**Engineering Fall 2021**

**[FAU: CEN4010]**

**ETL (Easy to Lift)**

**Group:13**

Muhammad Dagra (**Team Lead**)

**Team Members**  
Julio Berrocal

Jamal Grant

Madan Shrestha

Joao Staziaki

**Milestone 4**

**November 15, 2021**

**Product Summary**

**Project name and location**: E2L: Easy to Lift; <https://lamp.cse.fau.edu/~cen4010_fa21_g13/>

**YouTube link:** [**https://www.youtube.com/watch?v=VtaJAs2Z0D4**](https://www.youtube.com/watch?v=VtaJAs2Z0D4)

During the COVID-19 pandemic, staying in shape and keeping a healthy lifestyle became increasingly difficult. If you relied on a gym, then for a while you had no option but to try to figure out a home-workout with no direction. So, we decided to provide that direction. Easy to Lift is a home-workout and exercise tutorial aggregator/distributor and discussion space. Whatever the goal for your body is, whatever level of intensity you prefer or need, E2L aims to give you the tools to take your own health into your own hands. Just make an account and join the community!

**Committed Functions:**

* **Sign-up** – The site allows users to sign up by registering an account on the website; Users provide their first name, last name, email address, and a password. This information is logged into the database and a registered account is created from it.
* **Sign-in** – If a user has already signed up with the website, they are able to log in as a registered user by providing their email and password. Users are then able to view the content provided on the website as a registered user.
* **Password Encryption** – Using a PHP hash-function, passwords provided by users are encrypted before being stored in the database. When a user types in their password to sign in the system compares the “hashed” password to the one stored in the database, so no plain-text passwords are stored to be stolen.
* **Videos** – Workout video tutorials are embedded into the website from YouTube.
* **Comments** – Underneath the videos, users can upload comments which are displayed for other users to see and also to comment on themselves. (Under development)
* **Home Page** – This is the first page of the website and sets the tone and gives a general explanation of what the website is for; The main purpose of the website is to hook users’ attention and to serve as navigation hub to the other pages.
* **Services Page** – The services page is a navigation and aggregation page for several workout videos which are also in separate pages.
* **Our Mission Page** – This is an “about” page that explains in detail the purpose of the project.
* **Registration Page** – Serves as both the page to sign up with a new account or to sign in with an account that has already been registered. Switching between the two functionalities reveals a stylish transition-animation.

**Usability Test Plan**

**Background:**

Due to the corona virus, we intended to build a website that provides workouts that did not require any equipment and can be done from home. The targeted market is the people who do not have access to the gyms and proper workouts.

The testing was done wo we can measure the points stated below.

**User-friendliness:**

User-friendliness measures if people can understand the user interface and easily complete the tasks. When the user enters the (URL: <https://lamp.cse.fau.edu/~cen4010_fa21_g13/>), they are welcomed to the E2L home page where they are greeted and are asked to sign up.

To Sign-up, they are required to click on the icon that says Sign-up. The user is then taken to a form where they are required to put their first name, last name, and the email. After doing so, they are then required to login. After successfully logging in the system, they are redirected to a page that provides the Services that E2L has to offer, the user can choose what exercise they want to do. For example, if the user chooses to work out the back they are then redirected to the page where they are provided with a back workout that is easy and effective. At the very bottom of the page the user has the option to drop a feed back whether the workout was effective or not (Comments Development still in Progress). In the end, once the user has done the workout, they may Log out if the system.

**Efficiency:**

If user-friendliness focuses on how users navigate the interface and complete the tasks, efficiency measures how quickly they can do it. The less amount of time it takes to finish a task the better.

Using the example of signing up then working out, to leaving the suggestions (Comments still under development) we measured the time taken by the user to do these tasks.

This also shows which part of the tasks takes the most time. If participants spend too much time on one task, there may be a part of the interface that confuses them and needs improvement.

**User Satisfaction:**

Even if the participants can complete the task without errors, they may still provide insights to help improve user satisfaction.

The user may suggest a better visual design, color scheme, text and fonts, the correctness of the information provided.

