|  |  |
| --- | --- |
| CS 496 Assignment 2: Dynamic Pages | Kelvin Watson  ONID: watsokel  Student ID: 932540242 |

**Dynamic Pages Documentation**

URL to live web application: <http://malenah-portal.appspot.com/>

**Description of the web application/site**

The web application is called the M.A.L.E.N.A.H. portal. MALENAH stands for **M**obile **A**lly for **L**GBT people, **E**nhancing, through a **N**etwork, **A**ccess to **H**ealthcare. LGBT people face barriers to healthcare access and have a host of unique health-related needs. As such, the web application/site http://malenah-portal.appspot.com serves as an administrator portal, storing information about caring and culturally competent healthcare providers who are equipped to care for LGBT people in the hopes of closing the LGBT health disparities gap.

The MALENAH portal stores information about healthcare providers. Healthcare providers (class Provider) are stored as NDB models/entities in the NDB datastore, including physicians, dentists, pharmacists, nurses, counsellors, and others. Providers have the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute of the Provider class/NDB Model | NDB Property type [1] | Description | Form input |
| first\_name | ndb.StringProperty() | healthcare provider’s first name | <input type=”text”> |
| last\_name | ndb.StringProperty() |  | <input type=”text”> |
| phone | ndb.StringProperty() |  | <input type=”tel”> |
| email | ndb.StringProperty() |  | <input type=”email”> |
| website | ndb.StringProperty() |  | <input type=”url”> |
| best\_time | ndb.TimeProperty() | best time to contact the healthcare provider | <input type=”time”> |
| designation | ndb.StringProperty() | the healthcare provider’s designation/degree (e.g. MD, RN, Pharm.D) |  |
| service(s) |  | service(s) offered by the healthcare provider | <input type=”checkbox”> |
| accept\_new\_patients | ndb.BooleanProperty() | accepting new patients or not | <input type=”radio”> |

* first\_name
* last\_names
* phone
* emails
* website represented in the form<input type=”url”> designation and services provided). Healthcare providers are stored as NDB models (entities) in the NDB datastore, including physicians, dentists, pharmacists, nurses, counsellors, and others. The datastore also stores two additional NDB models as entities: healthcare services and healthcare professional designations, both of which have a name as a StringProperty() and are associated with healthcare providers.

**Test Plan**

To ensure that the web application is working properly, I used this test plain, containing the following manual tests.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Type** | **Test Case** | **Input** | **Expected Outcome** | **Observed Outcome** |
| 1 | Blank | Empty database. Attempt to submit the “Add Provider” form with empty fields in Google Chrome, Firefox. | Click on the submit button in the “Add Healthcare Provider” form. | HTML5 form should prompt the user to complete the required fields before form can be submitted. |  |
| 2 | Edge case | Empty database. Attempt to submit the “Add Provider” form with empty fields in Safari, where support for the “required” tag is only partial [1]. | Click on the submit button in the “Add Healthcare Provider” form. | HTML5 form should prompt the user to complete the required fields before form can be submitted. | Form Submission is Unsuccessful for the following reason(s):  You did not enter a first name.  You did not enter a last name.  You did not enter a phone number.  You did not enter an email.  You did not enter a website.  You did not enter a contact time.  You did not enter whether or not new patients are being accepted. |
| 3 | Blank | Submit empty designation | Click on the submit button in the “Add a Designation” form. | HTML5 form should prompt user to complete required field before submitting. | Form submission is unsuccessful for the following reason(s):  You did not enter a designation to add. |
| 4 | Blank | Submit empty service | Click on the submit button in the “Add a Service” form | HTML5 form should prompt user to complete required field before submitting. | Form submission is unsuccessful for the following reason(s):  You did not enter a service to add. |
| 5 | Edge | Empty database, create first (non-empty) service | ‘Counselling’ | Counselling service added successfully (as shown in the /view page) and appears as a radio button choice in the “Add Provider” form on form submission.  No records in the Provider or Designation tables |  |
| 6 | Edge | Empty database, create first designation | Pharm.D | Pharm.D designation service added successfully (as indicated in the /view page) and appears as a checkbox in the “Add Provider” form.  No records in the Provider or Designation tables |  |
| 7 | Edge  Blank | Empty database, create first provider without first creating a service or a designation | first name: Kelvin  last name: Watson  phone: 1234567890  email: [me@me.com](mailto:me@me.com)  website: <http://abc.com>  designation: ‘’  services: ‘’  best time to contact: 11:59pm  accepting patients: No | Provider successfully added as evidenced by the /view page  Provider appears in Provider table on form submission, with empty cells under the Designation and Services columns. |  |
| 8 | Edge  Blank | Empty database, create first provider without first creating a designation but having created a service | first name: Kelvin  last name: Watson  phone: 1234567890  email: [me@me.com](mailto:me@me.com)  website: <http://abc.com>  designation: Pharm.D  services: ‘’  best time to contact: 11:59pm  accepting patients: No | Provider successfully added as evidenced by the /view page  Provider appears in Provider table on form submission, with empty cell under the Services columns, but the “Pharm.D” designation under the Designation column.  Pharm.D should also be in the Designations table beneath the Provider table in the view and admin pages. |  |
| 9 | Edge  Blank | Empty database, create first provider without first creating a service but having created a designation |  |  |  |
| 10 | Edge | Delete all services, even though they have been associated with providers | Delete all services using the Datastore Admin and refresh the /admin page |  | Under “All Records Currently in Database,” Designation table is empty, but provider tables still contain designations. This is the desired. |
| 11 | Edge | Navigate to the /view page directly. |  | Redirects to /admin | Redirects to /admin |
| 12 | Edge | Navigate to the /edit page directly |  |  | Redirects to /admin |
| 13 | Edge | Navigate to an invalid url e.g. malenah-portal.appspot.com/a |  |  | Redirects to /admin |
| 14 | Edge | Navigate to the /admin page directly. |  |  | Correctly renders administration page with forms |
| 15 | Basic | Duplicate designation | Add designation “MD” twice | Reject the addition of designations with the same name as a designation entity already in the database. | Unable to add record for the following reason(s):  The designation you are trying to add already exists in the database. |
| 16 | Basic | Duplicate service | Add service “Counseling” twice | Reject the addition of services with the same name as a service entity already in the database. | Unable to add record for the following reason(s):  The service you are trying to add already exists in the database. |
| 17 | Basic | Enter invalid URL for website |  |  |  |
| 18 | Basic | Enter invalid phone number | ‘abc-def-ghij’ | Reject invalid phone numbers |  |
| 19 | Blank  Basic | Enter invalid email | ‘example123@@.com’ | Reject invalid emails |  |
| 20 | Blank  Basic | Submit “Add a Provider” form with missing first name in Chrome and Safari. (“required” attribute is not supported in Safari) |  |  |  |
| 21 | Blank  Basic | Submit “Add a Provider” form with missing last name. (“required” attribute is not supported in Safari) |  |  |  |
| 22 | Blank  Basic | Submit “Add a Provider” form with missing phone. (“required” attribute is not supported in Safari) |  |  |  |
| 23 | Blank  Basic | Submit “Add a Provider” form with missing email. (“required” attribute is not supported in Safari) |  |  |  |
| 24 | Blank  Basic | Submit “Add a Provider” form with missing designation. (“required” attribute is not supported in Safari) |  |  |  |
| 25 | Blank  Basic | Submit “Add a Provider” form without checking off services(s). (“required” attribute is not supported in Safari) |  |  |  |

