|  |
| --- |
| Muhammad Ashkar Yousuf Milton |
| WEB-601 |
| Milestone 2 |

|  |
| --- |
| Documentation and Report  10-24-2019 |

Table of Contents

[1. Introduction 2](#_Toc22878134)

[2. Database with MySQL 2](#_Toc22878135)

[a. Users: 2](#_Toc22878136)

[b. Graphics: 2](#_Toc22878137)

[c. SupportTeam 3](#_Toc22878138)

[3. Building the Website 3](#_Toc22878139)

[4. Server 3](#_Toc22878140)

[5. Client 5](#_Toc22878141)

[6. Information in the Development of the website 7](#_Toc22878142)

[7. References 8](#_Toc22878143)

# Introduction

For this assignment we have been focusing on connecting SQL and the website server. The main purpose is to create database entry and then connect our website to retrieve, to delete, to insert, or to add from our database with the help of website we are going to build. In order to that I have used VS Code to create the website and I used MySQL to create the database structure.

The logical design of the database structured used in the Website Development, includes:

Link of Website from GitHub is:

<https://github.com/loganwolf619/WEB601_Assignment_2>

# Database with MySQL

I used MySQL to create the database. The tables that are being used in the database includes:

* Users
* Graphics
* SupportTeam

## Users:

The Users table is where the information of all the users, who are Registering are being saved. The attributes of the Users table include:

* UsersID
* UsersEmail
* UsersFName
* UsersLName
* UsersPhone

Each User is recognized by the Email address they are going to register. So, when a user is going to register, their information will be saved based on the email he/she used to register.

## Graphics:

The Graphics table is where all types of Graphics is being added, updated, or deleted. The Graphics contains the information of each Graphics which includes the Artist, the type, and the Quality of the graphics. The attributes that are being used includes:

* GraphicsID
* GraphicsArtist
* GraphicsType
* GraphicsQuality

Each Graphics is recognized by the GraphicsID, the Graphics to be added.

## SupportTeam

The SupportTeam table is the table where a User can send message to the owner of the website. It is more or less is providing an opportunity for the users to connect with the owner of the Website and sort out any issues or questions accordingly.

Every time a Message is being sent; a unique number of the message will be generated. In order to send a message, the user must use the First Name, Last Name, Email address, and the Message the user would like to commit.

The attributes used for Support Team includes:

* SupportID
* FirstName
* LastName
* Email
* Message

# Building the Website

To build the website, I have to make sure I have created and put the files in the right folder. In order to do that, I have created folder for client and server. The client would be the front end of the website development where customers are going to see the development of the website. In the client folder I will be creating the files which I will be using for the client.

And the server would be the back end of the development where the developers would be running the system. In the server folder I created files where there will be files that I need to work for the server.

# Server

I created the server folder and in the server folder the files that I have included are config.js file, middleware.js, package.json, restful.js, routes.js, dbUsers.js, dbSupportTeam.js, and dbGraphics.js. The config.js file is used to connect the database (MySQL) where I created tables, used for collecting information from the website. In order to connect with the database, in the config.js file I used host, port, user, password, name of the database, and insecureAuth to have a successful connection.

// Server will be connecting with the host

const APIServerPort = 4200

// This is used to connect with my database

const database = {

    host: 'localhost',

    port: 3306,

    user: 'milton',

    password: 'Shayek619',

    database: 'apex\_shutterbug',

    insecureAuth: true

}

// This is used to export my database connection and the port that is being used

module.exports = {

    database,

    APIServerPort

}

Because of that, any updates, insert, or delete can be performed from the updates with the connection of the database. Then we are going to use middleware.js to create a variable by taking a GraphicsID.

