Designing and Developing Software Applications Task two

Background - iKanban

The application built is a “Trello” style online Kanban board in which users are able to create and collaborate on boards, dragging tasks between different, user-defined states. The application has been built on PHP using a MySQL database.

Setup

To setup the application copy the Assignment2 folder to a location you wish to host the site. Ensure you have MySQL installed and with a root user run the SQL file /assignment2/app/config/sql/createdb.sql. This will have created an Admin and a Standard user, a standard and admin role and given the admin/standard users the relevant roles.

To start the application navigate to the assignment2 folder and run using php –S localhost:8081.

Functionality

User registration and authentication

Users are able to register and login using the login/register tab on the navbar. Already created users can be found in the table below. When logged in as an admin user, users can access the admin tab on the Navbar which enables users to undelete boards (normal users can delete their own boards) and mark users as admin and as deleted.

Users must have a valid email address (valid format) and unique username however no restrictions are made on the size of the password or username. This would have however been desirable.

|  |  |  |
| --- | --- | --- |
| Username | Password (case sensitive) | Roles |
| Admin | Admin | Admin/Standard |
| Standard | Standard | Standard |
| Standard2 | Standard2 | Standard |

Creation of projects

All users are able to create boards and can choose whether these boards are public or private to them. To create a board, use the new board field on the home page or use the Create Board tab on the navbar. You can also select the colour of the board on the creation screen.

Searching of stored projects using a variety of criteria

Boards can be searched using the search bar on the home screen. Users can decide whether to search through public or private boards using the button available on the dropdown next to the search button.

Sharing of projects amongst users

Boards can be shared between users by using the invite users tab on the individual boards page. When typing in the Invite a User Box users will autocomplete in a dropdown box below and can be selected. Users can also be removed here.

Some aspect of the management of the project lifecycle

When created each board comes with 3 built in states (columns) which can be removed if required. New states can be added and the order altered if required using the arrows provided. Tasks can be added to each state and as each task moves through a project lifecycle, users can drag and drop the task to new states.

An appropriate user-interface and interaction style

The user interface is mobile friendly, easy to operate and efforts have been made to ensure the application works as well on mobile as it does on desktop. An additional Jquery library, Jquery UI touchpunch, has been used to assist with draggablity on mobile as well as desktop. HTML 5 Shim has been used to help on older browsers though this hasn’t been fully tested

Validation of newly uploaded data where appropriate

Usernames and emails have been validated on input, this is validated on both client and server side. States within boards can also not be moved beyond their max left or right. This is validated on client and server side.

The storage of all data in a database

All data is stored in a database.

REST API Usage

The API has been built in the /app/api folder. The API can be used in two ways. A JSON file can be submitted which will post the data to the database. Using Postman navigate to <http://localhost:8081/api/board> and using a post request submit the JSON data found in the root director called data.json. This will create two boards for user 1 named Tester1 and Tester2 but this can be changed in the JSON file.

The second way the API can be used is by submitting with parameters. Each type of http request (GET, PUT, POST, DELETE) can be submitted for Boards, Cards and States.

GET example request

http://localhost:8081/api/boards/1 will retrieve all details of board 1. A request of

It is possible to combine two entities in one request on the GET function. For example a request of <http://localhost:8081/api/boards/1/states> or <http://localhost:8081/api/boards/1/cards> will pull back the board information and the state or card.

DELETE example request

<http://localhost:8081/api/boards/2> Will delete board 2. Can also be performed on States and Cards. Will only mark as deleted in the database not actually delete. Can only delete so will fail if board has already been deleted.

POST example request

<http://localhost:8081/api/boards> this will insert a new board provided a board\_name parameter is provided in the post request. “Hex\_color” and “public” parameters can be provided also but will default if not. States can be inserted, “board\_name” and “state\_name” must be provided but position is optional. Cards can be inserted “state\_id” and “card\_name” must be provided.

PUT example

<http://localhost:8081/api/boards/1?hex_color=blue&deleted=0&board_name=Board1&public=1>

This will update board 1’s hex\_color, deleted, boardname and public attributes. All are optional but atleast one is required. Similar things can be done with state with attributes, state\_name and deleted and cards, with card\_name and deleted.