Testing Report

*Project Planner*

**Client**

Doug Craig

**Team Hailstorm**

Hemang Bhatt

Aidan Gray

Cameron Hensel

Jacob Lutz

Phuoc Nguyen

Nirav Pancholi

29 November, 2017

# 

|  |  |
| --- | --- |
| **Table Of Contents** | Page |
| 1. Introduction | 2 |
| 1.1 Purpose of this Document | 2 |
| 1.2 References | 2 |
| 2. Testing Process | 2 |
| 2.1 Description | 2 |
| 2.2 Testing Sessions | 3 |
| 2.3 Impressions of the Process | 3 |
| 3. Test Results | 4 |
| 3.1 Test Suite | 4 |
| 3.2 Test Results | 15 |
| Appendix A - Agreement between Customer and Contractor | 30 |
| Appendix B - Team Review Sign-Off | 32 |

### **1. Introduction**

1.1 Purpose of This Document

The purpose of this document is for system administrators and quality assurance teams to be able to understand and analyze the methods in which our software was tested.

1.2 References

This document makes references to the *Project Planner* System Requirements Specification Document, Software Design Document, and User Interface Design Document.

**2. Testing Process**

2.1 Description

We used a non-standard, custom testing process to ensure accuracy and reliability of the Project Planner software, specifically tested input validation, different modules of php, web page links, and proper execution flow of the web application. After implementing each module, the developers informally tested individual modules and their expected outputs. For the integration and configuration testing, we followed happy path testing to perform the setup of the database system and files. Following the configuration test, we performed happy path testing to verify the flow of the execution using different test cases and use cases and recorded defects and bugs.

Additionally, we performed input validation testing and user interface testing to measure the effectiveness and security of the program. Furthermore, we tested security of the application using incorrect combinations of user ID and password, session testing, and improper privilege escalation. Lastly, we verified that the features of the application meets what we designed and implemented.

2.2 Testing Sessions

**Meeting #1**

Date: 11/27/17

Location: CMSC 447 Classroom

Start Time: 1:30pm

End Time: 2:30pm

Participation: Full group participation

Units Tested/Description: Logging in, Logging out, Creating, Editing, and Deleting Projects, Tasks, and Phases

**Meeting #2**

Date: 11/28/17

Location: CMSC 447 Classroom

Start Time: 1pm

End Time: 5:15pm

Participation: Full group participation

Units Tested/Description: Creating, editing, and deleting users, adding users to phases and removing users from phases, first-time setup

**Meeting #3**

Date: 11/29/17

Location: CMSC 447 Classroom

Start Time: 1:30 pm

End Time: 8pm

Participation: Full group participation

Units Tested/Description: Additional testing of all modules as well as edge case handling and SQL injection testing

2.3 Impressions of the Process

Our overall impression of the process is that the methods we used were successful and thorough, however, there are certainly improvements that could be made. We feel that having the people who implemented the code perform informal testing as they completed the modules allowed us to fix many errors before starting the formal testing process. Additionally, having the entire team work together to create the Use Case Test Suite ensured that all members had input on which modules they thought required the most rigorous testing. Finally, having a team member who did not take part in the implementation perform the majority of the formal testing allowed us to remove bias from the process. If we were go through the software development cycle again, we would perform formal testing at more frequent intervals throughout the cycle.

Of all our modules, we believe that the two least likely to contain many remaining flaws would be the modules for project/phase/task deletion (Delete.php) and for changing a user’s role (Change\_Role.php). The deletion modules are the most secure in that they require very little input - what input it does require is automated and relatively hidden from the user. Additionally, the modules perform a singular specific function and nothing else, so additional flaws would become obvious if the modules’ single function was not performed. Changing the user’s role is similar: the module performs a single function which updates immediately, and if this does not occur, then flaws are obvious. For security, extensive attention was given to ensure that only the intended users - any manager besides whichever one is currently logged into the session - can access this function.