// This midleware is used to connect the backend and the frontend together.

function checkID(req, res, next) {

    const GraphicsID = +req.params.GraphicsID

//  We are going to create a variable to save the ID. Then we are going to check if the ID is an integer

    if (Number.isInteger(GraphicsID)) {

        next()

    } else {

        return res.status(400).json('The ID needs to be an integer')

    }

}

// We are using this to export module s

module.exports = {

    checkID

}

This will help us to check if we are using correct integer to connect with the database. Then we are using restful.js so that we can use connect our database with the help of KNEX. In the restful.js file, we are going to get, post, patch, or delete any changes that is being made in the website, and this will get updated and displayed on the database, that is related to KNEX. Then we are using routes.js to connect to the request and the KNEX that has been used for the development of the website.

//This is connect the Request and the KNEX query that I created for the database, which

// will be used in the development of this Website

const graphicsList = require('./dbGraphics.js')

const supportTeam = require('./dbSupportTeam.js')

const users = require('./dbUsers.js')

module.exports = {

    graphicsList,

    supportTeam,

    users

}

Then, the other files: dbGraphics.js, dbSupportTeam.js, and dbUsers.js are being used to have connection with the database so when there is any changes made in the website, it will make changes in the database.

# Client

In the client folder, we are going to create few folders which include src and public. In the public folder we are going to create folders which include icons and images, and the files which includes index.css and index.html. These folders and files are used by the website and each website pages have connection with them. If we are going to use the images or icons for the development of the website, we are going to retrieve the icons or images to be displayed on the website, from these folders. Then to style or change structure of the website we are going to create universal css file, which is index.css. In the index.css we are going to apply a universal design of the website which includes the design of the website for each page, the font colour and size, and others.

Next, the src folder is where we are going to create a folder which include components. We are also going to make few files which include: App.css, App.js, index.css, and index.js.

The index.css is the file where I will set the structure, design, background, font-size, font-colour, the structure of navbar, and other changes needed to make for the development of the website. Next, we are using app.js to provide links to the page on the navbar that we are going to use for the development of our website.

import React from 'react';

import {Switch,Route} from 'react-router-dom';

import './App.css';

import 'bootstrap/dist/css/bootstrap.min.css';

import Navbar from'./components/Navbar';

import Home from'./components/pages/Home/Home';

import ContactUs from'./components/pages/ContactUs/ContactUs';

import MyUsers from'./components/pages/MyUsers/MyUsers';

import Default from './components/pages/Default'

import BrowseGraphics from './components/pages/BrowseGraphics/Browse'

import Footer from './components/Footer'

import Login from './components/pages/Login/Login'

import Register from'./components/pages/Register/Register';

import MyGraphics from './components/pages/MyGraphics/MyGraphics';

import UploadGraphics from'./components/pages/UploadGraphics/UploadGraphics';

import EditUsers from'./components/pages/EditUsers/EditUsers';

import EditGraphics from './components/pages/EditGraphics/EditGraphics';

import AboutUs from'./components/pages/AboutUs/AboutUs';

const App = () => (

  <div className="container">

        <div className="header">

          <Navbar />

        </div>

        <div className="body">

        <Switch>

              <Route exact path="/" component={Home} />

              <Route path="/browsegraphics" component={BrowseGraphics} />

              <Route path="/contactus" component={ContactUs} />

              <Route path="/login" component={Login} />

              <Route path="/register" component={Register} />

              <Route path="/myusers" component={MyUsers} />

              <Route path="/mygraphics" component={MyGraphics} />

              <Route path="/uploadgraphics" component={UploadGraphics} />

              <Route path="/editusers" component={EditUsers} />

              <Route path="/editgraphics" component={EditGraphics} />

              <Route path="/aboutus" component={AboutUs} />

              <Route component={Default} />

            </Switch>

        </div>

        <div className="footer">

          <Footer />

        </div>

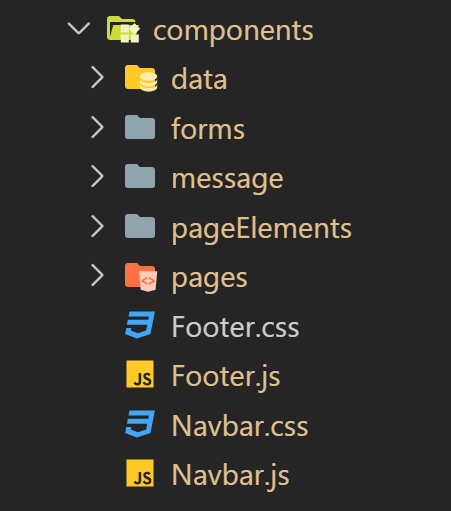
      </div>

)

