State University of New York at New Paltz

Nicholas Agneta

Project Type: Internship, Student’s “Class Key”: p-f18-01

“One EPIC Place Website Scheduler”

FINAL PROJECT REPORT

Computer Science Projects

Fall 2018

(Prof. Hanh Pham)
TABLE OF CONTENTS

1. Problem Description
   1.1 Business Context and Goals ........................................ page 03
   1.2 Technical Requirements ............................................. page 03
   1.3 Your Responsibilities ................................................ page 04

2. Technologies
   2.1 Related Technologies ................................................ page 05
   2.2 Newly Learned Skills/Technologies ................................ page 05

3. Design
   3.1 System Architecture ................................................ page 06
   3.2 Components ................................................................ page 08

4. Software/System Description ............................................. page 09

5. Test Results/Observations
   5.1 Experiment/Observation # 1 (Performance) ....................... page 17
   5.2 Experiment/Observation # 2 (Quality) ............................ page 20

6. Professional and Career Benefits ....................................... page 20

7. Conclusions ..................................................................... page 21

8. References ........................................................................ page 23

9. Appendixes ....................................................................... page 24
1. Problem Description

1.1 Business Context and Goals

One EPIC Place is a coworking space rental service located in New Paltz, New York. The co-founders Julie Robbins and Nicole Langlois rent out space in several different buildings to provide offices for individuals, groups, and classes to work, often alongside each other. They also provide classes about business management among other things to their members. They currently have their own website which houses information about their business at “oneepicplace.com,” but use a service called “YaRooms” for the actual reservations and bookings of the rooms. They use YaRooms because it has a point system, which they use as an exchange rate for membership fees and booking fees, as well as five-minute increments for the actual room bookings. Both of those features are useful to One EPIC Place. Although YaRooms does have many nice features that they can use, YaRooms does not currently have many of the customizable features that they want such guest booking or the ability to run reports.

1.2 Technical Requirements

The system created uses web-based programming technologies connected to a database. The processing for the software includes user registration, user login, logout, a profile page, a log of booked rooms, and the scheduler for booking rooms. The registration uses user information from the forms including, the users’ full name, email, password, occupation, and a description of their business which is then processed into the database. Login retrieves that data from the database and matches it against user input to create a session and log them in to the website. The session houses temporary data such as the user information for the website to use. The profile page retrieves information created by the user during registration that is held in the database and
displays it for the user to see. The scheduler allows the user to book a room using a variety of form entries, all the information entered is stored in the database. The log of booked rooms retrieves that data to show the users which rooms have already been booked. The user will then have the option to logout destroying the session and returning the user to the home page.

1.3 Your Responsibilities

There were many moving parts to this project and several things that had to be created and looked after simultaneously. My focus was centered around the server-side programming. I mainly dealt with how the PHP code could store user input and retrieve the information from the Database. I did this mainly with POST requests (see Appendix A) to simplify the program a bit. Max mainly dealt with the client-side code. Originally Andrew was supposed to oversee the Database, but once he left our project Max and I both oversaw that aspect of the software as well. With Ryan’s help we all looked after and developed each of the components of the software; we didn’t exclusively stick to one aspect of the software individually. This was beneficial in case any of us got stuck we could help each other out; we could learn a lot about all areas of the program and how each part works. We all had a hand in talking with Julie and Nicole about what aspects of the software they wanted to see in the final product. I had several questions for them after our initial interviews with them in the beginning of the semester to acquire more information about the project and how the final version of the software should operate.
2. Technologies

2.1 Related Technologies

For this project we used several different outlets to develop the software. Our primary source of collaborative software was the use of GitHub and GitKraken. We created a repository on GitHub and shared it between the three of us, then on our own time we could edit the code and push it via GitKraken. We ran into some difficulty using the shared code, for example, when someone would try to push new changes without first pulling older changes it could override new additions to the code. This created a few errors in our code that had to be straightened out individually. Eventually we would just communicate to each other after each change was made so everyone was on the same page.

