**SEIS752 Advanced Web Design**

**Spring 2015**

**Project Report**

**‘Vacation Planner’**

**Mike Palmer & John Stark**

**Table of Contents**

# Introduction

# Description

## Purpose

## List of web pages

## Business Logic

# Technologies employed

## Languages (Html, javascript, php, JSON…).

## External API’s (Google maps)

## Database (MySQL, tables created)

## IDE

## Site Host (dreamer42 … , XAMPP – LocalHost, )

## Audio and Video

# Scope

## In-scope

## Out-of-scope

## If We Had More Time We Would …

# Software Development Process

## Agile

## Start small, code a little- test a lot

# Technical challenges encountered

1. **Introduction**

This project provided an opportunity to apply the lessons learned during the course and to develop an application to a greater extent that was possible in the weekly homework assignments. The application addresses the need to store, organize and access important information related to a vacation. Saving reservations, confirmations, phone numbers, etc. in this secure website makes them accessible by several means. They cannot left behind, blow out the car window or be eaten by the dog as a paper list might. Items needing further attention can be highlighted to provide a visual reminder.

This report assumes the reader has visited the website to gain some familiarity with it. <http://www.dreamer42.com/vacation/index.php>

1. **Description**

## purpose

This project developed and deployed a website for recording vacation plans. The user begins by registering or logging in, creating a vacation (which establishes a unique ID for it in the database) and then entering their daily itinerary. While planning the vacation the user can enter, change or delete information such as reservations, confirmation numbers, phone numbers and email addresses for lodging, sites to visit, daily destinations, etc.

The user can then reference and update their itinerary during their travels. In addition, the website’s business logic can provide some analysis of the travel plans. A google map is conveniently accessed by the click of a button with no need to re-enter location information.

## List of webpages.

The user interacts with some 9 webpages.

* [http://www.dreamer42.com/vacation/**index.php**](http://www.dreamer42.com/vacation/index.php)
  + Login via a drop down box then 🡪
  + Register [http://www.dreamer42.com/vacation/**register.php**](http://www.dreamer42.com/vacation/register.php)
* [http://www.dreamer42.com/vacation/**welcome.php**](http://www.dreamer42.com/vacation/welcome.php) Chose to create a new vacation or work with an existing one
* [http://www.dreamer42.com/vacation/**createNewVacation.php**](http://www.dreamer42.com/vacation/createNewVacation.php)
  + Enter vacation name and start date 🡪 enterDayDetails
  + [http://www.dreamer42.com/vacation/**enterDayDetails.php**](http://www.dreamer42.com/vacation/enterDayDetails.php)
* [http://www.dreamer42.com/vacation/**listOfVacations.php**](http://www.dreamer42.com/vacation/listOfVacations.php) choose a vacation to work with
* [http://www.dreamer42.com/vacation/**vacationSummary.php**](http://www.dreamer42.com/vacation/vacationSummary.php) edit, delete or add a row
  + Edit via <http://www.dreamer42.com/vacation/enterDayDetails.php> )
  + View Map [http://www.dreamer42.com/vacation/**displayMap.php**](http://www.dreamer42.com/vacation/displayMap.php)
  + Analyze travel plans via [http://www.dreamer42.com/vacation/**displayAnalysis.php**](http://www.dreamer42.com/vacation/displayAnalysis.php)

## Business logic.

The SQL queries provide some business logic by selecting only the appropriate information. The displayAnalysis.php page is backed by logic in \_\_\_\_\_\_\_\_\_\_\_\_ pages that compute and analyze the daily travel distances based on the starting locations and destinations entered. The results of the analysis is provide to the user textually and visually. The distances traveled determines which .jpeg image is loaded (hammock on the beach, of packed car on the road).

Although it’s not business logic, a random number is generated when the user clicks on the car image on the Index.php page. That random number then indexes into an array of audio clips that play.

# Technologies employed

## Languages (Html, javascript, php, JSON…).

Type some text

Google Map API (see below)

Bootstrap

Has built in resizing, formatting, colored buttons, …

Bootbox

Modal message boxes built on bootstrap and jQuery

Query parameters ….

better security and special characters

Login using encryption and SALT for storing password.

## External API’s (Google maps)

We used the Google Map API in our website. Our integration provides the user with a map showing their travel route, and computes their travel distances. The map informs the user of their route and may highlight any gross errors in the entered starting points and destinations.

## Database (MySQL, tables created)

We used MySQL which is part of the XAMPP stack, and is available on the remote host. The MySQL interface

## IDE

We used NetBeans and IntelliJ. NetBeans did not catch as many JavaScript errors as Chrome’s Inspector did. For local website hosting we used XAMPP.

## Site Host (dreamer42 … , XAMPP – LocalHost, )

We used FileZilla to upload our files to the remote host.

