INFO2180 - Final Project (40 marks)

Due Date: **Friday, November 29, 2019**

BugMe Issue Tracker Overview

BugMe Issue Tracker is a simple issue tracking app that allows software development team members to log new feature proposals, tasks or bugs for a web app that they are currently working on.

This project is a group project and will require you to be in groups of maximum 4 persons. You must declare your groups by the latest November 8, 2019.

Feel free to use AWS Cloud9 to work on this project so as to use its collaboration features with your group members and you must create and commit your code to a Github repository.

Database

You should create the following database tables with the following fields:

Users
id
firstname
lastname
password
email
date_joined
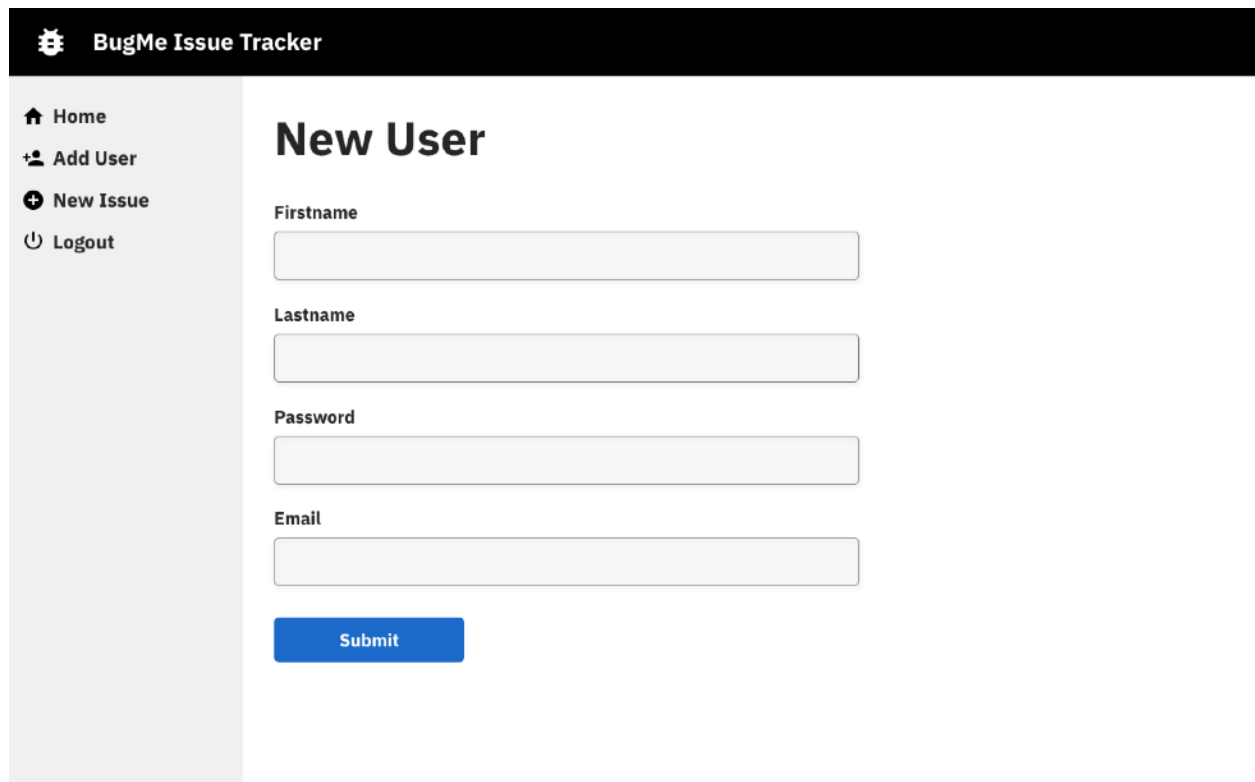
Issues
id
title
description
type
priority
status
assigned_to *(store the appropriate user id)*
created_by *(store the appropriate user id)*
created
updated

Note: You should create a file called **schema.sql** with the relevant **CREATE TABLE** statements for the tables above and include in your submitted code. Also ensure you have an **INSERT** statement that adds a user with the email address **'admin@bugme.com'** and the password **'password123'** (ensure that the password is appropriately hashed of course).

Features

Adding a user

To simplify things for this assignment, users can only be added by an administrator, there is no feature for new users to self-sign up. An administrator logs in and completes the new user form (See Figure 1). You should use regular expressions to ensure that passwords have at least one number and one letter, and one capital letter and are at least 8 characters long. The password **MUST** be hashed before being stored in the database. Also you should ensure the other fields are validated and that user inputs are escaped and sanitized before being stored in the database.

