|  |  |
| --- | --- |
|  | **CIS-2288 Internet Programming II** |

Assignment 2

|  |  |
| --- | --- |
| To: | Don Bowers |
| From: | Christopher Sigouin |
| Date: | October 14, 2015 |
| Subject: | Assignment 2 – Guest Book Web Application |
|  |  |

**QUICK START GUIDE**

1. The assignment zip file can be downloaded from SAM or cloned via BitBucket.

REPOSITORY: cis2288\_20152016\_assignments\_sigouin\_christopher.git

1. Extract the application zip file into your web server folder.
2. This application has 4 packaged dependencies. Two of them need to be downloaded and installed: Securimage and Ckeditor

Download and install them from their designated homepages. They should be extracted directly into the root of the application directory structure.

<https://www.phpcaptcha.org/download/>

<http://ckeditor.com/download> ( Basic Package )

**If you cannot see the captcha image in the register modal or ckeditor in the add or edit modals, it is because you do not have permissions to them. In order to get permissions, you will need to provide " Read, Write, Execute " permissions. If your OS is Linux based, you can use these commands from within the root directory:**

**chmod 707 -R securimage**

**chmod 707 -R ckeditor**

**For Windows, just make sure the root directory of ‘ckeditor’ and ‘securimage’**

**is located inside the root directory of the application. Windows built-in zip application**

**will add another folder/container and store the contents within it when you extract.**

1. Open ‘phpMyAdmin’, choose ‘Import’ and click ‘Choose File’. Search for the database.sql file located within the directory ‘db’ in the root of the application.

(This will create 3 users and some comment data that is assigned to each user. This is only for testing purposes and can be deleted if you wish. The users are listed as follows:

Users: admin / password

John / doe

Sally / abc123

1. For access to the database through phpMyAdmin, the username and password would be the following: guestBookUser / guestBookPassword
2. Locate the ‘index.php’ file within the root of the application file structure and execute it from the web server folder. This will start the application.

**NOTE:**

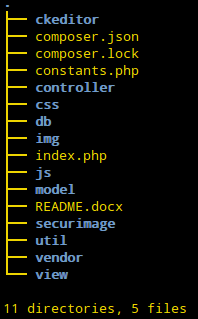
* + Admin credentials are worked into the application, so as admin you are able to add, edit ( any users comment ) and delete ( any users comment ).

**DOCUMENTATION AND DIRECTORY STRUCTURE**

The application for this exercise was to design a ‘Guest Book’ that would allow users to do the following:

* Register a new account
* Login / Logout of user accounts
* View all comments made by all users
* Add a new comment
* Edit comments ( users can only edit their own comments )
* Delete comments ( users can only delete their own comments )

A directory structure for the application is shown below. Included are some brief points describing their purpose within the application. When building this application, I attempted to design the structure with a ‘Model-View-Controller’ outlook, so the corresponding directories will match types of source files.



**FILES**

COMPOSER.JSON

COMPOSER.LOCK

Files that belong with the configuration for the application. These are produced by the dependency manager ‘Composer’.

CONSTANTS.PHP

Holds constants that are used to define source files which are utilized throughout the application

INDEX.PHP

Starting page of the ‘Guest Book’ application.

README.DOCX

This file.

**DIRECTORIES**

CKEDITOR (DIR)

Holds the WYSIWYG editor that attaches itself to text area elements for the add and edit comment sections

CONTROLLER

Source files that manage data between the views and models of the application. Only one source file is contained within this directory.

CSS

Used for custom CSS style sheets.

DB

Holds files pertaining to the database

IMG

Images associated with the application

JS

Used for custom JavaScript source files

MODEL

Source files that represent the ‘models’ of the application. These are the chosen objects such as User, Guestbook, and Comment. They would interact directly with the database and pass off data to the application controller.

SECURIMAGE

Free PHP based CAPTCHA application that was integrated with the registration process

UTIL

Holds classes that are used with the database and any helper classes and files.

VENDOR

Any dependencies that are required through Composer are placed into this directory. Two additional technologies were implemented into this project. Bootstrap and JQuery.

VIEW

Holds all source files that represent the user’s view of the application. These include index.php as well as modals from Bootstrap.