**Fix Politics Test Plan**

**Version 1.0**

**08/31/2020**

**Document Approvers:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | E-mail | Title | Date Approved |
| Tony Pearson | tpearson@us.ibm.com | Developer / Architect |  |
| Tommy Adams | tnadams@us.ib.com | Tester / Author |  |
| Beth Morgan | Bethany.Morgan@ibm.com | UI Designer |  |
| Nikhil Raja | Nikhil.Raja@ibm.com | Project Manager |  |
| Mao Vang Corne | MaoVangCorne@ibm.com | Content Curator |  |

**Revision Control:**

|  |  |  |
| --- | --- | --- |
| Version Number | Date | Changes |
| 1.0 | 08/31/2020 | Initial draft of document |
|  |  |  |

Table of Contents

[Section 1 Preface 4](#_Toc49765428)

[1.1 Emb(race): Policy and Legislation Reform 4](#_Toc49765429)

[1.2 What's the problem? 4](#_Toc49765430)

[1.3 How can technology help? 5](#_Toc49765431)

[1.4 The idea - Fix Politics App 5](#_Toc49765432)

[1.5 Revision Control 5](#_Toc49765433)

[1.6 References 5](#_Toc49765434)

[Section 2 Scope 5](#_Toc49765435)

[Section 3 Test Case Template 6](#_Toc49765436)

[Section 4 Test Activities 6](#_Toc49765437)

[4.1 User Actions 6](#_Toc49765438)

[4.1.1 Register New User 7](#_Toc49765439)

[4.1.2 User Login 7](#_Toc49765440)

[4.1.3 User Logout 8](#_Toc49765441)

[4.1.4 Modify User Profile 8](#_Toc49765442)

[4.2 Searching the Database 9](#_Toc49765443)

[4.2.1 Search While Logged In 9](#_Toc49765444)

[4.2.2 Search While Logged Out 9](#_Toc49765445)

[4.3 Exporting Results 10](#_Toc49765446)

[4.3.1 Print Results 10](#_Toc49765447)

[4.3.2 Download Results as CSV 10](#_Toc49765448)

[4.3.3 E-mail Results 11](#_Toc49765449)

[4.4 Site Navigation 12](#_Toc49765450)

[4.4.1 Menu Options 12](#_Toc49765451)

[4.4.2 Browser Back/Forward Button 12](#_Toc49765452)

[4.5 Bad Path 13](#_Toc49765453)

[4.5.1 Invalid User ID 13](#_Toc49765454)

[4.5.2 Invalid Password 13](#_Toc49765455)

[4.5.3 Invalid Registration 14](#_Toc49765456)

# Section 1 Preface

### 1.1 Emb(race): Policy and Legislation Reform

Technology has the power to drive action. And right now, a call to action is needed to eradicate racism. **Black lives matter.**

We recognize technology alone cannot fix hundreds of years of racial injustice and inequality, but when we put it in the hands of the Black community and their supporters, technology can begin to bridge a gap. To start a dialogue. To identify areas where technology can help pave a road to progress.

This project is an effort to utilize technology to analyze, inform, and develop policy to reform the workplace, products, public safety, and legislation.

This is one of three open source projects underway as part of the [Call for Code Emb(race) Spot Challenge](https://github.com/topics/embrace-call-for-code) led by contributors from IBM and Red Hat.

### 1.2 What's the problem?

Concerned citizens and impacted residents don't have a straightforward way of knowing what or how policies and regulations impact them or what they can do in response. A community leader could use this tool to help motivate their social followers. Our target user will be referred to as "advocate".

Hills (who, what, and wow)

1. Advocates are aware of policy that is being considered that is highly impactful to them, without needing to follow every vote.
2. Advocates are able to understand the specific impact of proposed policy on them without being a legal expert.
3. Advocates are able to share opinions so they can influence policy decisions before they are finalized.
4. Advocates can easily ascertain the voting record themes or trend of their elected officials and political candidates without prior knowledge of who they are.
5. Policy makers have visibility into how diverse citizenry will be impacted by multiple variations of a proposed policy.

### 1.3 How can technology help?

Searching for legislation can be complicated by the fact that each government level has their own repository, their own numbering convention, and different formats. Technology can assist by allowing curated content to be centralized into a single database, readily accessible to advocates that need simple-to-read information.

### 1.4 The idea - Fix Politics App

Fix Politics is a web-based application written in Python programming language, using the Django framework and Bootstrap user interface styling. Its primary goal is to help advocates find legislation of interest based on an advocate's preferences for impact areas and geographical location.