FIGURE 1: WIREFRAME OF NEW USER CREATION SCREEN

User login

A user goes to the login page and logs in with their *Email address* and *password*. The system keeps track of the user using PHP sessions. Once logged in they are presented with the Dashboard/Home Screen which shows a list of issues in the issue tracker.
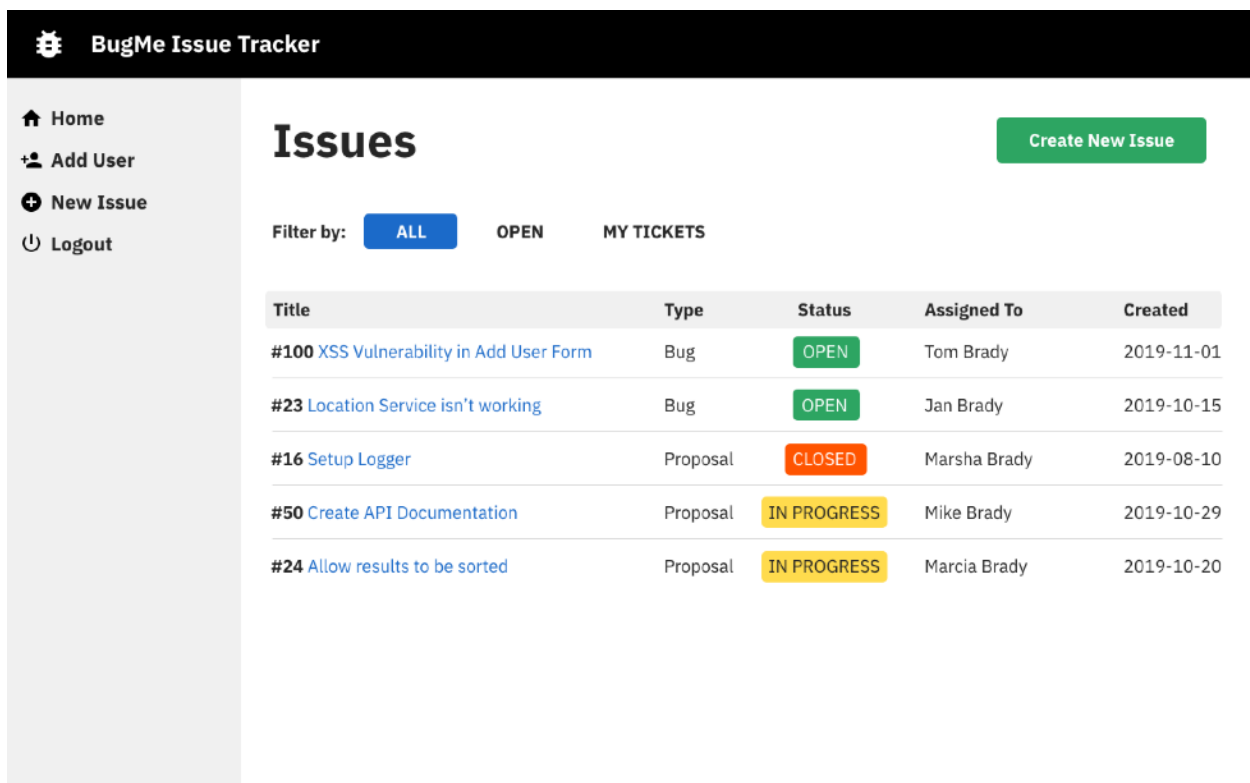
User logout

There will be a logout link/button which a user may click in order to logout of the system. When this is done, the PHP Session should be destroyed and the user redirected to the login screen.

Dashboard/Home Screen

The Dashboard/home screen allows a logged in user to see a list of all the issues. The list of issues should display the Title (with the Ticket ID Number), Type of Issue, Status, the full name of the user the ticket was assigned to and the date the ticket was created. There should also be a link that when clicked will allow the user to view the full Issue Description. (See Figure 2)

The page should also have a list of filters to display "*All Tickets*" no matter their status, "*Open Tickets*" only or "*My Tickets*" which should display only the tickets assigned to the currently logged in user.

FIGURE 2: WIREFRAME OF DASHBOARD/HOME SCREEN SHOWING ALL ISSUES AS WELL AS A FILTER TO DISPLAY ONLY OPEN ISSUES OR ISSUES BELONGING TO THE CURRENT LOGGED IN USER

Create a New Issue

The Create a New Issue screen includes a form with the following input fields: ***Title, Description, Type*** *(e.g. Bug, Proposal or Task),* ***Priority*** *(e.g. Minor, Major, Critical) and* ***Assigned To*** *(which should list the names of all the users)* input fields. (See Figure 3).

Whenever a new issue is created, it is automatically assigned a Status of "Open". You should also store the user id of the user who created the ticket and the **created** and **updated** date columns should be set to the current date and time. Also you are to ensure the input fields are validated and that user inputs are escaped and sanitized.

FIGURE 3: WIREFRAME OF CREATING A NEW ISSUE

Viewing Full Job Details

When an issue is clicked it will open and show the full details of the Issue (See Figure 4). These details should include the *Title, Description, Type, Priority, Status, Date Created and by whom, Date last updated, as well as who it is assigned to*. There should also be a button to Close the ticket that when clicked will update the ticket status to be "Closed" and another button that when clicked will mark the ticket as "In Progress". Whenever either of these buttons are clicked they should also update the "updated"

date column for the issue with the current date and time when the respective button is clicked.

FIGURE 4: WIREFRAME OF FULL DETAILS OF AN ISSUE

No Page Refreshes

All new pages should load without browser refresh, in other words you will need to implement an AJAX based approach to loading new content into the browser.

Submission

You will submit using the "Final Project Submission" link on OurVLE. Since it is a group project, only one group member is required to submit the relevant project links.

You are required to commit your code to Github and submit the link to your group project Github repository (e.g. https://github.com/yourusername/info2180-final-project).