Total Life Manager

Logan Grisham, Sebastian Hardesty, Stephen Johnson

Dept. Of Computer Science and Information Technology

Abstract

Total Life Manager is a streamlined web server application that allows users to track nutritional and health goals. The website provides tools to a user to identify habits and plan for future health milestones. The website will track all the user's entered information and build charts and graphs to help the user digest the information instead of getting confused and discouraged. The user can benefit from the site as a whole-life tool because important information to daily life to help keep track of important events or goals will be available on the dashboard through the weather and a calendar.
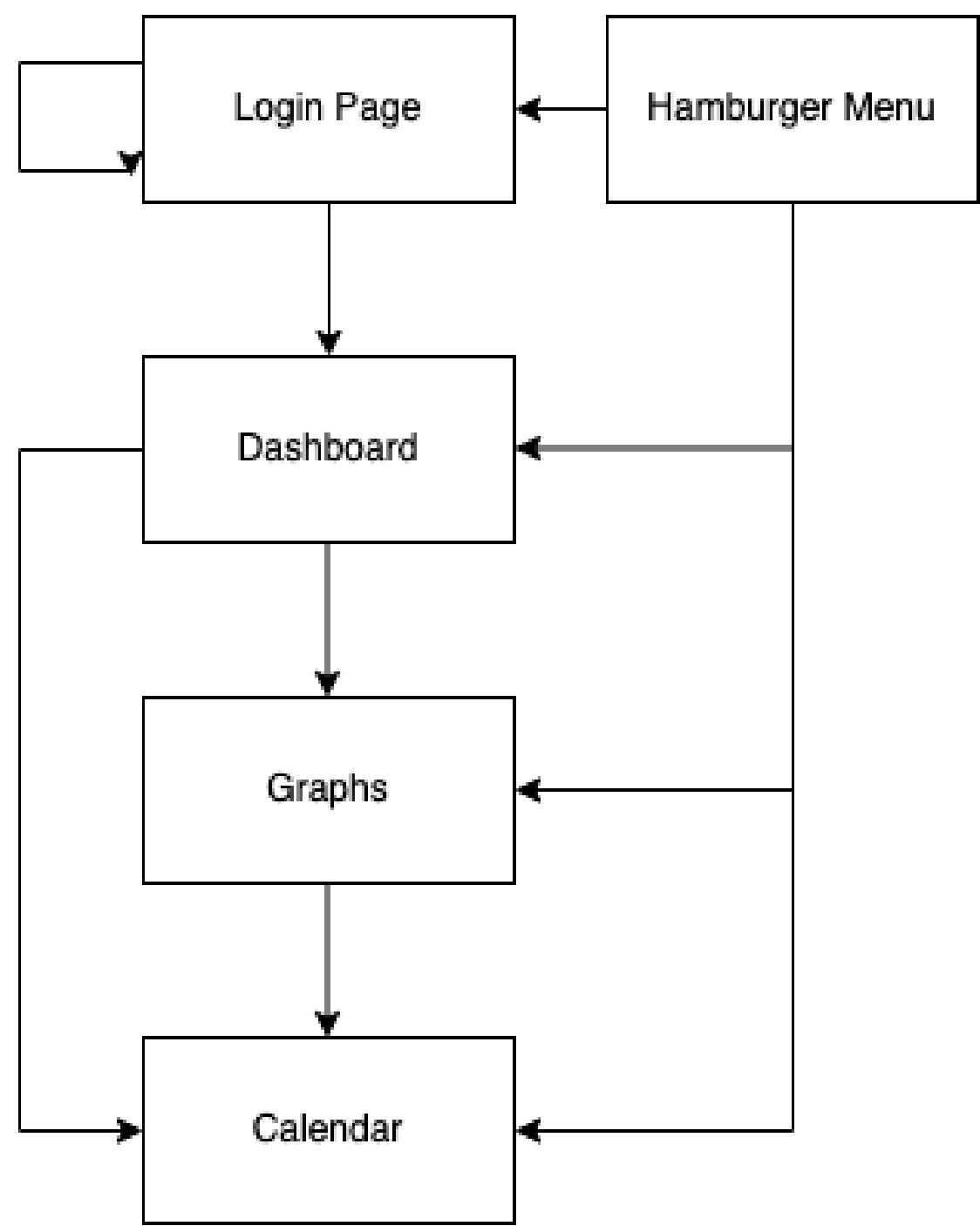
Figure 1: Block diagram of the applications page interaction and reachability. The display shows the page layout, and the access that the hamburger menu has across the application. Menus identified in blue, are reachable by the hamburger menu.