# Scope

## In-scope

Type some text

## Out-of-scope

There are many things that would still need to be done for a commercial website

Out of Scope items included

* + Security audit / refactoring
  + Refactor to add accessibility
  + Review fonts, CSS, for consistency page-to-page
  + More UI testing
  + More performance and scale testing

## If We Had More Time We Would …

Do more Security testing including use of an automated tool like Vercaode. More load and scale testing. More CSS work to improve the appearance. More mobile friendly design and then some mobile testing and rating

# Software Development Process

## Agile

Agile

* + Weekly iterations
  + Daily story / task assignments
* realistic scope

## Start small, code a little- test a lot

Type some text Start small, make it work, expand on it

## reposity

GIT worked great

# Technical challenges encountered

Neither of us had much experience developing websites, so finding the technologies and then learning how to use them was our first technical challenge. We applied the PHP, Javascript, etc. we learned in class. Google searches for solutions led us to Bootstrap, Bootbox, parameterized queries, etc. which were new to us and required some research to learn how to use them.

Creating an MLink allowed us to point GitHub and our IDE to the same set of local files. It took a little research and a couple tries to get things set up properly.

NetBeans IDE did not always show errors, Chrome Inspector was better at this

Bootstrap versions were not forward/backward compatible.

Javascript is very picky and not informative about errors

**APPENDIX A.**

**SEIS 752 – Final Project Proposal**

**Team Members:**

* Mike Palmer
* John Stark

**Project Description:**

The website resulting from this project will provide a traveler with a way to organize their travel plans. Users will be able to create vacation itineraries. This will help them during the planning process as they assemble the pieces. It will also be a helpful tool on the vacation to reference reservation numbers and chart the progress of their trip.

**Key Features:**

Create User Account

User Login

Create New Vacation

Display Vacation

Update Existing Vacation

Delete Vacation

Store Core Information about Each Day

**Bonus Features:**

Store Addition Advanced Information

Calculate Mileage Traveled Each Day

Provide a Map Showing Route

Provide Reports about Trips

Color Code Vacation to Alert to Areas of Concern

Security Measures Beyond Basic Login

Polished UI

**User Stories:**

#1 - planning

User wants to plan a vacation. User enters each day into the system. Fills in starting city, ending city, main activity for the day, and lodging info.

#2 – vacationing

User references the app each day. Retrieves their hotel reservation info. Reviews the activities for the upcoming day.

**Mock-up of Possible Screen**



**Challenges**

Using an external API to calculate travel time between two cites by driving.

Ensuring data is validated.

Preventing security breaches.

**Project Timeline**

|  |  |  |
| --- | --- | --- |
| Event | Start Date | End Date |
| Prepare Proposal | 2/18/2015 | 3/4/2015 |
| Gather Additional Requirements | 2/25/2015 | 3/11/2015 |
| Database Table Created | 3/11/2015 | 3/18/2015 |
| Login Screen | 3/11/2015 | 3/18/2015 |
| Display Single Vacation Day | 3/11/2015 | 3/25/2015 |
| Display Multiple Vacation Days | 3/11/2015 | 4/1/2015 |
| Entry of Single Vacation Day | 4/1/2015 | 4/8/2015 |
| Entry of Multiple Vacation Days | 4/1/2015 | 4/15/2015 |
| Add, Update, Delete | 4/8/2015 | 4/22/2015 |
| Bonus Features | 4/15/2015 | 5/6/2015 |
| Testing & Fixing | 4/15/2015 | 5/12/2015 |
| Launch! |  | 5/13/2015 |