On the other hand, the two modules most likely to contain remaining flaws would be the modules for user account creation and editing. Both modules require multiple input fields, all of different types, from the user, and this input is used for multiple SQL queries. These modules also vary widely depending on the user logged in and his or her role, with certain functions appearing for managers, and others being hidden for the user currently logged in. This leaves open the possibility of SQL injection, mismatched data types, and permission escalation in unintended situations. While we tested for these problems and believe we have resolved them, the nature of these modules does not guarantee our resolutions are complete.

**3. Test Results**

3.1 Testing Suite

The following tables describe each use case that was tested, along with the results of valid and invalid inputs.

|  |  |
| --- | --- |
| **Use Case** | **Login (Manager)** |
| Valid Username & Password | Test credentials -  Username: manager1  Password: manager |
| Invalid Username and/or Password | Test credentials -  Username: manager1  Password: badPass |
| Purpose | Test the ability to login as a manager |
| Expected Valid Result | Successfully login and view the project homepage. |
| Expected Invalid Result | Fail to login and receive an “Invalid username or password” warning. |
| Boundary Conditions | Invalid login credentials results in error |

|  |  |
| --- | --- |
| **Use Case** | **Login (Employee)** |
| Valid | Test credentials -  Username: employee1  Password: employee |
| Invalid | Test credentials -  Username: employee1  Password: badPass |
| Purpose | Test the ability to login as employee |
| Expected Valid Result | Successfully login and view the project homepage with only the projects that employee is assigned to. |
| Expected Invalid Result | Fail to login and receive an “Invalid username or password” warning. |
| Boundary Conditions | Invalid login credentials results in error |

|  |  |
| --- | --- |
| **Use Case** | **Logout** |
| Situation | User clicks the sign out button. |
| Purpose | Test the ability to logout of an account. |
| Expected Result | User is logged out and taken to the login screen. |

|  |  |
| --- | --- |
| **Use Case** | **Create Project** |
| Valid Parameters | Project name, client (or none) selected, project status, positive integer budget, selected date, and test description (or empty) entered. |
| Invalid Parameters | Any of the above fields not filled, or a negative budget entered. |
| Purpose | Test the ability to create a new project and add it to the database. |
| Expected Valid Result | A new project is created, the user is redirected to the project page, and the new project is displayed. |
| Expected Invalid Result | The user is shown which fields have not been entered correctly and the project is not created. |
| Boundary Conditions | Invalid budget, empty project name, unselected project status, or unselected date all result in error. |

|  |  |
| --- | --- |
| **Use Case** | **Edit Project** |
| Valid Parameters | Project name, client (or none) selected, project status, positive integer budget, selected date, and test description (or empty) are unchanged or changed correctly. |
| Invalid Parameters | Any of the above fields are changed to be unfilled or budget is changed to a negative value. |
| Purpose | Test the ability to edit the details of a pre-existing project. |
| Expected Valid Result | The user is taken back to the project overview page with the new details being displayed. |
| Expected Invalid Result | The user is shown which fields have not been edited correctly and the project details are not saved. |
| Boundary Conditions | Invalid budget or empty project name will result in error. |

|  |  |
| --- | --- |
| **Use Case** | **Delete Project** |
| Situation | The user (as a manager) clicks on the “Delete Project” button. |
| Purpose | Test the ability to delete an existing project. |
| Result | The user is prompted with a message asking if they are sure they would like to delete the project. Upon selecting “OK”, the project is removed from the database, the user is taken back to the homepage, and the project no longer appears in the list of projects. |

|  |  |
| --- | --- |
| **Use Case** | **Add Client** |
| Valid | Company name, Contact first and last name, Contact email, Contact phone number, Specialized Industry, Street address, City, State, Zip code, Country |
| Invalid | Any required field not filled, Zip code not a number, Contact phone number not a number, State length longer than 2, Zip code longer than 5, Contact phone number longer than 10 |
| Purpose | Test the ability to add a client to the database. |
| Expected Valid Result | A new client is added, the user is redirected to the Create Project (or Edit Project) page depending on the previously-visited page, and a new client can be selected on that page. |
| Expected Invalid Result | The user is shown which fields are invalid or empty, or input fields stop accepting input if the character limit is exceeded. |
| Boundary Conditions | Invalid form data is either not accepted or results in an error. |

