

INFO2180 - Project 2 (40 marks)

DUE DATE

Friday, December 3, 2021

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 5 persons. You must declare your groups by the latest **Friday, November 12, 2021**.

Database

You should create a database called `bugme` which contains the following database tables with the following fields:

Users Table

Column Name	Data Type
id	INTEGER
firstname	VARCHAR
lastname	VARCHAR
password	VARCHAR
email	VARCHAR
date_joined	DATETIME

Issues Table

Column Name	Data Type
id	INTEGER
title	VARCHAR
description	TEXT
type	VARCHAR
priority	VARCHAR
status	VARCHAR
assigned_to <i>(store the appropriate user id)</i>	INTEGER
created_by <i>(store the appropriate user id)</i>	INTEGER
created	DATETIME
updated	DATETIME

NOTE

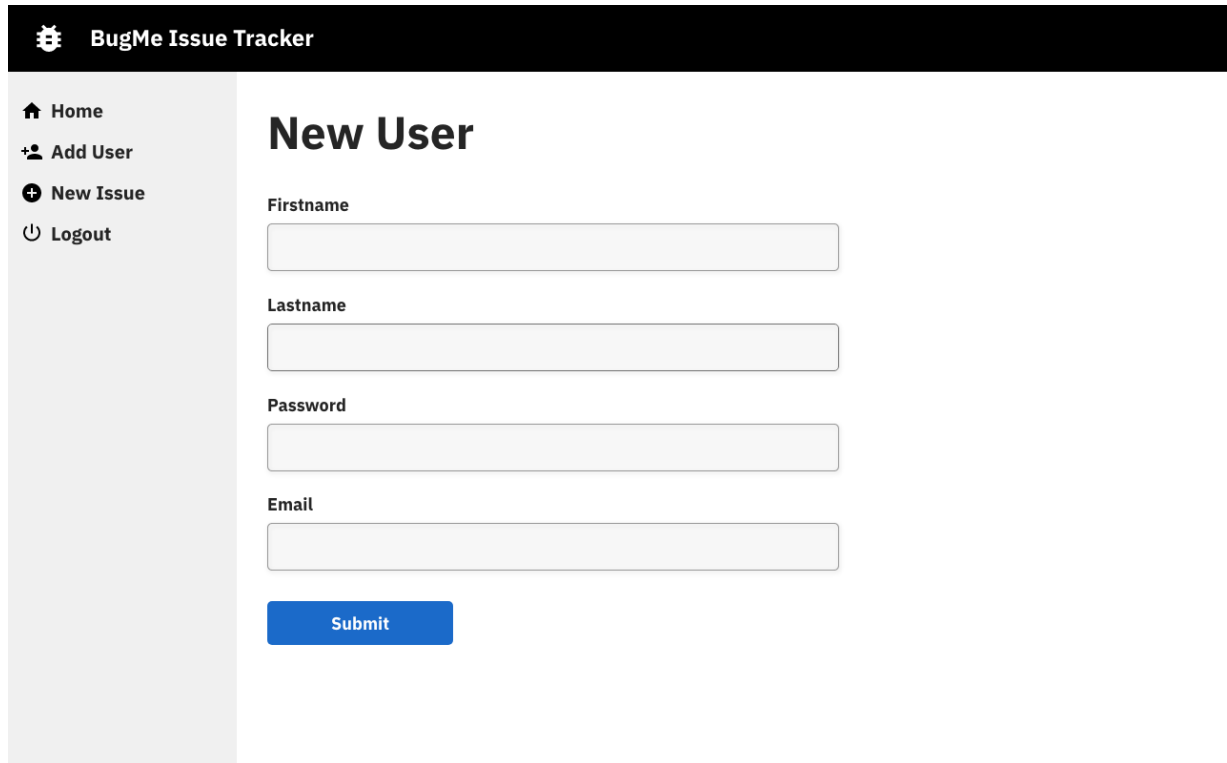
Note: You should create a file called `schema.sql` with the relevant `CREATE TABLE` statements for the tables above and include in your submitted code. Also ensure you have an `INSERT` statement that adds a user with the email address `admin@project2.com` and the password `password123` (Of course, you need to ensure that the password is appropriately hashed before using in your `INSERT` statement).

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. I suggest you use the PHP `password_hash()` [↗](#) function to hash your passwords. Also you should ensure the other fields are validated and that user inputs are escaped and sanitized before being stored in the database.



The image shows a wireframe for a 'New User' creation screen. At the top is a black header bar with a white bug icon and the text 'BugMe Issue Tracker'. Below the header is a light gray sidebar containing four menu items: 'Home' (with a house icon), 'Add User' (with a person icon), 'New Issue' (with a plus icon), and 'Logout' (with a power icon). The main content area has a white background and is titled 'New User' in a large, bold, black font. Below the title are four text input fields, each with a label above it: 'Firstname', 'Lastname', 'Password', and 'Email'. The 'Password' field is visually distinct from the others. At the bottom of the form is a blue rectangular button with the word 'Submit' in white text.