**Tentative Technology Plan**

LAMP stack

GIT

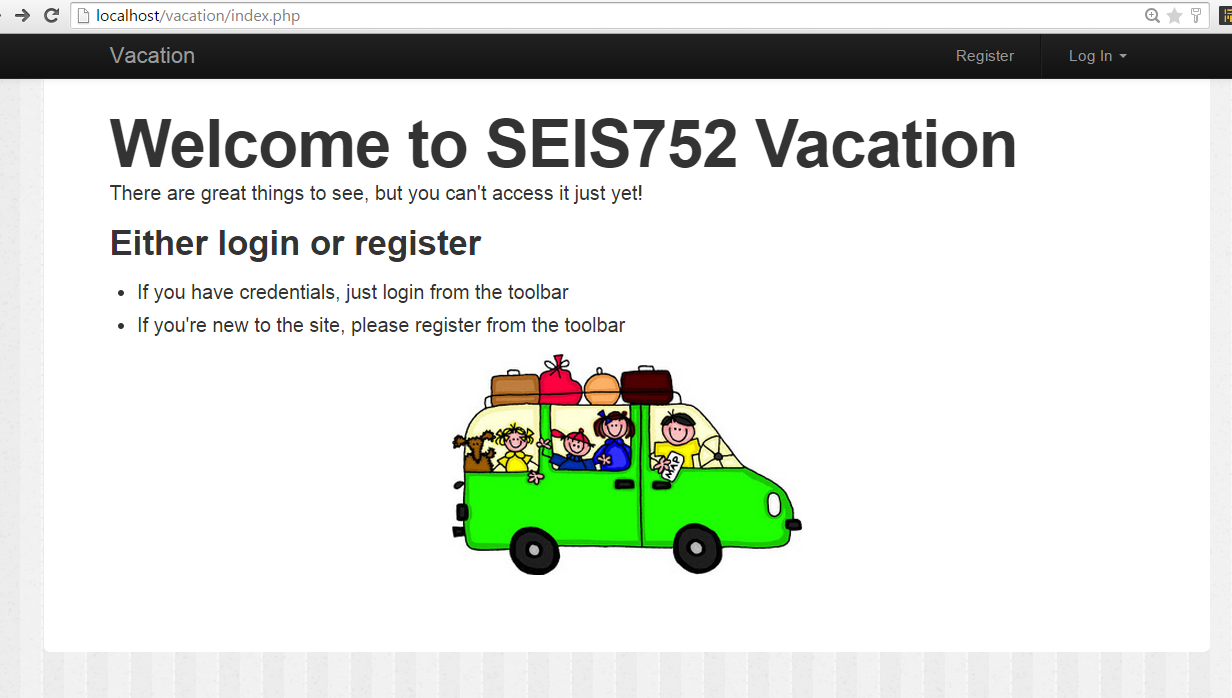
MySQL

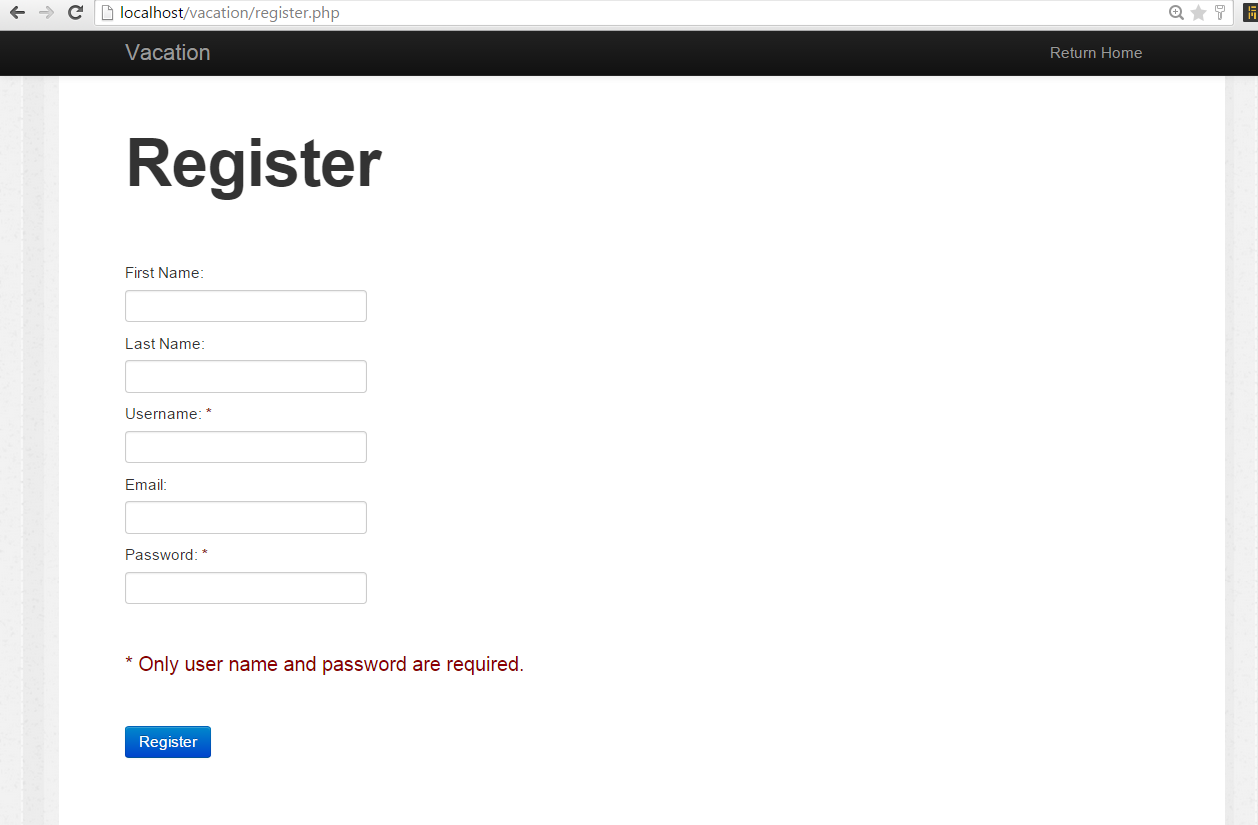
HTML

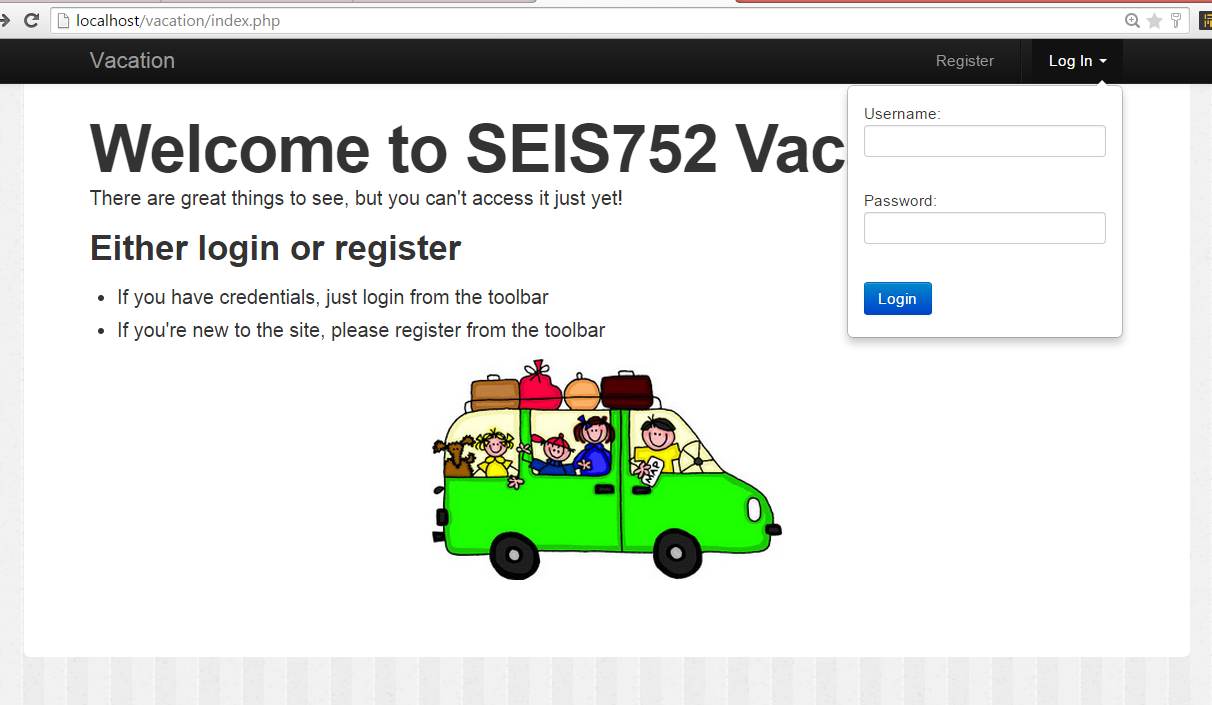
