|  |  |
| --- | --- |
| 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, called the M.A.L.E.N.A.H. Portal, is constructed on the Google App Engine cloud service/platform in Python.

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() | Healthcare provider’s last name | <input type=”text”> |
| phone | ndb.StringProperty() | Healthcare provider’s phone number | <input type=”tel”> |
| email | ndb.StringProperty() | Healthcare provider’s email address | <input type=”email”> |
| website | ndb.StringProperty() | Healthcare provider’s website url | <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) | <input type="radio"> |
| service(s) | ndb.StringProperty  (repeated=True) | Service(s) offered by the healthcare provider | <input type=”checkbox”> |
| accept\_new\_patients | ndb.BooleanProperty() | whether or not the healthcare provider is accepting new patients | <input type=”radio”> |

* 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute of the Designation class**  **(NDB Model)** | **NDB Property type [1]** | **Description** | **Form input** |
| name | ndb.StringProperty() | Name of the designation associated with a healthcare provider | <input type=”text”> |

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute of the Service class**  **(NDB Model)** | **NDB Property type [1]** | **Description** | **Form input** |
| name | ndb.StringProperty() | Name of the service provided a healthcare provider | <input type=”text”> |

**Test Plan**

To ensure that the web application is working properly, I used this test plan, containing 31 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. | As expected, the healthcare provider form does not allow submission without required fields completed. |
| 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. | The following error messages appear, as expected:  *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. | In the Google Chrome browser, the designation form does not allow submission due to the required attribute on the input tag. This is the correct outcome.  In the Safari browser, as expected, the form submits but the following error message is displayed to the user: *Unable to add record for the following reason(s): You did not enter a designation to add.* This is also the correct outcome for this browser. |
| 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. | In the Google Chrome browser, the designation form does not allow submission due to the required attribute on the input tag. This is the correct outcome.  In the Safari browser, as expected, the form submits but the following error message is displayed to the user*: Unable to add record for the following reason(s): You did not enter a service to add.* This is also the correct outcome for this browser. |
| 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 | Service is recorded as expected    Service now appears in the provider form |
| 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 | Designation recorded as expected    Designation appears in the provider form as expected. |
| 7 | Edge  Blank | Empty database, create first provider without first creating a service or a designation | first name: Kelvin  last name: Watson  phone: 9716786247  email: [abc@abc.com](mailto:abc@abc.com)  website: <http://abc.com>  best time to contact: 11:59pm  designation:  services:  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. | Record successfully added as expected. |
| 8 | Edge  Blank | Empty database, create first provider without first creating a service but having created a designation | first name: Kelvin  last name: Watson  phone: 9716786247  email: [abc@abc.com](mailto:abc@abc.com)  website: <http://abc.com>  best time to contact: 11:59pm  designation:  services: Counselling  accepting patients: Yes | Provider successfully added as evidenced by the /view page | Provider is successfully added and does not show any associated designations, as expected. |
| 9 | Edge  Blank | Empty database, create first provider without first creating a designation but having created a service | first name: Kelvin  last name: Watson  phone: 9716786247  email: [abc@abc.com](mailto:abc@abc.com)  website: <http://abc.com>  best time to contact: 10:59AM  designation: B.Sc  services:  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 “B.Sc” designation under the Designation column.  B.Sc should also be in the Designations table beneath the Provider table in the view and admin pages. | Record is added as expected. |
| 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 | “All Records Currently in Database” provider table should still render the previously added services. | Under “All Records Currently in Database,” Designation table is empty, but provider tables still contain designations. This is the desired outcome. |
| 11 | Edge | Navigate to the /view page directly. | malenah-portal.appspot.com/view | Redirects to /admin | Redirects to /admin as expected. |
| 12 | Edge | Navigate to the /edit page directly | malenah-portal.appspot.com/edit | Redirects to /admin | Redirects to /admin as expected. |
| 13 | Edge | Navigate to an invalid url e.g. malenah-portal.appspot.com/a! | malenah-portal.appspot.com/a! | Redirects to /admin page | Redirects to /admin as expected. |
| 14 | Edge | Navigate to the /admin page directly. | malenah-portal.appspot.com/admin | Correctly renders administration page with forms | 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 | Enter “abc.abc” in the website field of the provider form. | Should reject invalid URLs, such as those with incorrect or missing protocols and/or top-level domains | The website field expects either http:// or https:// in the field and invalid inputs such as “abc.abc” are correctly rejected.    However, as long as the protocol is provided, an invalid URL is still accepted. This is not a desired outcome. |
| 18 | Basic | Enter invalid phone number | ‘abc-def-ghij’ | Application should reject invalid phone numbers. | The application accepted an invalid phone number (a string containing non-numeric characters). This is not the desired result. |
| 19 | Blank  Basic | Enter invalid email | “helena@bolton” | Application should reject invalid emails | The application accepted an invalid email address missing a top-level domain. This is not the desired result. |
| 20 | Blank | Submit “Add a Provider” form with missing first name in Chrome and Safari. (“required” attribute is not supported in Safari) | Empty form field | Application should either reject the form submission (Chrome) or display an error message (Safari). | In the Google Chrome browser, the “required” attribute in the input tag correctly disallows form submission without first entering input into the form field.    In the Safari browser, the correct error message is displayed:  *Unable to add record for the following reason(s):*  *You did not enter a first name.* |
| 21 | Blank | Submit “Add a Provider” form with missing last name. (“required” attribute is not supported in Safari) | Empty form field | Application should either reject the form submission (Chrome) or display an error message (Safari). | In the Chrome browser, form submission is rejected due to the “required” attribute for the input tag, as expected:  In the Safari browser, the result is the following error message, as expected:  *Unable to add record for the following reason(s):*  *You did not enter a first name.* |
| 22 | Blank | Submit “Add a Provider” form with missing phone. (“required” attribute is not supported in Safari) | Empty form field | Application should either reject the form submission (Chrome) or display an error message (Safari). | In the Chrome browser, form submission is rejected due to the “required” attribute for the input tag, as expected:  In the Safari browser, the result is the following error message, as expected:  *Unable to add record for the following reason(s): You did not enter a phone number.* |
| 23 | Blank | Submit “Add a Provider” form with missing email. (“required” attribute is not supported in Safari) | Empty form field | Application should either reject the form submission (Chrome) or display an error message (Safari). | In the Chrome browser, form submission is rejected due to the “required” attribute for the input tag, as expected:  Unable to add record for the following reason(s): You did not enter an email. |
| 24 | Blank | Submit “Add a Provider” form with missing designation. | Submit provider form with empty designation field. | Application should allow submission | The form is submitted successfully, as expected, since designation is not a required form field: |
| 25 | Blank | Submit “Add a Provider” form without checking off services(s). | Submit provider form with empty service field. | Form should still be successfully submitted. | The form is submitted successfully, as expected, since service(s) is not a required form field: |
| 26 | Basic  Blank | Entering only space characters into the text input fields. | Submit only space characters into the text input fields. | Should not allow form submission. | Fields containing only whitespace characters are correctly rejected, and the user is prompted to provide non-whitespace input, as indicated by the following error: Unable to add record 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. |
| 27 | Basic | Illegal characters in any of the text fields (first and last names, phone, url, email etc) | Submit non-numeric characters to the phone number and/or website field(s). | Should not allow form submission, or display error message to user. | Form submission was allowed, but this is not the desired result. Notice the invalid characters in the email and website fields. |
| 28 | Basic | Edit provider with no associated services | Click on “Edit” next to a provider with no associated services. | Edit form should show no checked services. |  |
| 29 | Basic | Edit provider with no associated designation. | Click on “Edit” next to a provider with no associated designation. | Edit form should show no checked designation. | Because the provider had no associated designations prior to clicking “Edit,”, the edit form correctly shows the designation field with no prepopulated radio buttons.    Checking off Pharm.D as a designation, the form submits successfully and the record is correctly updated. |
|  | Basic | Edit provider’s designation | In the Edit form, select a different designation than the original designation. | Edit form should be successfully submitted and database record successfully updated. | In the Edit form, “RN” is pre-populated as expected.  On changing the designation from RN to RD, the /view page shows the record has been successfully updated. |
| 30 | Basic | Edit provider’s services (add an additional service) | In the Edit form, select an additional service. | Edit form should be successfully submitted and database record successfully updated. | In the edit form, Nutrition and Psychiatry is prepopulated, as expected.    Checking off an additional service adds another service to the provider, which is correctly submitted as expected. |
| 31 | Basic | Edit provider’s services (remove a service) | In the Edit form, unselect a service. | Edit form should be successfully submitted and database record successfully updated. | The edit form correctly prepopulates the three services associated with the provider.    Removing Diabetes Education correctly results in the service being removed. |
| 32 | Basic | Edit whether or not provider is accepting patients in a provider that did not accept patients | Click on “Edit” next to the provider. Edit the response to “Yes” under “Accepting Patients,” then click on the “Update Record” button. | If the provider’s did not originally accept patients, the “No” radio button should be checked off in the edit page. Checking off “Yes” should correctly edit the record. | In the edit form, the “No” next to Accepting Patients is correctly checked off.    Checking off the record to “Yes” under “Accepting Patients” correctly changes the record. |

