

# Testing Plan for Dog Dates



## *ChangeLog*

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
1.0.0	06/03/2022	Shawn	Original Log

# 1. Introduction

## 1.1 Scope

---

We will be testing all the main features of the app as they are implemented.

Sprint 2:

- Home Page

- Sign Up (POST)

- Log In (POST)

- User Profile (GET, PATCH, DELETE)

Sprint 3 & 4:

- Like (GET, PATCH)

- MATCH (GET, PATCH)

## 1.2 Roles and Responsibilities

---

Each Sprint will have users taking on different roles. For sprint 2 the roles where:

Name	GitHub username	Role
Sammul	Sammulto	Full Stack Development
Liza	ZhijieZheng-UM	Full Stack Development
Zhijie	LizaYa	Operations, Testing
Shawn	Aiphox	Operations, Testing

## 2. Test Methodology

### 2.1 Test Levels

#### Unit and Integration Testing

For sprint 2 the front end and back end will have automated unit and integration testing. Some features, like correct page loading and redirection will be unit testing, but features that involve multiple levels of the program will be both unit and integration tested. This will include the following main features:

- Home / Login
  - o We will make sure that the page loads and displays correctly as well as buttons redirect to the correct pages
  - o We will also test login to make sure it returns the proper results
- Sign up (front end and back end)
  - o We will test that the page loads and displays correctly and that buttons submit information and redirect to the correct pages.
  - o We will also test field validation for the submission form and that it returns the proper codes upon submission.
- User Account (front end and back end)
  - o We will test account access (no access if expired login)
  - o Have appropriate warnings and verifications displayed when deleting or updating an account
  - o We will make sure that get/ update/ delete user info is sending the proper information.

#### Acceptance Testing

Although passing the unit and integration tests will signal that the program works and is bug free, it is important as a user to go through the site and make sure it performs properly for a human. Acceptance testing will be executed as follows:

User Executes	Expected Response
On home page click the register button	User should be taken to the registration page
Fill in the fields with your information and click sign up	User should be taken to the account page where your name, dogs name, location, description, and picture are displayed
Click update my profile	User should be taken to update account profile page
Change the fields and click update profile	User should be shown a confirmation that the account was updates and a prompt to log in again.
Click Log In at the top right corner of the screen	User should be taken to the log in page
Enter information and click log in	User should be taken to the account page and see account information
Click delete account	User should be shown a prompt warning them that they are deleting their account
Click go back	User should return to the account page
Click delete account and then click delete my account	User should see a page saying we hope to see you again
Click log in at the top right corner and enter log in information	User should not be allowed to log in since account is deleted
Click Dog Dates button at the top left	User should be brought back to the home page

## 2.2 Test Completeness

---

We expect all tests to pass for each push to the associated directory.

# 3. Resource & Environment Needs

## 3.1 Testing Tools

---

For our automated testing we use GitHub Actions – Automated Testing.. This will automatically run our React Testing Library tests for the backend when there is a push to the backend directory, and test the frontend when there is a push to the frontend directory.

## 3.2 Test Environment

---

Testing is done remotely by GitHub. It is agnostic to the user's machine as it creates it's own Linux Ubuntu environment to run the tests.

# 4. Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test
CI / CD	Continuous Integration / Continuous Deployment
MERN	MongoDB, Express, React, Node.js