**Test Plan**

**Black Box Testing:**

The testing method that we applied for our website is the black box testing. Our tester had no knowledge about the internal structure of the website. However, our tester had a computer, good internet connection and the knowledge of how to work on the computer. The tester was given the URL: (<https://lamp.cse.fau.edu/~cen4010_fa21_g13/index.php>) to test the friendliness, robustness, functionality, the login, signup, logout and the suggestion/ comments.

**Starting Point.**

The starting point was the home page where we greet the user and ask them to sign-up/ login.

**Tasks To Be Accomplished:**

The tasks to be accomplished were if the tester can easily sign up/ login or not. After sign-up/ login if the workouts are easily available to the tester. Are the videos executable or not?

The intended user is affected by the coronavirus Lockdown and is not able to leave the house to go to the gym and workout.

The completion criteria here is, is it easy for the tester to navigate through the whole website? Are the workouts functional? Can the tester easily leave suggestions/comments after each workout? (Comments Development still in Progress).

**URL Of The System:**

URL: (<https://lamp.cse.fau.edu/~cen4010_fa21_g13/index.php>)

**QUESTIONAIRE FORM:**

Please answer the questions below with the **check** mark in the boxes provided for the questions.

|  |  |  |  |
| --- | --- | --- | --- |
|  | SATISFACTORY | GOOD | EXCELLENT |
| HOW IS THE LAYOUT OF THE WEBSITE? |  |  |  |
| HOW IS THE LOG-IN / SIGN-UP? |  |  |  |
| IS THE WEBSITE ROBUST? |  |  |  |
| IS THE WEBSITE USER FRIENDLY? |  |  |  |
| WERE YOU SATISFIED WITH THE WORKOUTS? |  |  |  |
| WERE YOU ABLE TO LEAVE A COMMENT/SUGESSION? (UNDER DELEOPMENT) |  |  |  |

**Quality Assurance test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Action (Test Title) | Test input (Test Description) | Expected Results | Pass |
| 1 | Sign up | Once the “Register” Icon is click you are prompt to enter your sign-up information | Arrive at “Sign up” |  |
| 2 | Login | Once successfully signed up, use the same information to login on the website | Can successfully login |  |
| 3 | Home | Once the “Home” Icon is clicked you are brought to the website’s homepage | Arrive at “Home” page |  |
| 4 | Services | Once the “Services” Icon is clicked you are brought to the services page | Arrive at “Services” page |  |
| 5 | Our Mission | Once the “Our Mission” Icon is clicked you are brought to our mission’s page | Arrive at “Our Mission” page |  |
| 6 | About Us | Once the “About Us” Icon is clicked you are brought to the about us page | Arrive at “About Us” page |  |
| 7 | Videos | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. | Arrive at the videos section of the services page |  |
| 8 | Videos/workout (Cardio) | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. Click the video labeled cardio. | Video can be executed on the click of play |  |
| 9 | Videos/workout (Arms) | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. Click the video labeled arms. | Video can be executed on the click of play |  |
| 10 | Videos/workout (Chest) | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. Click the video labeled chest. | Video can be executed on the click of play |  |
| 11 | Videos/workout (Core) | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. Click the video labeled core. | Video can be executed on the click of play |  |
| 12 | Videos/workout (Back) | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. Click the video labeled back. | Video can be executed on the click of play |  |
| 13 | Videos/workout (Legs) | Once the “Services” Icon is clicked you are brought to the services page. Scroll down and click from a variety of specialized videos. Click the video labeled legs. | Video can be executed on the click of play |  |
| 14 | Comment/feedback | Under development | Under development |  |

**Code Review**

**PHP code used for log-in / sign up; from functions.inc.php**

The following is an excerpt of PHP code from functions.inc.php referenced by signup.inc.php, or the registration page of the project. This is a group of functions that takes user-provided information: names, emails, and passwords, and then uses them to create user-profiles and store them securely in the MySQL database for the website.