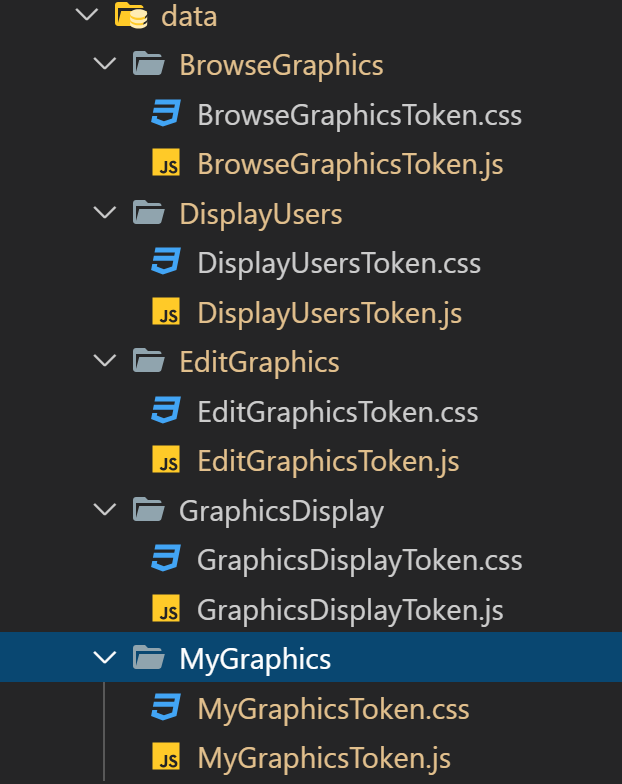
// We will be exporting the variable

export default App;

Then we have created components folder and inside the components folder we have data folder, forms folder, message folder, and the pageElements folder.



In the data folder we are going to create different subfolders which include:



These files would be used to Browse Graphics in the Browse Graphics tabs, that has been added in the website. Next, display users is used to display the users that are connected to the database. We are going to use Edit Graphics to show what we have edited.

We have forms folder so that when users can use the form to register their credentials. Here, the user is going to register their name by typing the user’s email and the password. To make sure the user entered the correct format, I have use Validator. It will check the validity of the information the user has entered. If a user has entered wrong entry then an error message will be sent, to let the user know about the error. I got this idea from Daniel, who showed and helped me to work on this.

Next, I got the message folder that contains the message.js and message.css file. This file is the message that is going to be displayed when the user enters the wrong information. I also have pageElements folder, which include Button.js, Title.js, DropDown.js, and the StyledNavBar.js

I also have included pages folder and each pages folder shows all the page elements that is going to be displayed on the navbar.

# Information in the Development of the website

When I was completing this assignment, it took me a while to do a lot. So, much information and so much things to know at once, its hard to carry on with the assignment. Most of the things I don’t understand, and it took me time to figure out those questions. It took me a lot of time to connect my database with the website. But, anyway I figured it out somewhat, still needed some work to make it run alright.

# References

1. <https://github.com/DanielM-IT/WEB601_Project>
2. <https://www.youtube.com/watch?v=G4e15f7qfVs&t=1877s>
3. <https://www.youtube.com/watch?v=IPQoMiu6zTI&t=3s>
4. <https://www.youtube.com/watch?v=LVfH5FDOa3o>
5. <https://www.youtube.com/watch?v=hE5zeEiVqpw>