” message to the user and update the view of the table
- Handle **Failure**
  - Show a “Failed to Update” Message to the user

Create an interface for deleting an item from the database

- Create an option to select a row from a currently displayed table
- Create a button for “Delete” to handle the request via the backend using Javascript and PHP
- Handle **Success**
  - Show a “Successfully Deleted” message to the user and update the view of the table
- Handle **Failure**
  - Show a “Failed to Delete” Message to the user

Backend

Handle adding an item to a table in the database

- Validate Input Parameters with Javascript
- Connect to Oracle Server through PHP Script
- Construct SQL `INSERT` Query within the script
- Execute the SQL `INSERT` Query

Return the database on the client side

- Connect to the Oracle Server through PHP Script
- Construct SQL `SELECT * FROM [selected table]` Query within the script
- Execute the SQL SELECT Query and return the results as an HTML Table using `Echo` commands

Return Filtered database on the client side

- Continue from the implementation of the above feature to view the database.
- Validate the input data of the form using Javascript
- Continue constructing the `SELECT` query to include a `WHERE` clause for filtering the dataset.
- Execute the SQL `SELECT * FROM [selected table] WHERE [constructed argument]` and return the result as an HTML Table using `Echo` commands.

Update an item in the database and return to Client

- Validate the input fields using javascript
- Establish a connection with the Oracle server within the script
- Construct an `UPDATE [selected table] SET [column attribute] = :[updated field] WHERE [primary key] = :[column identifier]` query
- Execute the SQL `UPDATE` query.
- Update the values of the table using another `SELECT` query as stated above

- Return the result as a table using `Echo` commands

Deleting an item from the database and return to Client

- Establish a connection with the Oracle server within the script
- Construct an `DELETE FROM [selected table] WHERE [primary key] = :[column identifier]` query
- Execute the SQL `DELETE` query.
- Update the values of the table using another `SELECT` query as stated above
- Return the result as a table using `Echo` commands

Timeline

Week	Date Start	Date End	Task(s)	Work Assigned To
Week 01	Nov 29	Nov 4	**Initialization**: Setup Oracle Database and SQL Script	Davis, Simrit, Davis
			Initialization: Create HTML/PHP Environment	Simrit, William, Davis
Week 02	Nov 5	Nov 11	**Frontend:** Create an input form for adding data to the database.	William
			Backend: Handle adding an item to a table in the database	Davis
			Frontend: Create an interface for viewing the database on the client side	Simrit, William

			Backend: Return the database on the client side	Simrit, Davis
Week 03	Nov 12	Nov 18	**Frontend**: Create an interface for filtering the database and viewing on the client side	Simrit
			Backend: Return Filtered database on the client side	William
			Frontend: Create an interface for updating an item in the database	Davis, William
			Backend: Update an item in the database and return to Client	Davis, Simrit
Week 04	Nov 19	Nov 25	**Frontend**: Create an interface for deleting an item from the database	Simrit, William
			Backend: Deleting an item from the database and return to Client	Davis, William
Week 05	Nov 26	Nov 1	**Finalizing**: Clean up and optimize code in preparation for Milestone 05	Simrit, William, Davis
Week 06	Dec 1	Dec 4	**Prepare for Project Demo**: Prepare commands to re-create and	Simrit, William, Davis

			repopulate tables for the demo.	