These functions assume that the information is correct and not blank, as prior in the file there were methods used for checking if that information is valid. As the information is valid, these functions create users by storing them in the database and encrypts the password of the user using a hash-function. If the user tries to log in, the system compares the hashed password with the actual encrypted password. No plain-text passwords are ever stored.

The code, as it is organized and written in the actual file is spaced out in a comfortable and easily readable way. The way that it is formatted in this document makes it less difficult to read and understand than in the file. As someone reviewing and not writing the particular program, the functionality was very easy to understand. However, the code could use with more concise commenting explaining the functionality of each method.

The code style used in this excerpt seems reminiscent of what might be called PSR-2, which is the extension to the Basic PHP Coding Standard. Some examples of elements of this coding style are the camelCase naming convention, use of 4 spacebar clicks instead of tab for indentations. However, the style of this code breaks with conventions associated with the standard. For example, the PSR-2 standard requires that there be a new line before you type in a curly-brace. In addition, method and class definitions do not follow the strict rules of the standard. Overall, the style is readable and elegant. It's a standard style that works very well and is consistent throughout.

function createUser($conn, $firstName, $lastName, $email, $password) {

$sql = "INSERT INTO Registration (firstName, lastName, email, password) VALUES(?, ?, ?, ?);";

$stmt = mysqli\_stmt\_init($conn);

if (!mysqli\_stmt\_prepare($stmt, $sql)) {

header("location: ../signup.php?error=stmtfailed");

exit();

}

**// this function, assuming that the elements are either not taken prior // or left empty, creates a new user and stores them into the mysql // database**

$hashedpwd = password\_hash($password, PASSWORD\_DEFAULT);

$stmt->bind\_param("ssss", $firstName, $lastName, $email, $hashedpwd);

$execval = $stmt->execute();

echo $execval;

//echo "Registration successfully...";

$stmt->close();

$conn->close();

header("location: ../signup.php?error=none");

}

**// this function takes a user’s password provided during sign-up and runs // it through a PHP hash which encrypts it, and then that is saved**

function emptyInputLogin($email, $password) {

if (empty($email) || empty($password)) {

return true;

}

else {

return false;

}

}

Function loginUser($conn, $email, $password){

$emailExists = emailExists($conn, $email);

if ($emailExists === false) {

header ("location: ../signup.php?error=wronglogin");

exit();

}

**// checks for if the email provided during user login exists or not**

$sql = "SELECT `password` FROM `Registration` WHERE `email` = '".$\_POST['email']."'";

$result = $conn->query($sql);

$row = $result->fetch\_assoc();

$pwdHashed = $row["password"];

$checkPwd = password\_verify($password, $pwdHashed);

if($checkPwd === false){

header ("location: ../signup.php?error=wrong");

}

**// given the prior information is correct, this seelcts the password // from the registration table, and corresponding email, and compares // it to the hashed password provided during login, if // the two // match, then the login is successful**

else if ($checkPwd === true){

session\_start();

$\_SESSION["email"] = "email";

header("location: ../services.php");

exit();

}

}

**Self-check on best practices for security**

1. List major assets you are protecting.  
     
   As more cases of security breaches occur day by day, it is important for us to ensure the security of our client’s information as it is our major asset to protect. The two most important being the following:  
     
   - User’s Personal Information: Includes Name, Last Name and Email.  
   - User’s Password: User’s password are encrypted and difficult to hack.
2. Confirm that you encrypt password in the DB  
     
   For our password encryption we have used the PHP native hashing method called password\_hash(). This algorithm is a strong one-way hashing algorithm that will keep our client’s passwords difficult to hack and unreadable in case of breach. Here is a snapshot of our current database table Registration. Graphical user interface, text, application

   Description automatically generated
3. Confirm Input data validation (list what is being validated and what code you used)  
     
   One input validation we are using is email validation. When a client tries to sign up for our website, an email address is required. We ensure that the email can only be used by that one client. In terms of code used for this validation we have a function that calls to the database to compare the email with the ones already used. If the comparison returns true, then the client is shown an alert informing them that the email is already in use. Otherwise, it calls the create user function.