For our code we used HTML, CSS, JavaScript, and PHP for the Front-end development and PHP for the Back-end development. We accessed the database stored on the SUNY New Paltz server via phpMyAdmin. Once changes were made to our code we would not only push it to GitHub via GitKraken, but we would store the files on the school’s Computer Science server in various folders. We used WinSCP to access the files stored on the server from our own machines and stored the project files in the “var/www/projects/f17-08/vX” folders; the X was the version number for each major change made with the final version being v29, thus the full file path is: “var/www/projects/f17-08/v28.” The project can be accessed online by visiting https://cs.newpaltz.edu/p/f17-08/v29.

2.2 Newly Learned Skills/Technologies

This experience was beneficial to us because of how many different aspects of programming and software design we were able to learn. I learned a lot about how databases
operate and learned PHP from scratch. I also did some research into the differences between
GET and POST requests to figure out what was most beneficial for our project given the
timeframe.

I was able to learn more about how to interact with customers and get some hands-on
experience at an internship. The amount of self-teaching for this project was particularly
daunting but I manage to find many resources that helped us along the way. Being able to reuse
code from other sources would have been nice, however the ones we found online were either
not easily integrable, not what we were looking for, or only available as a pay-to-use option.
Therefore, we had to create all the assets from scratch which was time consuming but allowed us
to learn more about JavaScript and CSS.

3. Design

3.1 System Architecture

There are a few different layers to the complete design of the software. A representation
of the architecture can be found in Figure 1. In terms of users we have designed the website to
account for four different types of users: admins, staff, members, and guests. Guests would have
the least amount of privileges, only being able to book a room. Members would have more
privileges than guests; members would be able to book a room, keep their login information
stored, and update payment methods via member tiers and points. Staff would have the second
highest amount of privileges, allowing them to manage members and book rooms like members
would be able to. Admins would have the most privilege being able to change and manage other
users as well as be able to do anything a staff member could. Currently the system is set up to
only allow guests to register and book rooms.
The user interface is entirely written in HTML, CSS, JavaScript, and PHP. This allows the user to interact with various forms and input data to book a room using the scheduler. The user can click between various menus to see the home page, about page, scheduler, and a list of currently booked rooms. The forms that the user can input data into are then processed by the next level of the system, “Processing.” Each PHP file that the user can interact with has a separate PHP file associated with it in the “php_methods” folder. The files in the folder contain all the information for processing the data that the user enters. The information is sent to the database where it is stored and used for retrieval in protocols such as login and viewing the booked rooms list.

Although the system is web-based and should be able to run on any development machine, provided the correct information regarding the database is changed, the software is currently stored on the “cs.newpaltz.edu” server. The University’s server allows us to store the files that make up the website and share files between different members of our team. One EPIC Place currently uses DreamHost as their server to house their website; the software will be ported to DreamHost for launch and integrated into their current website when development is over.
3.2 Components

There are three separate types of components for this software: Front-end, Back-end, and the Database. The Front-end component is the main client-side code. Using HTML, CSS, JavaScript, and PHP the user can interact with the different parts of the website. The user can navigate through pages by clicking and the user can fill out forms by clicking and typing the information into the forms. The second part of the software is the Back-end component. The Back-end component is the server-side programs that allow for the transfer of data between the client and the Database. A representation of the database and its tables are outlined in Figure 2. The Back-end component is written entirely in PHP. This component allows us to not only
transfer and check data as needed, but also create sessions for the users to collect temporary data without having to make several calls to the Database. The third and final component is the Database. The Database is comprised of several tables to store all the user information entered in the various forms. The Database allows us to have a record of all the people who register on the site and who book rooms.

4. Software/System Description

Each of the three components, as described previously, are outlined with various different files. The Front-end component includes several PHP files that act as an interface for the user to interact with. They are, about.php, index.php, login.php, profile.php, register.php, scheduler.php, and viewreserved.php. The about.php file contains information regarding the service and a link to the main website: “oneepicplace.com.” A representation of the about protocol can be found in Figure 3. The index.php file contains the home page for the software. Upon logging in successfully, the user is greeted with a personalized welcoming on the home page. A
representation of the home protocol can be found in Figure 4. The login.php file allows the user to log into the software and the prompts the software to create the session which is used throughout the rest of the website. The profile.php file displays various information about the user after signing in. A representation of the view profile protocol can be found in Figure 5. The register.php file allows someone to register and have their information stored in the database. The scheduler.php file holds the main program, which is a series of dropdown forms and date selection tags. The viewreserved.php file contains the display information for all the previously booked rooms.

