Group Members and ID Numbers:

* Ashley Perkins - 620129607
* Jaun-Luc Brown - 620130186
* Matthew Palmer - 620131688

Project Specification:

Database Section:

* A file named schema.sql should be able to recreate the database
* Password hashing should be implemented
* A user with the email: [admin@project2.com](mailto:admin@project2.com) and password: password123 should be inserted after creation, along with password hashing of course.

System Functions:

* User Addition
  + This is an administrator functionality so regular users should not have access to this.
  + This “page” contains a form with the fields: First Name, Last Name, Password and Email.
  + Regex will be used in validating user input. [Client Side]
  + Passwords must contain at least 1 number, 1 letter and 1 Capital letter and must be at least 8 characters long.
  + Passwords must be hashed before being stored in the database.
  + Data must be revalidated, escaped and sanitized before being stored in the database.
* User Login
  + Login Page will contain a form with two fields: Email Address and Password .
  + So Login Page links to the Dashboard/Home Screen if successful
  + User Activity will be kept track of using PHP Sessions.
* User Logout
  + Once the Logout button is pressed, PHP Session will be destroyed.
  + It links back to the Login Screen
* Dashboard/Home Screen
  + Once Logged In, A list of the issues will be presented. [Most Likely will be a table]
  + The table will show Title, Type of Issue , Status , Full Name of User it was Assigned to , And the Creation Date.
  + The Title in each record composes of The Ticket Number plus a link to A View Issue Page specific to the issue selected.
  + Above the table is a Filter Selection with Tabs : “All”, “Open”, “My Tickets”. [I assume that the All Tab is selected on Default]. My Tickets selects all the tickets assigned to the currently logged in user. This might be useful <https://www.w3schools.com/howto/howto_js_tabs.asp>
* Create An Issue:
  + Form with fields: Title, Description, Type, Priority and Assigned To:
  + Type, Priority and Assigned To are Drop Down Fields. Type is either Bug, Proposal or Task, Priority is either Minor Major Critical and Assigned To is a list of all Full Names within the system.
  + When an issue is created, automatic status of Open.
  + The user id of the person who created the issue should be stored. The created and updated fields should be set to the current date and time.
  + Validation, Escaping and Sanitization should be conducted on User Input.
* View Issue:
  + So this page should display full details of an issue once clicked
  + It will display the Title, Description , Type , Priority , Status, Date Created and by whom, Date Last Updated and Who it was assigned to.
  + There should be two buttons : Mark as Closed and Mark as In Progress which will update the status of the issue with the respective status as well as update the updated column with the current date and time.

Other Functionality:

* The design of this web app should take an ajax approach so new content should be loaded in without refreshing the browser.

Other Notes:

Design of the Interface:

The wireframes in the project seem relatively similar in terms of the layout. They each contain the header which contains an image and the name of the application. Their main content also each have an aside section for like action selection. The difference is in the content beside the aside section. To note, the login “screen” will not have an aside section, just its content centered.



The main could either have a grid layout with two columns , one for the aside section and one for the screen contents or just have a single column / flex container for its screen contents.

Interface Operation:

Regarding how the page will load in new content:

* We could take the approach of having new content just be loaded into main using js and php.