**Results of Executing Test Plan**

The process of documenting the test plan initially revealed a number of bugs, which have now been fixed. These bugs were mostly due to inadequate error checking, and including the following:

|  |  |
| --- | --- |
| **Bugs** | **Status** |
| * duplicate entries of designations and services did not get caught | Fixed |
| Routing bug - navigating directly to \_\_ does not automatically redirect to the /admi page |  |
| Routing bug – handlers placed in the wrong order resulting in the admin page always being rendered instead of the edit page. | FIXED  The default route ‘.\*’ should be the listed last as the application tries to match the routes in order. [5] |
| Missing input validation for empty form input fields. Safari allows forms to be posted/submitted even when input fields containing the “required” attribute are left empty |  |
| * Safari |  |
| * missing return statement in post and get methods when form input invalid |  |
| * duplicate entries of designations and services did not get caught |  |

* duplicate entries of designations and services did not get caught
* routing issues - navigating directly to \_\_ does not automatically redirect to the /admi page
* Safari
* no return statement when input validation for

., and executing the test plan,

**Explanation of Fixes**

**Explanation of Failures**

**Discussion of Templating to Display Data**

Specifically, using Jinja templating to implement logic, render variables, access properties. For logic, I used the following control structures: the *for* loop, the *if/elif/else* statements. I also used Jinja for comparison of strings for equality, for example, {% if s.name == t.name %} compares two strings.

By using the *extends* tag, I was able to extend an html template called index.html to all of my other page templates. For instance, in my admin.html template, I used the statement {% include index.html' %} to prevent the need to duplicate html tags and jinja expressions common among the pages, such as <html>, <head>, <title> <body> or {{title}}.

Using the *extends* tag also allowed me to override blocks within the index.html template as necessary. For instance, index.html contains a content *block* {% block content %} as well as an “all database data” block {% block all\_database\_data %}. Only the admin and view pages override the all\_database\_data block. All pages override the content block.

Finally, I used filters to pipe expressions into Jinja’s built-in filters, including *length()* to get the number of items in a list, and *string()* to convert a string to Unicode.

**Changes I would make if I had to do it over again**

I would design my application more thoroughly prior to coding to reduce the amount of code duplication. For instance, the get() and post() methods in the view.py ViewHandler contains many lines of code in common. This indicates that there is could be an appropriate design pattern that could be used to

Thinking ahead to the mobile application portion of the course, if I had to do this assignment over again, I would have made the code more adaptable/flexible to creating a sort of backend API to respond to requests from a mobile application.

**Sources**

[1] <https://cloud.google.com/appengine/docs/python/ndb/db_to_ndb>

[2] <http://caniuse.com/#feat=form-validation>

[3] <http://jinja.pocoo.org/docs/dev/>

[4] <http://jinja.pocoo.org/docs/dev/templates/>

[5] <https://webapp-improved.appspot.com/guide/routing.html>