The application is customizable, allowing application staff to specify the location hierarchy and impact categories. Complex legislation is curated, resulting in classifying the location scope and impact area, with a brief laymen-readable title and summary.

By classifying legislation by impact and location, we hope to increase awareness of current and pending legislation and their ability to affect change through voting or other activism.

### 1.5 Revision Control

Revisions to this document will be tracked after the initial release of Version 1.0. Deletions for each subsequent revision are identified by ~~strike through~~ and additions are identified by red font.

### 1.6 References

For reference, please see the following links:

Emb(race) Call for Code GitHub repository

* <https://github.com/topics/embrace-call-for-code>

Fix Politics GitHub repository

* <https://github.com/embrace-call-for-code/fix-politics>

# Section 2 Scope

This test plan documents the test activities as well as general use cases for the Fix Politics application. It assumes a running application, either on local host or in IBM Cloud, to perform these activities.

For now, the testing will be done manually. As more data and use cases are added, however, it is suggested to move to an automated web application test framework, such as Robot Framework and Selenium. This will allow the test activity to scale with the expanded functionality and data set used in future versions of the application. Along with this, Combinatorial Test Design (CTD) should be used as the data set increases to allow for full coverage of test input variations without the need to explicitly test every use case.

Testing outside of this plan includes:

* IBM Cloud
* Developer tools (Django, Pyenv, Gunicorn, etc.)
* Databases (SQLite, PostgreSQL)

# Section 3 Test Case Template

**Abstract**

Quick description of the test case

**Pre-requisites**

Any pre-requisites that must exist before running this test case

**Criteria**

Expected results in order to pass the test case

**Method**

Procedure for testing this function

**Variations**

Different inputs required to cover all use cases

# Section 4 Test Activities

### 4.1 User Actions

#### 4.1.1 Register New User

**Abstract**

Register a new valid Username and fill out a user profile

**Pre-requisites**

Use a valid Username and Password

**Criteria**

New user is created with password and updated user profile. Verify user profile by browsing to the Profile page.

**Method**

1. Browse to the Register page
2. Enter a valid Username
3. Enter a valid Password
4. Confirm your Password
5. Click Register
6. Fill in your info on Update Profile screen
7. Click Save changes
8. Verify Username now appears in top menu along with a new Profile option

**Variations**

1. Select various options while updating initial user profile

#### 4.1.2 User Login

**Abstract**

Log in with a valid User ID and password

**Pre-requisites**

A valid Username and Password combination already exists

**Criteria**

Existing user is logged into application

**Method**

1. Browse to the Sign In page
2. Enter a valid User ID and password combination
3. Clock Sign in
4. Ensure login is accepted and user is now logged in
5. Verify Username now appears in top menu along with a new Profile option

**Variations**

none

#### 4.1.3 User Logout

**Abstract**

Log out of an existing user account

**Pre-requisites**

Valid user account is already logged in

**Criteria**

User account is logged out of application and Profile option is not available.

**Method**

1. Select Sign out from top menu
2. Verify Signed Out screen appears
3. Verify Username and Profile no longer show in top menu

**Variations**

none

#### 4.1.4 Modify User Profile

**Abstract**

Modify user profile

**Pre-requisites**

Valid user account is already logged in

**Criteria**

Profile settings are updated and saved

**Method**

1. Select Profile from top menu
2. Click Update Profile
3. Update user profile options
4. Click Save changes
5. Select Profile from top menu
6. Verify new options have been saved correctly

**Variations**

1. Update different options in profile

### 4.2 Searching the Database

#### 4.2.1 Search While Logged In

**Abstract**

Perform a search while logged in to the application

**Pre-requisites**

User is already logged in

**Criteria**

Search page loads with options from the saved user profile

**Method**

1. Select Search from top menu
2. Ensure Location and Impacts default to saved profile options
3. Click Search Legislation
4. Verify results pertain to selected Location and Impacts
5. Click Search Again
6. Ensure Location and Impacts default to saved profile options

**Variations**

#### 4.2.2 Search While Logged Out

**Abstract**

Perform a search while not logged in to the application

**Pre-requisites**

No user is currently logged in

**Criteria**

Search page loads with nothing currently selected

**Method**

1. Select Search from top menu
2. Ensure Location and Impacts default to nothing selected
3. Select a Location from the drop-down list
4. Select one or more Impacts from the checkbox list
5. Click Search Legislation
6. Verify results pertain to selected Location and Impacts
7. Click Search Again
8. Ensure Location and Impacts default to nothing selected