Along with the Front-end component files, the software also contains Back-end component files which handle all the processing for the various input forms. Those files include, booking.php, connect.php, login.php, logout.php, register.php, and session.php. All of those files are found in the php_methods folder. The booking.php file is directly linked to the scheduler.php file in the Front-end component. The booking.php file handles all the POST requests to send the information inputted by the user in to the “booking” table in the database as well as displays a message in the URL bar to congratulate the user on their successful booking. The booking.php file also handles redirecting the user to the “View Schedule” page once the form input data is complete, as well as simple error handling such as checking for empty fields. A representation of the booking and view schedule protocols can be found in Figures 6 and 7 respectively. The connect.php file houses all the information for connecting the software to the database; this information can be used for all of the form processing application of the website.

The login.php file in the php_methods folder handles the POST requests for checking the input data against the users table in the database to see if the software should allow the user to log in. The file also handles the error checking to make sure there are no problems while the
software is running. A representation of the login protocol can be found in Figure 8. The login.php file then sends the respective login_id field to the session.php file so the session can start. The session.php file takes the login_id field provided by the login.php file and checks the id against the “users” table in the database. It then creates a session based on the information in the database and stores it as temporary variables while running. A representation of the session protocol can be found in Figure 9. The logout.php logs the user out of the website and ends the session. The logout.php file then returns the user to the home page. A representation of the logout protocol can be found in Figure 10. The final PHP method file is the register.php file. The register.php file handles the POST requests created after the user submits the input data. For new users the register.php file sends the login information to the session.php file so the session can start, and the user is automatically logged in. A representation of the register protocol can be found in Figure 11. A complete visual representation of the entire website can be found in Figure 12.

Along with processing POST requests and error checking the software also houses the look of the website based on CSS files stored in the “css” folder as well as assets used for One EPIC Place such as their logo which can be found in the “assets” folder. Some other functionalities of the software are found in the components folder. They include the files: welcome-user.php, header.php, nonuser-navbar.php, and user-navbar.php. The welcome-user.php file allows the home page to display a personalized greeting to the user after they have logged in successfully. The header.php file contains the information used in the header of the website such as displaying the name of the company and the logo. The nonuser-navbar.php file is the navigation bar the user will see before logging in. It includes options for the user to visit the home page, about page, register page, or login page. Once the user successfully logs in they will
see the navigation bar created by the user-navbar.php file. The user will then have the option to choose between the home page, their profile page, the scheduler page, view schedule page, the about page, and have the option to logout.

Figure 3. A visual representation of the about protocol.

Figure 4. A visual representation of the home protocol.
Figure 5. A visual representation of the view profile protocol.

Figure 6. A visual representation of the booking protocol.
Figure 7. A visual representation of the view schedule protocol.

Figure 8. A visual representation of the login protocol.
Figure 9. A visual representation of the session protocol.

Figure 10. A visual representation of the logout protocol.
Figure 11. A visual representation of the register protocol.

Figure 12. A visual representation of the design of the software.
5. Test Results/Observations

5.1 Experiment/Observation # 1 (Performance)

We created a variety of tests and checks to make sure our software runs correctly. We were able to try to register different users providing various types of information. For example, the software does not allow a user to register if their email does not contain the “@” symbol between two strings of characters. Users can register with misspelled email addresses which is something to address in future versions of this software. Multiple users can be logged in at the same time provided they are using different computers or different browsers to do so, otherwise the session will just be reloaded upon the second login attempt. An example of how the user can input data into the registration page is shown in Figure 13. The data the user enters for registration can be found in “users” table which is outlined in Figure 14. The user can the book a room using the “Make A Reservation page” which is outlined in Figure 15, that data is then stored in the “booking” table in the database. The “View Schedule” page retrieves the data from the “booking” table and displays it for the user. A representation of the “View Schedule” page and the “booking” table can be found in Figures 16 and 17 respectively.
Figure 13. An example of user input for registration.