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.

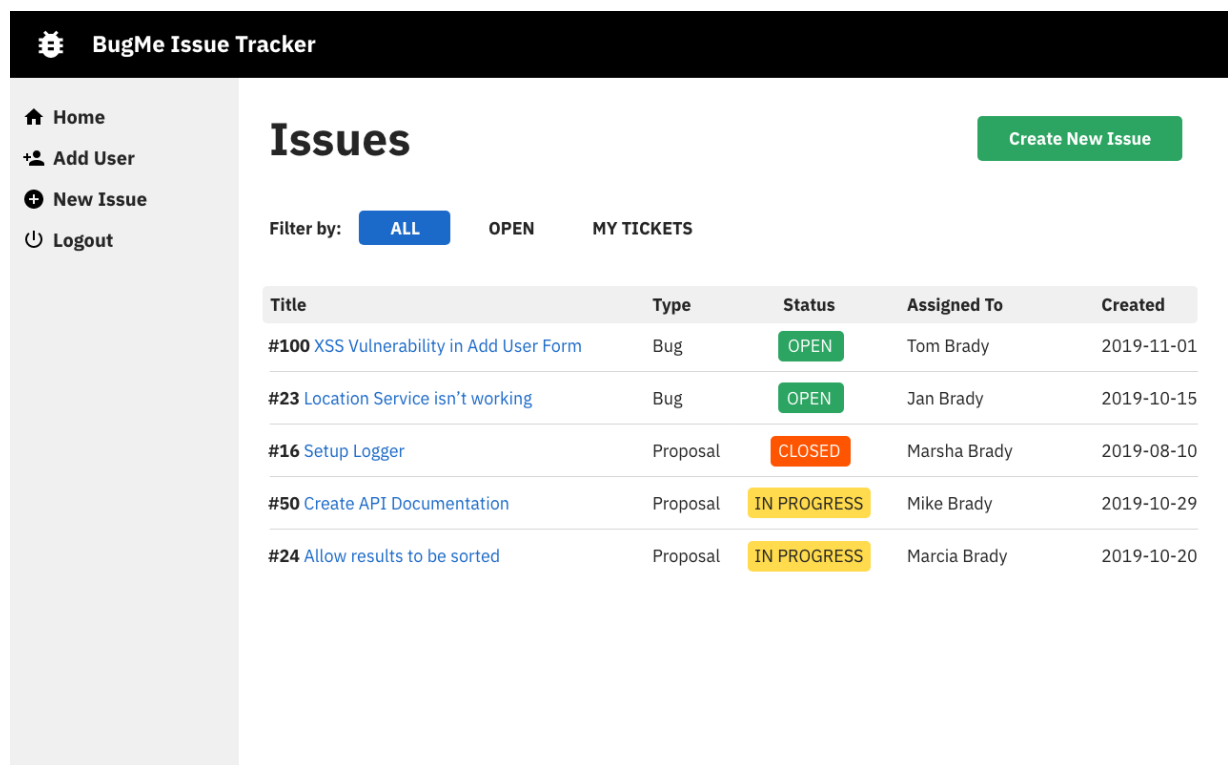
User Logout

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.



The wireframe shows a dashboard for 'BugMe Issue Tracker'. It features a sidebar with navigation links: Home, Add User, New Issue, and Logout. The main content area is titled 'Issues' and includes a 'Create New Issue' button. Below the title, there are filter buttons: 'ALL' (selected), 'OPEN', and 'MY TICKETS'. A table lists five issues with columns for Title, Type, Status, Assigned To, and Created. The issues are: #100 XSS Vulnerability in Add User Form (Bug, OPEN, Tom Brady, 2019-11-01), #23 Location Service isn't working (Bug, OPEN, Jan Brady, 2019-10-15), #16 Setup Logger (Proposal, CLOSED, Marsha Brady, 2019-08-10), #50 Create API Documentation (Proposal, IN PROGRESS, Mike Brady, 2019-10-29), and #24 Allow results to be sorted (Proposal, IN PROGRESS, Marcia Brady, 2019-10-20).


Title	Type	Status	Assigned To	Created
#100 XSS Vulnerability in Add User Form	Bug	OPEN	Tom Brady	2019-11-01
#23 Location Service isn't working	Bug	OPEN	Jan Brady	2019-10-15
#16 Setup Logger	Proposal	CLOSED	Marsha Brady	2019-08-10
#50 Create API Documentation	Proposal	IN PROGRESS	Mike Brady	2019-10-29
#24 Allow results to be sorted	Proposal	IN PROGRESS	Marcia Brady	2019-10-20

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.

 BugMe Issue Tracker

Home

Add User

New Issue

Logout

Create Issue

Title

Description

Assigned To

Marcia Brady

Type

Bug

Priority

Major

Submit

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 by using AJAX

All new pages should load the appropriate page information/data without the browser refreshing. In other words you will need to implement an AJAX based approach to loading new content into the browser and when doing form submissions.

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-finalproject>).