**Database Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *Kody’s Blog website* | |
| **Date:** | *1/23/2021* | |
| **Revision:** | *1.8* | |
| **Team:** | 1. *Kody O’Neill* | |
|  | |
|  | |
|  | |
| **Team Status:** | |  |  |  |  | | --- | --- | --- | --- | | **Task** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *Update comments in each script* | *Kody O’Neill* | *1* | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | |  | *Kody O’Neill* |  | *0* | | |
| **GIT URL:** | *N/A* | |
| **Hosting URL:** | *https://kodysblog.azurewebsites.net/myapp/blog/index.php* | |
| **Peer Review:** | *Y/N* | We acknowledge that our team has reviewed this Report and we agree to the approach we are all taking. |

**Supporting Design Documentation**

**Install Instructions:**

*Setup:*

*Database set up:*

*1)go to https://portal.azure.com/#home*

*2)go to “insert web app name here”*

*3)click on “MYSQL in App” on the left hand side*

*4)Click the “Manage” at the top to open the web app to manage your SQL server*

*5)click on the databases tab*

*6)put in a name in the "Database name" and click create*

*7)click on the new database to the left*

*8)under structure create new table with 4 columns*

*9)set column values for ID, Email, Username, Password, and Login Attempts*

*10)ID is Type INT with AI checked, Login Attempts is INT with 0 as DEFAULT value, the rest are Type VARCHAR with 100 char limit*

*11)Create new table with BLOG\_ID as INT with AI, BLOG\_TITLE is VARCHAR at 100, BLOG\_BODY is VARCHAR at 5000, USER\_ID is brought over from previous table, DATE is set with current timestamp as default.*

*12)Database setup is complete. Next upload your web app files to azure through a zip file or through a repository*

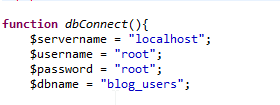
*13)Navigate to:*

*The website URL that you have set up in Azure*

*Configuration:*

*You will need to update the PHP files with the names for your database, username, password and table names. You can get these from azures SQL configuration file. Please see red arrows for configuration change location.*

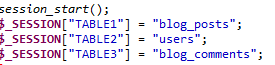
*myfuncs PHP Script*

**

*Index.php*

**

*loginHandler.php (did this incase user went straight to login and skipped index.php)*

**

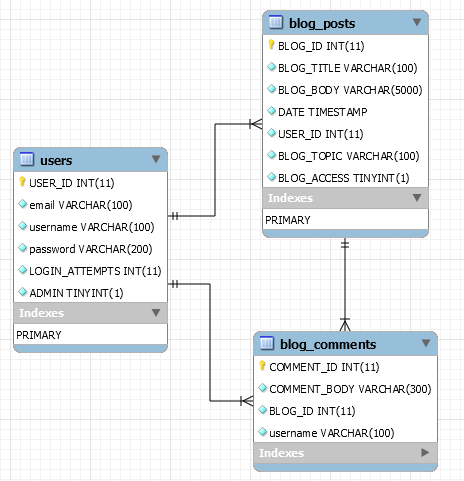
**General Technical Approach:**

*I broke down the problem into bit sized pieces and tackled each one individually. First I started with the HTML pages and made sure I had all the necessary fields and pages created. Next, I updated the database tables to accept these new values. Next, I tackled the values being posted and stored in variables in the PHP scripts. Next, I built a special homepage for admins that had specifics links not normally available. Lastly, I tackled updating and deleting currently stored values for users.*

**Key Technical Design Decisions:**

*I created a simple in program filter to find and replace keywords or phrases that are forbidden. Also used a find and replace to allow apostrophe’s in the blog post since that is the proper spelling of words. Included a log out function to clear the session user id variable so that you cannot go directly to posting a blog without signing in first. Created a simple search that brings up all blogs by their blog ID and title based on whoever is logged in. Created more links to go to previous pages or specific pages. I made an emphasis to make sure that a user was logged in by storing their user ID in a session variable. Then when they logged out or went back to the main menu or index page it unset these variables. This way users cannot access unauthorized files and blog posts. I have now moved the table names to session variables to cut down on the areas you would have to update for configuration. Built out new links in the home page for the admins. Allowed the admin to view, edit, and delete all blogs. When admin deletes user it also deletes all blog posts made by the user. Set up new search functions that are separate based on if Admin is logged in or standard user. Put in login check before allowing comments to be posted, updated, or deleted.*

**ER Diagram:**

**

**DDL Scripts:**

*None created at this time*

**Sitemap Diagram:**

*None created at this time. Very easy to navigate*

**User Interface Diagrams:**

*The flow from one page to another is via links and buttons. Very self-explanatory.*

**Other Documentation:**

*No additional information was produced.*