Purpose

The purpose of Total Life Manager is to provide a service where a user can track their health habits and information. This service will allow users to see their data and display it in a digestible system of charts and widgets. The daunting experience of working on health and habits is eased when introducing the ability to visually understand trends and data. The entire goal of the service is to give users a method of approaching their personal health goals and identifying positive habits with the tools provided by the service. This will aid users in tracking their data and reaching the goals they set. The purpose the platform serves caters to a wide audience, whether experienced in health goals and achieving milestones, or not.

Technology

The technology the system is predicated on has its foundation rooted in the LAMP stack. The LAMP stack is comprised of Linux, Apache, MySQL, and PHP. The components each have their own characteristics and avenues of change that make the system function in its totality. To start, the Linux component is key to the server and the hosting service the project lives on. Currently, the server resides on an Ubuntu Linux distribution that we have hosting the Apache server. Naturally, next comes Apache in the stack, which is the web server, and is the most common web server for Linux powered web applications. A notable backend process is the MySQL component in the stack, which is hosted on the Apache server. The server hosts a MySQL database which is critical for the operation of the web application. In terms of the stack, the final component is PHP. This is the server language that communicates between the frontend and backend, allowing the stack components to all work together.

The technologies used to power this product are as follows:
1. Platform: Firefox
2. Operating System: Windows 10
3. Languages: HTML, CSS, JavaScript, PHP, MySQL
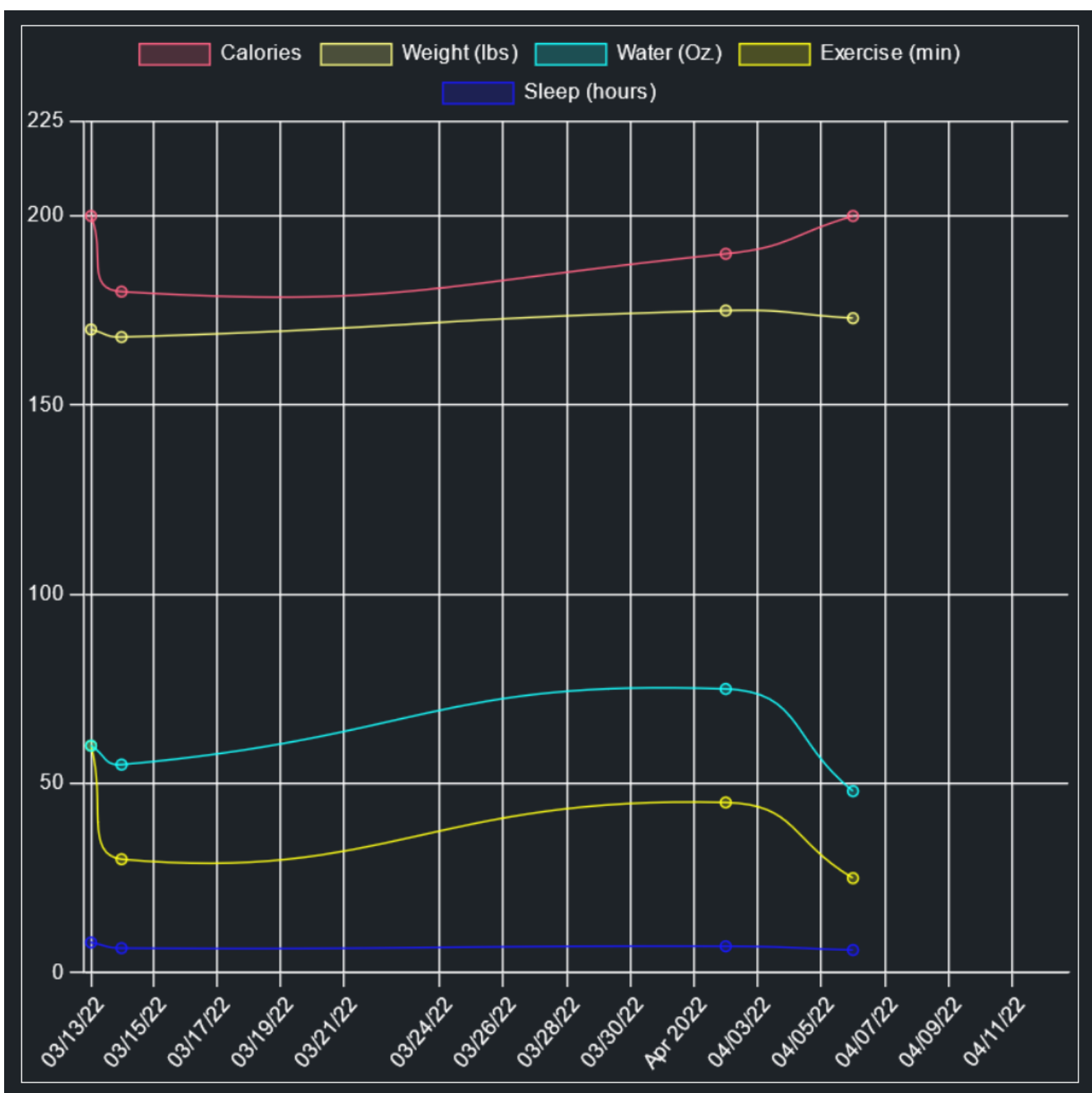4. Libraries: Chart.js