PHP

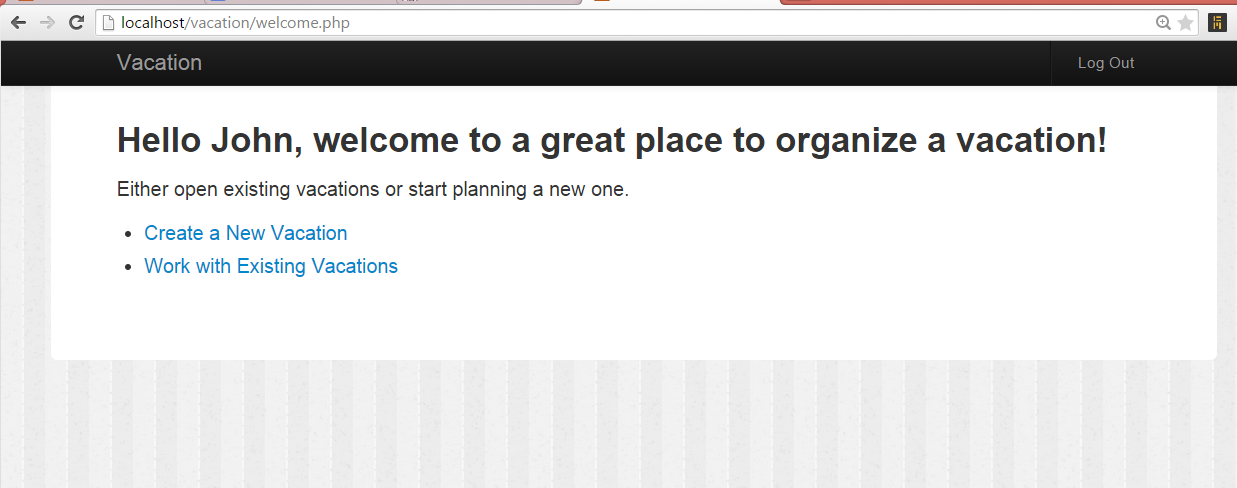
Javascript

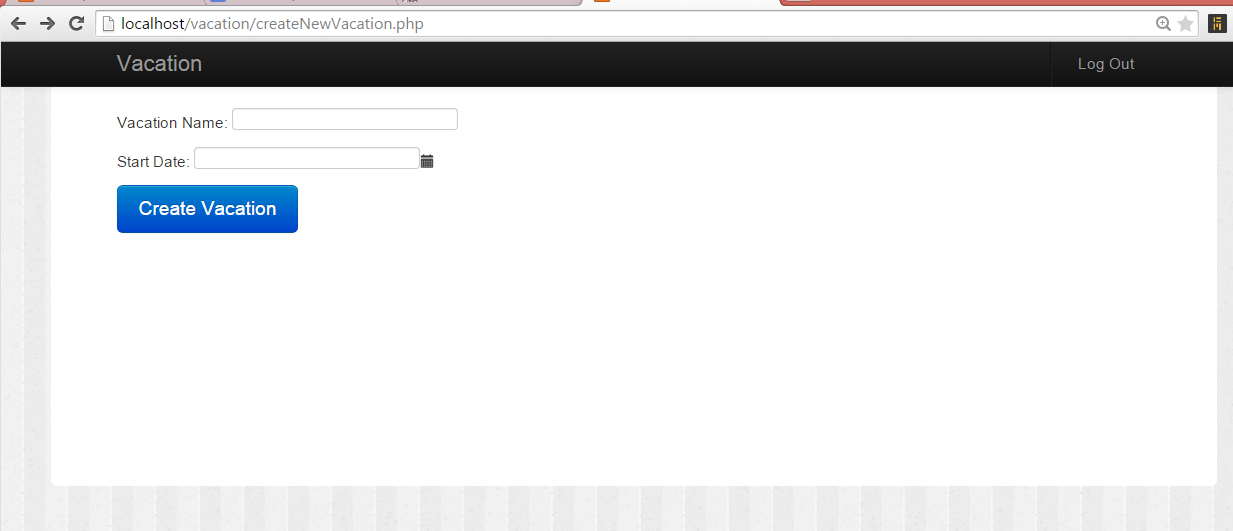
**APPENDIX B. Screen shots of pages**