**Variations**

1. Select various combinations of Location and Impacts (CTD will help cut down on test combinations while ensuring full test coverage of options)

### 4.3 Exporting Results

#### 4.3.1 Print Results

**Abstract**

Print search results

**Pre-requisites**

Search results contain a non-empty set

**Criteria**

User is able to print search results

**Method**

1. Search on selected Location and Impacts
2. On Search Results page, select Print
3. Ensure option to print appears (specific option will vary by user and OS)
4. Print results

**Variations**

1. Print different search results
2. Print from various OS and browsers
3. Print from both Logged in and Logged out states

#### 4.3.2 Download Results as CSV

**Abstract**

Download search results in CSV format

**Pre-requisites**

Search results contain a non-empty set

**Criteria**

User is able to download search results in CSV format

**Method**

1. Search on selected Location and Impacts
2. On Search Results page, select Download CSV
3. Ensure option to open or save file appears (specific option will vary by user’s browser preferences)
4. Save the file
5. View and verify the information in the saved file

**Variations**

1. Download different search results
2. Download from various OS and browsers
3. Download from both Logged in and Logged out states
4. Select Open file instead of Save file

#### 4.3.3 E-mail Results

**Abstract**

Send search results via e-mail

**Pre-requisites**

Search results contain a non-empty set and user is logged in

**Criteria**

User is able to receive search results to their e-mail address (for the initial deployment, mailtrap.io will be used to simulate an e-mail SMTP server)

**Method**

1. Search on selected Location and Impacts
2. On Search Results page, select Send Results
3. Ensure confirmation message appears for e-mailing search results
4. Check e-mail for search results and verify content

**Variations**

1. User real e-mail address with SMTP server
2. Verify both HTML and simple text versions available

### 4.4 Site Navigation

#### 4.4.1 Menu Options

**Abstract**

Ensure all menu options route to the correct page

**Pre-requisites**

none

**Criteria**

Each menu option should route user to the selected page

**Method**

1. From anywhere in app, select an option from the top menu
2. Ensure correct page is loaded

**Variations**

1. Select all menu options available
2. Browse from both Logged in and Logged out states

#### 4.4.2 Browser Back/Forward Button

**Abstract**

Ensure site is compatible with browser Back/Forward buttons

**Pre-requisites**

none

**Criteria**

User is able to navigate back and forward with browser buttons without error

**Method**

1. From any page in app, use browser buttons to move back or forward pages
2. Ensure no errors or inconsistent data is presented to user

**Variations**

1. Browse from both Logged in and Logged out states
2. Attempt on all pages available
3. Fill in data prior to browsing back/forward, then browse back to page

### 4.5 Bad Path

#### 4.5.1 Invalid User ID

**Abstract**

Ensure user is unable to log in with an invalid User ID

**Pre-requisites**

User is not currently logged in

**Criteria**

User should be unable to log in to application with an invalid User ID

**Method**

1. Browse to Sign In page via top menu
2. Input an invalid Username with a valid (known) password
3. Ensure the error message “Please enter a correct username and password. Note that both fields may be case-sensitive.” is displayed
4. Re-type the Username that corresponds to the entered password
5. Ensure User is able to now log in successfully

**Variations**

none

#### 4.5.2 Invalid Password

**Abstract**

Ensure user is unable to log in with an invalid password

**Pre-requisites**

User is not currently logged in

**Criteria**

User should be unable to log in to application with an invalid password

**Method**

1. Browse to Sign In page via top menu
2. Input a valid Username with an invalid password
3. Ensure the error message “Please enter a correct username and password. Note that both fields may be case-sensitive.” is displayed.
4. Re-type the Password that corresponds to the entered Username
5. Ensure User is able to now log in successfully

**Variations**

None

#### 4.5.3 Invalid Registration

**Abstract**

Ensure user is unable to register without following Username and Password requirements as shown on Register page

**Pre-requisites**

User is not currently logged in

**Criteria**

User should be unable to register unless all Username and Password guidelines are followed

**Method**

1. Browse to Register page via top menu
2. Input invalid Username and/or Password into the Register fields
3. Ensure the proper error message is displayed according to the restriction that was not followed.
4. Repeat to cover all potential restrictions

**Variations**

1. All potential restrictions should be tested. These restrictions could change in future versions of the tool.
2. Invalid Username and invalid Password scenarios should both be tested.
3. Password confirmation field does not match should also be tested.