CS 148 Database Design for the Web

Joe Siebert and Alan Chu

Final Project

Stuff You Should Do At UVM

Version <1.0>

Table Of Contents

Time Log 3

Introduction 4

Purpose 4

Definitions, acronyms, and abbreviations 4

Overall Description 4

Story Board 5

Assignment requirements 6

Specific requirements 7

|  |  |  |  |
| --- | --- | --- | --- |
| Time Log | | | |
| Estimated time to complete assignment => | | | Hours |
| Date | Time Spent  (in hours) | Description | Author |
| 11/3/15 | 1.0 | First version of software requirements | Joe Siebert |
| 11/4/15 | 0.25 | Sketched preliminary storyboard | Joe Siebert |
| 11/5/15 | 1.5 | Created data dictionary; created tables using PHP myAdmin, typed in sample records, and tested functionality of table organization; created .sql files needed to recreate tables | Joe Siebert |
| 11/6/15 | 1.0 | Sized out elements for fluid CSS grid | Joe Siebert |
| 11/8/15 | 0.25 | Adding additional fields to tblActivities | Joe Siebert |
| 11/8/15 | 0.5 | Inserting records in tblActivities, tblTowns, and tblVotes; adjusted top-10 SQL query to grab only the top ten records | Joe Siebert |
| 11/9/15 | 0.5 | Researching logic necessary to update variables for up-/downvoting system | Joe Siebert |
| 11/9/15 | 0.5 | Created form.php; created read-only NetID field on form that gets the username for the UVM authentication system | Joe Siebert |
| 11/9/15 | 0.75 | Added two fields to form.php: activity name and category. | Joe Siebert |
| 11/10/15 | 1.0 | Added several fields to form.php; added an insert statement that inserts data into required fields of tblActivities | Joe Siebert |
| 11/11/15 | 2.0 | Added security feature to check whether users are administrators; began admin.php, adding the first feature: an interface for admins to approve submitted records | Joe Siebert |
| 11/11/15 | 1.25 | Added basic voting system to top-10.php | Joe Siebert |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | Mark me wrong if not updated | <=Total Time Spent |  |
|  |  |  |  |

# Introduction

### Purpose

The purpose of this document is to describe the requirement specification for the website “Stuff You Should Do At UVM”.

The intended audience for this document is our instructor and the TAs.

*First it is the contract between you and the client. Second it is for the developer of the site. Two primary goals are:*

1. *Client gets a very good understanding of what you will be delivering.*
2. *Developer has a very good understanding of what they are going to create.*

### Definitions, acronyms, and abbreviations

*Sample for the client:*

HTML – Hypertext markup language – used to define your content.

PHP – Personal Home Page – language that helps to customize html.

CSS – Cascading Style Sheets – used to define the look of a web site.

W3 Validation – refers to both Html and CSS validation tool provided by the W3c.org. the html validator is located at:

<http://validator.w3.org/>

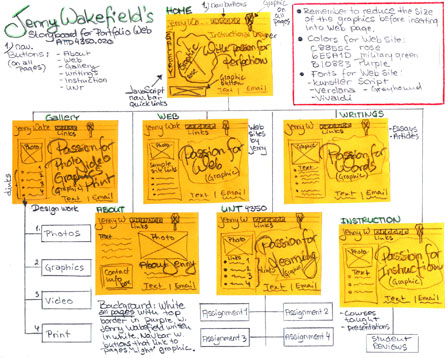
with the CSS validator located at:

<http://jigsaw.w3.org/css-validator/>

# Overall Description

Our site uses a database to track UVM students’ favorite activities. It displays the top activities on one page, and those outside the top ten on another page. Activities on either page can be up- or downvoted. Users can also submit new activities to the database.

# Story Board



# Specific requirements

General

1. *Meta description for all pages*: *"This site displays present and past student's favorite activities from when they were students at the University of Vermont. It's meant as a loose road map for current students and a source of nostalgia for alumni."*
2. *Meta charset for all pages*: UTF-8
3. *Pages will be created on a five-column fluid grid. The size of the various elements will be determined according to this grid.*
4. *Header of all pages includes a logo:*
5. *Navigation menu including links to all pages, except the page that is currently being viewed.*
6. *Footer will appear at the bottom of every page, and it will include the name of the site developers, and links to the About page.*
7. *Responsive page design that will reorganize the page content when the browser window drops below certain sizes. This effect will be achieved using CSS rules and the media query.*
8. *Site is maintained on GitHub, with various stages of edits being tracked here: https://github.com/bigSiebs/assignment10*

index.php

1. *Title:*
2. *A large banner-like element comprised of several small images that correspond to activities in the list. The smaller images will be overlaid with the name of the activity and its current rank.*
   1. *Images:*
3. *Summaries of the three most recently submitted activities.*
4. *A link in the page’s content to the top-ten list.*

top-10.php

1. *Title:*
2. *A table of the top-ten, most popular activities that have been submitted.*
3. *An up-arrow and a down-arrow, both included in each table row, which allow the user to upvote or downvote the activities listed. Pressing these buttons should communicate with the SQL database and update the Votes table.*
   1. *Users should not be able to vote something up or down more than once. However, if they vote something up once, then later down-vote it, they will be able to vote twice: once to set their score to zero, and another time to set the score of their vote to -1.*
   2. *The first version of this system should reload the page every time the user votes (i.e., be implemented using exclusively PHP). If there’s enough time, and there exists a more elegant solution that could be implemented using JavaScript or Ajax, a system using these techniques will replace the PHP-only system.*
4. *An element that displays additional information for activities that users click on. This element will be populated using JavaScript, and it will update each time the user clicks on a different activity.*

not-10.php

1. *Title:*
2. *A table displaying the remaining activities, limited to ten at a time. The user will be able to click buttons to see the next or previous ten.*
3. *The same voting system will be used on this page.*
4. *The same JavaScript-implemented element will be present on this page.*

form.php

1. *Title:*
2. *This page will contain a form that will:*
   1. *insert data into tblActivities, tblTowns, and tblAffiliates*
   2. *include a textbox to collect the user’s email address (if left blank, their netID will be used to created a default address)*
   3. *include several additional textboxes to collect information, including:*
      1. *activity name*
      2. *the town name*
      3. *the cost of the activity*
      4. *a URL associated with the activity*
      5. *a more detailed location description*
   4. *include a list box to collect the activity’s category.*
   5. *include a submit button*
   6. *validate data and display appropriate error messages*
   7. *send an email to the person who filled out the form*

You need a minimum one form

* include three check boxes.
* include three radio buttons.

about.php

admin.php