Figure 2: This capture of the systems chart module displays the technology in use for the website. Currently, the graphs display all entered data in varying linear graph representations. The graphs also include a multitude of components directly from Chart.js, such as a variety of views and display modes.

Design

The system's overall design is a web application that hosts a number of services for a user to track their health and habits. This service will be centralized around a dashboard where the user can see the majority of their information at a quick glance. The individual dashboard modules will come in a variety of different widgets placed around the screen. The main widget will display the users exercise, hydration, sleep, and calorie intake. The exercise will be in minutes, the hydration in ounces, the sleep in hours, and the calories in calories. These will display rings that show a percentage of completion for the user's individual goals. The dashboard will also display a widget for the user's graphs. This shows a preview of a user's graph and leads directly over to the graph page. The next widget is a weather API that displays basic weather information for a user so they can plan their day. Lastly, there will be a widget for a calendar. The user will have a basic view of a calendar and be able to go to a full view that shows the user slightly more detailed information about events or other data that has been placed on the calendar. The entire website will be reachable through a convenient hamburger menu on the top of the page. This will provide navigation to the all the pages a user can access on the website, including a logout option to log the user out of the website.
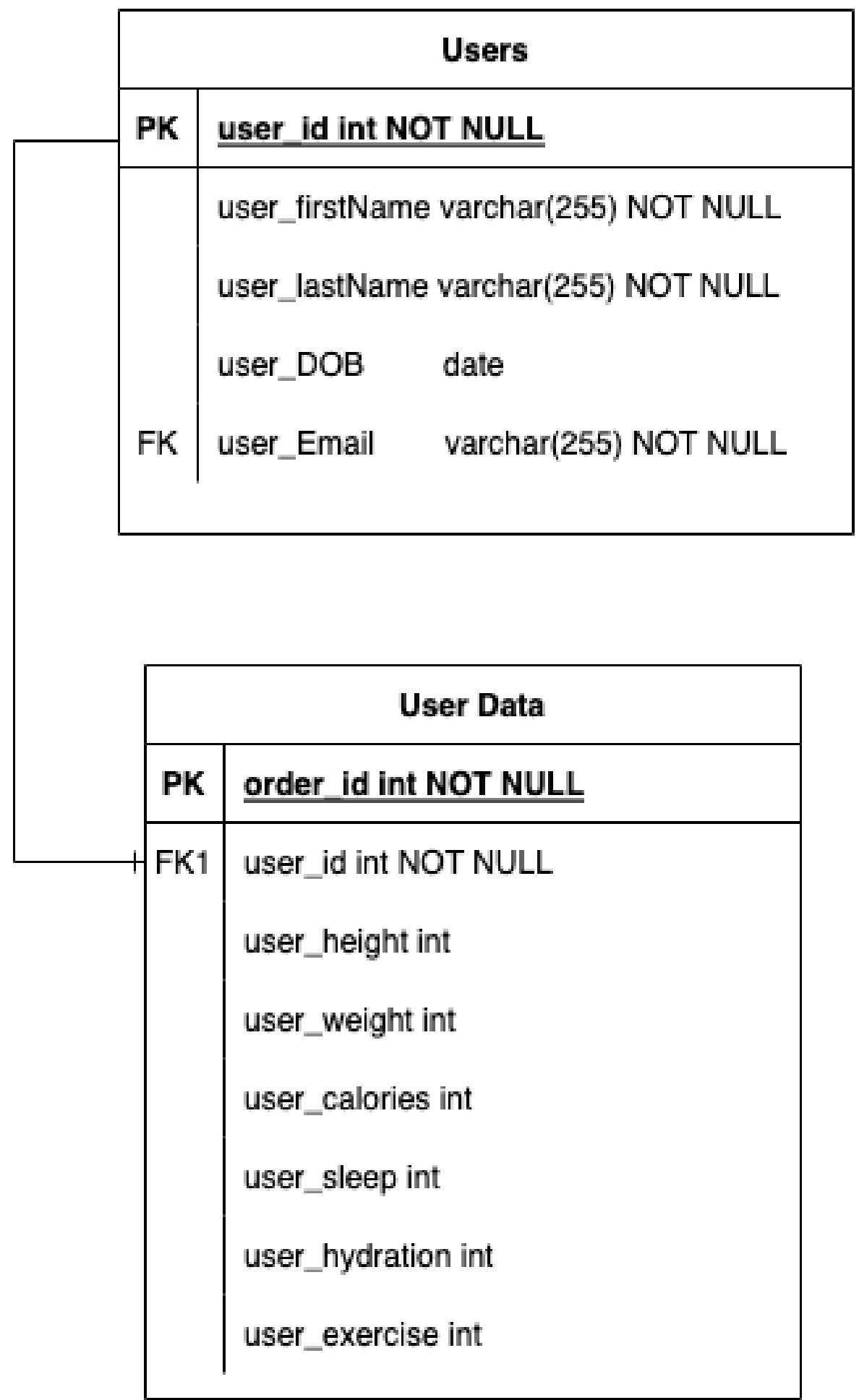

Figure 3: The data is used in a series of user data entries to the system, which stores an event that the user can see on their graphs, and also fills the rings on the dashboard.