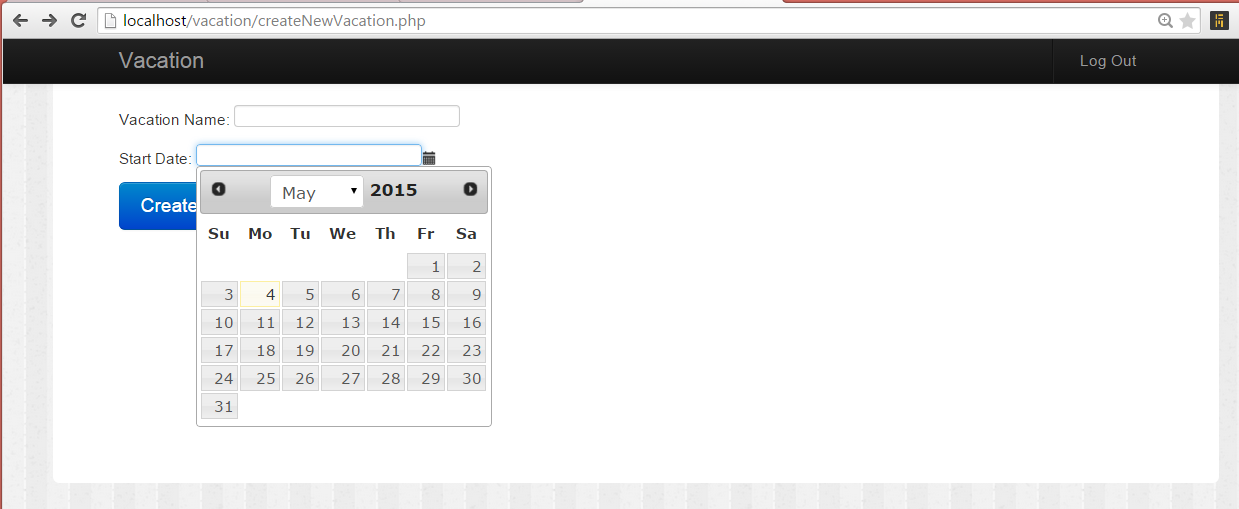


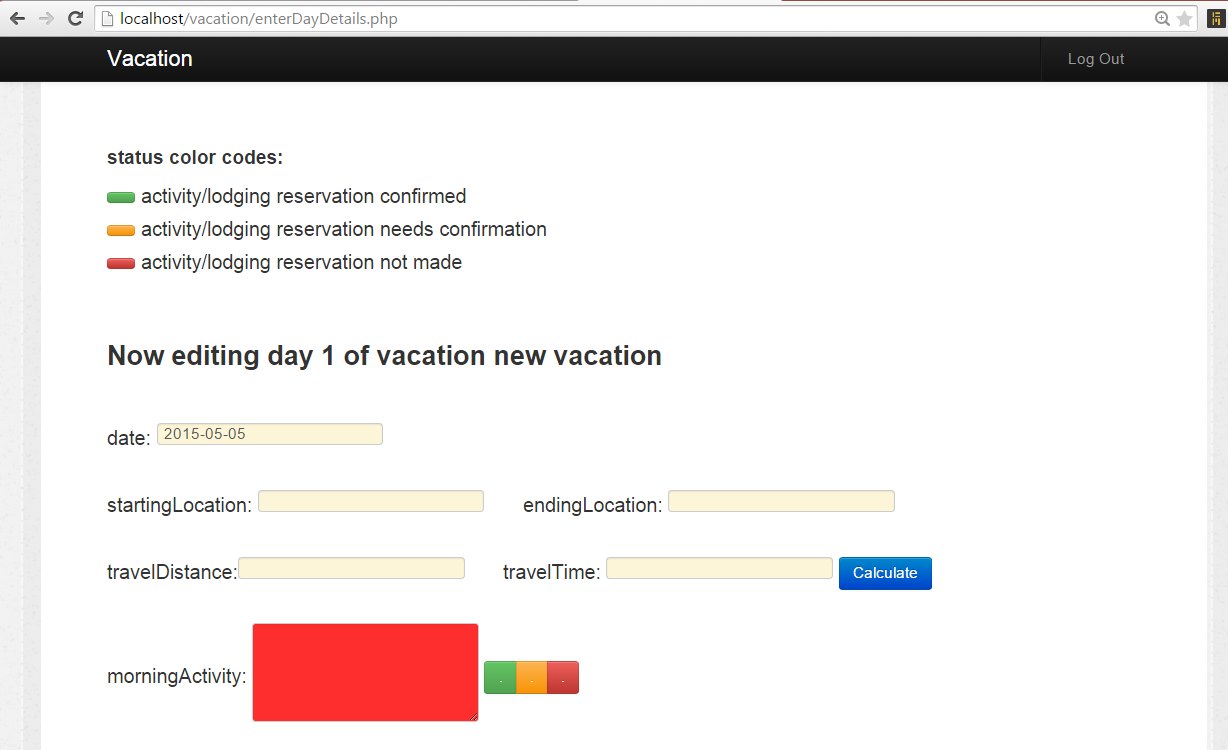


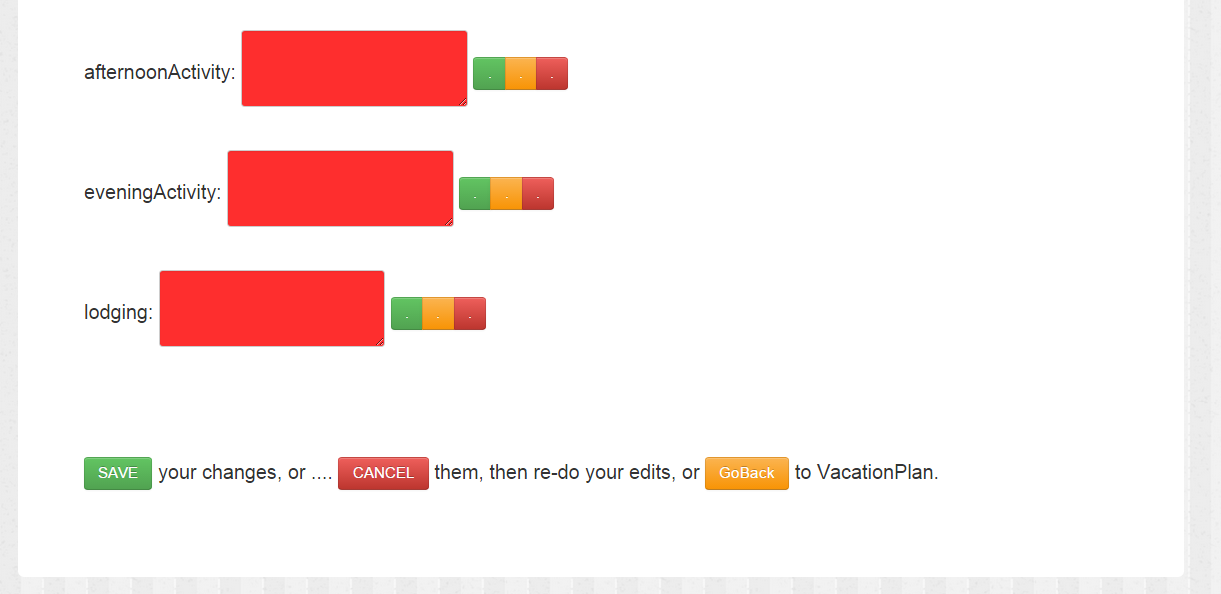


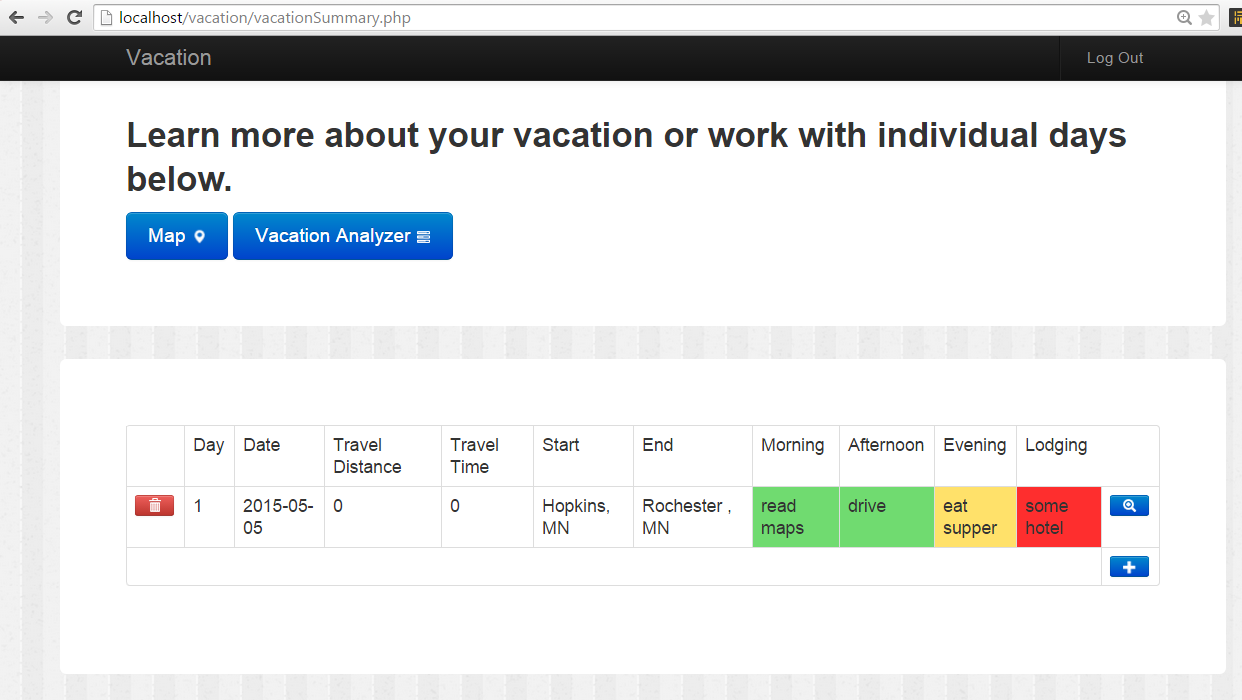