**List of non-functional requirements**

**Performance Requirements**

**Responsiveness** – System will be responsive, operating on various monitor sizes and will be responsive to a variety of resolutions.

**Cycle Time** – The cycle time at expected performance will be 1.0. With this in mind, the system will operate between 1.0 – 1.2 with a load of 5-10 concurrent users or a slight lag. The system will operate with a 1.21 – 1.3 with a load of 11-25 concurrent users oro a moderate lag. The system will operate with a 1.31-1.5 with a load of 26-45 concurrent users or a heavy lag. Finally the system will operate with 1.51-1.7 with a load of 46-50 concurrent users or a very heavy lag. Any number of concurrent users over 50 will cause the systems performance to halt briefly until a user finishes.

**Speed per Transaction** – 20-100 milliseconds, depending on the cycle time. The system will process 10-50 transactions per second.

**Test Requirements** – the test requirements for performance will include an expected load test as well as testing on all of the functional specs listed and their speed per transaction.

**Reliability** – Mean time between failures is that it must have 1 hour or less of downtime in a total of 3 months. This downtime can be used to perform maintenance and update information. The system should be operation 100% of the time for the first 99.8% of the calendar for the first year of its operation.

**Minimum Bug Counts** – No more than 5 bugs in the system during integration and testing, no more than 3 can remain after delivery

**Execution Speed** – 100-200 milliseconds depending on the current cycle time, at the initial home page on a high-speed internet

**Storage Utilization** – 75-90% of available storage provided at the time ast o not get too close to using all storage and having a technical issue if more storage is needed.

**Robustness** – Under an hour to restart after a failure.

**Usability Requirements**

The ease of use will be a top-priority of the website, as navigation and functionality is to have minimal training time. Site design is user-friendly and intuitive

Interoperability Requirements

**Browser Compatibility** – System is a website that operates on at least two of all the major browsers including google chrome, mozilla firefox, safari, opera, and internet explorer. It will have functionality that will provide alternatives if the browser does not have javascript installed in it.

**Computer and OS Compatibility** – System operates on various types of operating systems, including Windows, OS X, and Linux. Also operates on any type of computer which can run a browser which is supported.

**Accessibility Requirements**

All content uploaded and the website will be accessible to persons with disabilities as there will be options and settings as well as content catering to those needs.

**Expected Load**

There are allowances of up to 50 users at the same time. Load testing will be done to record performance times during periods of high traffic, both in a continuous and in a spiked pattern

**Storage Requirements**

The storage for our system will consist of the lamp server holding our mySQL database within an unknown capacity as well as holding our files for the actual site.

**Security Requirements**

**Login/Password System** – The system has a login and password system to regulate content uploads.

**Encryption** – Passwords and other content are to be encrypted as per the safety and privacy concern of the users. Regular users use the same passwords for multiple websites so there will be no plaintext storage of passwords or private user information.

**Access Control** – Editing of front and back end code will be available to anyone of the development team. Users and developers with moderation privileges will have access to some information that regular users don’t have access to. Users and visitors will have a limited access to using the system based on the user interface.

**Spam Protection** – The site will utilize a captcha system to create an account, prevent bots from spamming the site and creating fake accounts.

**Resource Utilization** – Resources will be accessed with the usernames and passwords of special development accounts.

**User Safety and Moderation**

User-uploaded content will be moderated for its content. Anything deemed to be harmful or dangerous or malicious by the moderation team will be removed with notice to the user. Users may report content that breaches the safety and conduct standard. User safety and privacy is to be a main concern.

**Availability**

**Accessible times** – Site will be accessible 24 hours a day, 7 days a week and will be up and running for as long as the lamp server is available.

**Downtime impact** – Minimal downtime but when it is necessary, the system will inform users that the site is undergoing maintenance and it will also let the users know when the site with undergo maintenance in the future.

**Support** – Support availability will be via the email provided in the About page of the website, responsive within 24 hours.

**Fault Tolerance**

Should an exception occur, an explanation will be provided to the user and give them a chance to input the correct answer, or they will be taken back to the home page. Self-checking software won’t be implemented as it’s not necessary.