Sign Up

Email
agneta1@hawkmail.newpaltz.edu

Password
***

First Name
Nicholas

Last Name
Agneta

Occupation
Student

Description
Here is some information about myself.

Sign Up

Figure 14. A visual representation of the “users” table using phpMyAdmin. The registration example shown in Figure 13 is also shown in the 12th entry to the table.
Figure 15. An example of user input for the “Make A Reservation” page.

Figure 16. An example of the “View Schedule” page. The booking example shown in Figure 15 is displayed as the bottom row in the table.
Figure 17. A visual representation of the “booking” table using phpMyAdmin. The booking example shown in Figure 15 is also shown in the 38th entry to the table.

### 5.2 Experiment/Observation # 2 (Quality)

We have tested the application on several different devices and noticed that it scales well with desktops as well as mobile platforms. We have noticed that there tends to be some crowding in the view schedule page on mobile devices but, the information overall is readable. We have noticed that the sessions on the website will not run unless the website URL is appended with “https://” before the rest of the URL. If it is started with “http://” it will not work. The website will load regardless of the beginning characters, but the session will fail to load on login without “https://” thus users should use the following link: [https://cs.newpaltz.edu/p/f17-08/v29](https://cs.newpaltz.edu/p/f17-08/v29).

### 6. Professional and Career Benefits

This opportunity has helped us greatly in learning web development and software engineering. Being able to work with clients on an actual project that will eventually be used in the real world is a very rewarding experience. This process has allowed us to explore the many
facets of web development as well as allowed us to try on many hats as we discover they types of development we would like to continue to do in the professional world. Being able try our hand at Front-end development and then move into Back-end development while learning about databases has been especially useful and has provided us with a well-rounded real-world experience of working on a software development project with a team of people.

Although I did take Web Server Programming, the graduate level web development course offered at SUNY New Paltz, it did not teach me all of the skills necessary for this project. The Web Server Programming course was more aligned with using technologies such as JQuery and Angular rather than focusing on PHP or POST requests. Although there are many ways of accomplishing the same goal, especially when it comes to web programming, the fundamentals are important, and I believe that the Web Server Programming course missed a lot of the key components of developing web-based software such as integrating databases, sessions, and storing user input. The class was beneficial in exposing us to the wide array of different options we have in terms of web programming and giving us a taste of several different avenues to be explored later. This project allowed us to take some of those routes and expand our knowledge surrounding them.

7. Conclusions

Although we have made great strides in completing many parts of the project, there is still much more work to be done before the software is ready for public use. We would like to implement an email verification method that would prompt the user to verify their email before being immediately logged in; that way no one can register with a non-existent or incorrect email. Secondly, we would like to implement membership tiers and the point system - rather than a
traditional currency-based payment system - to allow the members to refill their points or have them rollover based on the specifications of One EPIC Place’s business policy. We would also like to eventually see a better visual representation of the booked rooms so users can’t double book a room.

Some problems we faced throughout the semester include having to learn many of the web programming components and intricacies on our own as well as passing variables from JavaScript to PHP. Since we made the clickable calendar found in the “Make A Reservation” page in JavaScript, we were able to have it return a date, however, we were unable to pass that date to the PHP method that was collecting the information to send to the database. We would like to implement the clickable calendar sometime in the future since it does have a clean design that most users would be able to interact with easily. We realize that the scope of the project was too big for one semester and will have to be completed later one; it is unfortunate that previous semesters work was scrapped due to inconsistencies and errors in the code. We believe that we have successfully laid the foundation for future semesters of Computer Science students to finish the project and deliver a fully functional website scheduler for One EPIC Place.
8. References


W3Schools, www.w3schools.com/php/.
9. Appendixes

Appendix A:
The difference between GET and POST requests

Although both GET and POST requests can accomplish the same tasks, they are fundamentally different. GET requests are saved in the browser’s history because they are part of the URL, POST requests are not saved in the browser’s history at all. We chose to use POST methods because they are generally more secure than GET methods and have no restrictions on form data length, unlike GET methods which are often limited to 2048 characters (GET vs POST).