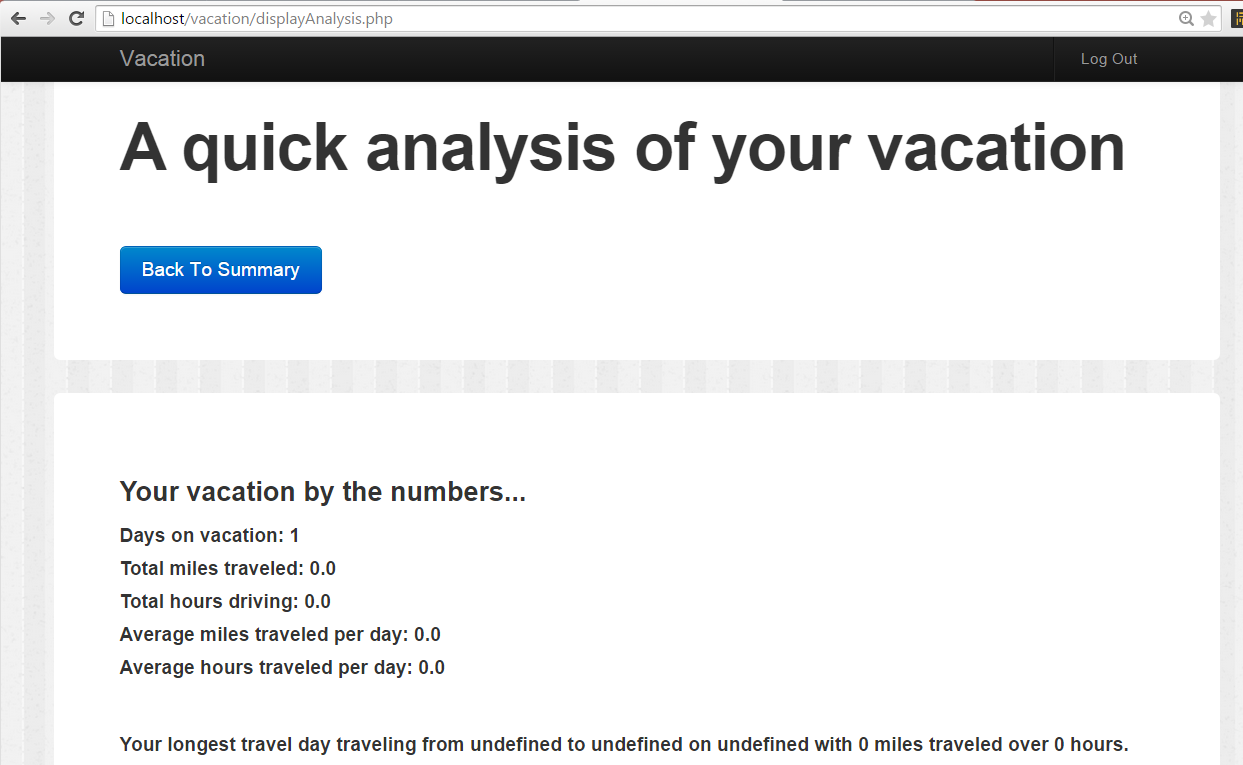




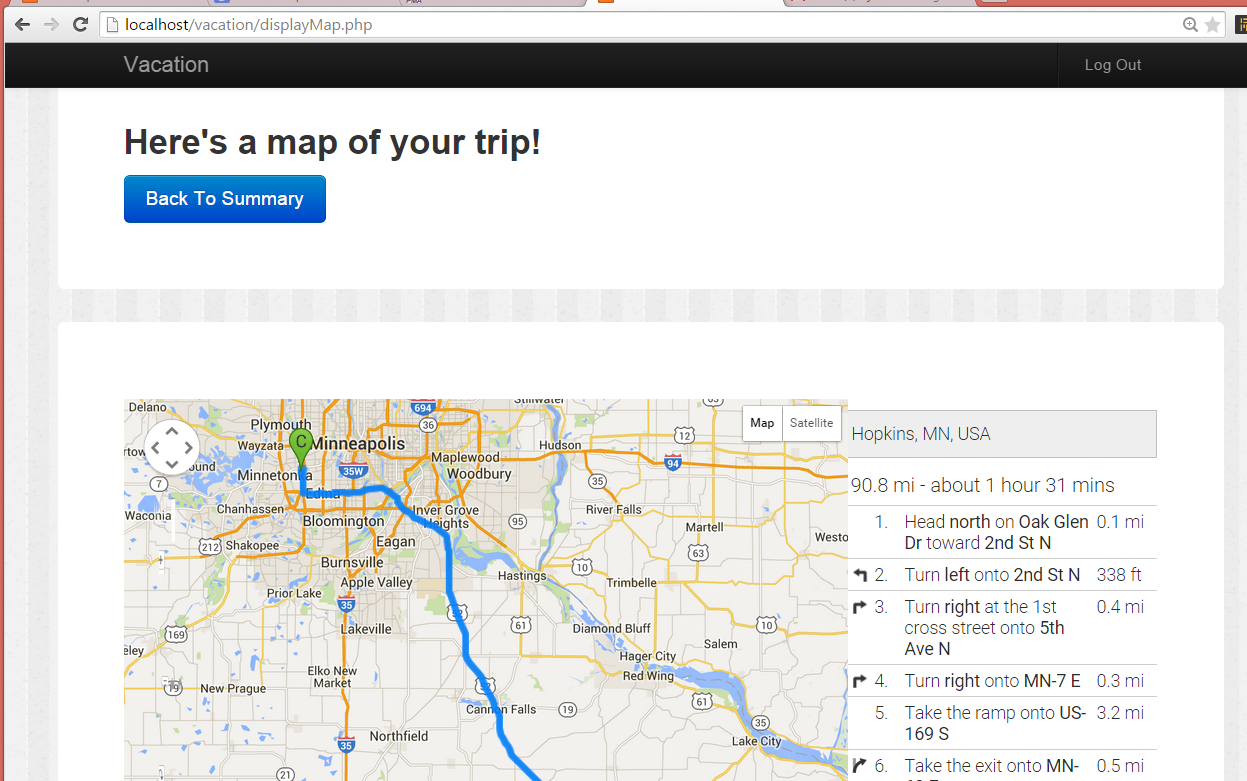


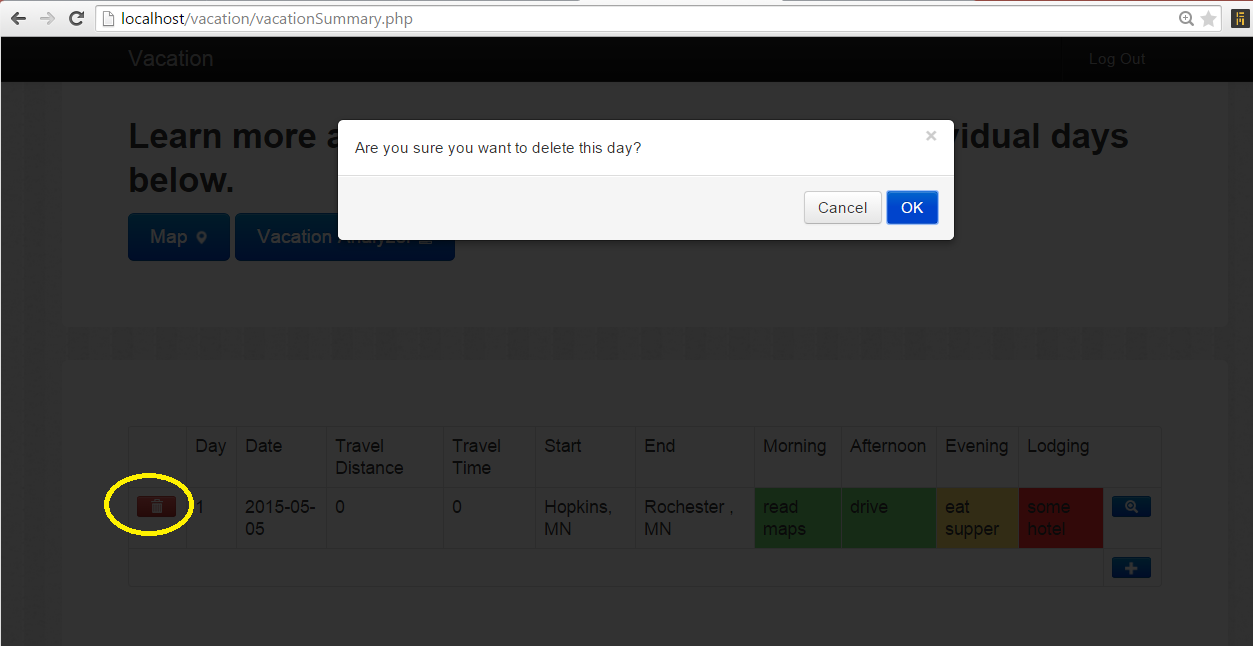












NOTES PASTE THINGS BELOW HERE THAT WE NEED TO INCLUDE IN THE REPORT

THIS IS JUST A SCRATCH PAD FOR OUR NOTES

Should technologies employed look like

This list shows up to 3 instances of where used

* Php
  + fetchVacationRow.php
  + insertVacation.php
  + deleteVacationPlan.php
* javascript
  + index.php
  + vacationSummary.php
  + setDayDetails.php