Future Work

The future work the team intends to implement revolves around the core services, as well as ancillary modules that will further the user's experience. The core changes to the system would be the introduction of support on the website, through a chat or email system to the team. After this, the main changes would be to the user's personal account on the website, as user account customization is limited in the current state of the system. Ideally, the user would be able to change all of their information regarding location and unique profile identifiers, such as a profile picture or avatar. The extended functionality planned for the service mostly revolves around the introduction of new API's to services. For example, the service originally planned to have an extended nutrition section that would include food data form the USDA's nutritional information API.
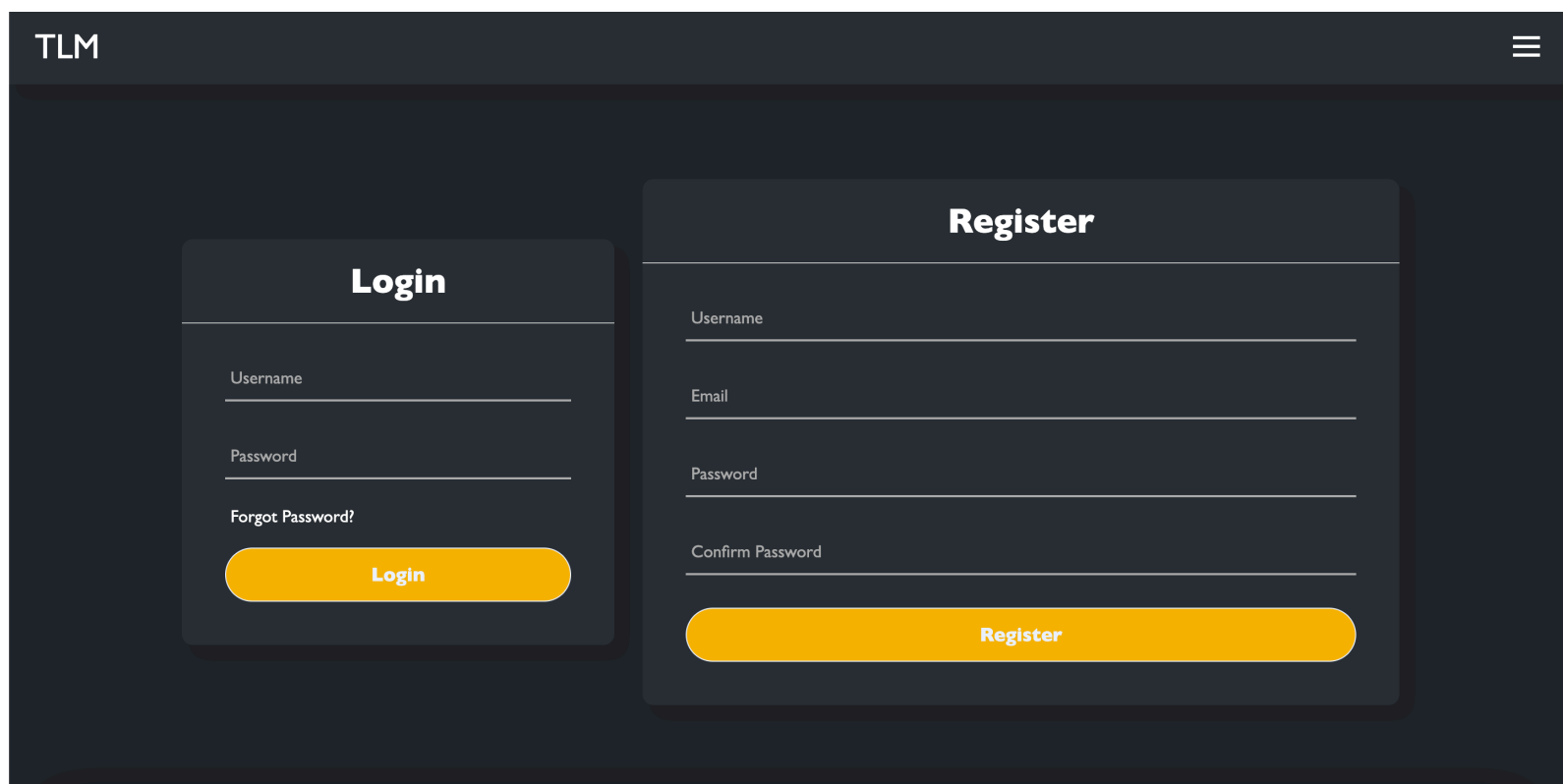

Figure 4: The login screen is an accurate representation of the website's styling and design. For future work, the screen was proposed to have moving background images in the form of a slideshow.

Resources

1. Linode: https://www.linode.com/
2. Chart.Js: https://www.chartjs.org/
3. Ubuntu: https://ubuntu.com/
4. Apache: https://www.apache.org/
5. MySQL: https://www.mysql.com/

Acknowledgements