**Results of Executing Test Plan**

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

|  |  |
| --- | --- |
| **Bugs** | **Explanation of Fixes and Failures** |
| Duplicate entries of designations and services did not get caught | FIXED |
| Routing bug - navigating directly to \_\_ does not automatically redirect to the /admi page | FIXED |
| 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 | FIXED |
| Missing return statement in post and get methods when form input invalid | FIXED |

**Explanation of Failures**

There are several outstanding bugs, which include the following:

1. The application currently does not parse URL’s, and accepts invalid URL’s with missing top-level domains and stores them in the database as strings.

* In order to fix this bug, I could use regular expressions. An easier way would need to import a library such as the validator library, urlparse from urlparse, or the URLValidator from django.core.validators in Python to parse URL’s. [6] [7]

2. The application currently does not parse phone numbers. Instead, it accepts invalid phone numbers and stores them in the database as strings..

* In order to fix this bug, I would need to import a library such as phonenumbers 7.2.3 in Python to parse phone numbers. [8]

3. The application currently does not parse email addresses, and accepts invalid email addresses and stores them in the database as strings.

* In order to fix this bug, I would need to import a library such as validate\_email 1.3 in Python to parse phone numbers. The Python validators library is also able to validate emails. [6] [9]

**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 incorporating an appropriate design pattern could have resulted in much cleaner code.

Some features that I missed were text validation, including stripping of leading and trailing white space. In addition, there should be stricter input validation of phone numbers <input type=”tel”> and emails <input type=”email”> records

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>

[6] <https://validators.readthedocs.org/en/latest/>

[7] <https://docs.python.org/2/library/urlparse.html>

[8] <https://pypi.python.org/pypi/phonenumbers>

[9] <https://pypi.python.org/pypi/validate_email>