|  |  |
| --- | --- |
| **Use Case** | **View Project** |
| Situation | The user selects a project from the homepage by clicking on it. |
| Purpose | Test the ability to view a project’s details. |
| Result | The user is brought to the project overview page which shows the navigation tree on the left, with the phases and tasks of those phases, as well as the project’s full details on the right. |

|  |  |
| --- | --- |
| **Use Case** | **Create Phase** |
| Valid Parameters | The phase name and phase description fields must be filled out. |
| Invalid Parameters | Either the phase name or phase description fields are not filled out. |
| Purpose | Test the ability to create a new phase for a project. |
| Expected Valid Result | The phase is created and added to the database and the user is returned to the project details page. |
| Expected Invalid Result | The phase is not created and the user is shown which fields have been entered incorrectly. |
| Boundary Conditions | Invalid form data results in an error. |

|  |  |
| --- | --- |
| **Use Case** | **View Phase** |
| Situation | The user selects a phase from the homepage by clicking on it. |
| Purpose | Test the ability to view a phase’s details. |
| Result | The user is brought to the phase details screen with the ability to edit the phase, delete the phase, or return to the project. |

|  |  |
| --- | --- |
| **Use Case** | **Edit Phase** |
| Valid | Phase name, Phase description, estimated budget and estimated hours are all filled in. Estimated budget and estimated hours have nonnegative values |
| Invalid | Any input is not filled in and/or a negative value is entered into estimated budget or estimated hours |
| Purpose | Test ability to update an existing phase |
| Expected Valid Result | The user is returned to the phase view page with the updated information displayed |
| Expected Invalid Result | The user is not returned to the phase view page. The page displays a message indicating invalid values |
| Boundary Conditions | Invalid form data results in an error |

|  |  |
| --- | --- |
| **Use Case** | **Search for User** |
| Valid (Existing) User Entered | Test data: jacob |
| Invalid (Non-existing) User Entered | Test data: aidan |
| Purpose | Test the ability to search for an existing user and assign them to a phase. |
| Expected Valid Result | All users containing the name entered are displayed. |
| Expected Invalid Result | No users are displayed. |
| Boundary Conditions | Non-existing users produces no display. |

|  |  |
| --- | --- |
| **Use Case** | **Delete Phase** |
| Situation | The user clicks the “Delete Phase” button from the phase details screen. |
| Purpose | Test the ability to delete an existing phase. |
| Result | The user is prompted with a message asking if they are sure they would like to delete the phase. Upon selecting “OK”, the phase is removed from the database, the user is taken back to the project overview page, and the phase, along with its tasks, are removed from the navigation tree. |

|  |  |
| --- | --- |
| **Use Case** | **Create Task** |
| Valid | Name, description, hours, and budget fields are filled out and the hours/budgets fields do not contain any non-negative values |
| Invalid | Any field is not filled in or a negative value is entered into the budget or hours fields |
| Purpose | Test ability to create a new task |
| Expected Valid Result | A new task is created and the user is returned to the project view page. The created task is displayed under the appropriate phase |
| Expected Invalid Result | The user is not returned to the project view and the page highlights which fields need to be filled in correctly |
| Boundary Conditions | Invalid form data creates an error |

|  |  |
| --- | --- |
| **Use Case** | **Edit Task** |
| Valid | Name, description, hours, and budget fields are filled out and the hours/budgets fields do not contain any non-negative values |
| Invalid | Any field is not filled in or a negative value is entered into the budget or hours fields |
| Purpose | Test ability to edit an existing task |
| Expected Valid Result | User is returned to the project view page and the updated task information is displayed |
| Expected Invalid Result | User is not returned to the project view page and the empty/incorrect fields are highlighted |
| Boundary Conditions | Invalid form data creates an error |

|  |  |
| --- | --- |
| **Use Case** | **Delete Task** |
| Situation | User clicks the delete task button on the task view page |
| Purpose | Tests the ability to delete an existing task |
| Result | User is prompted with confirmation to delete the task. When ok is clicked the task is deleted and the user is returned to the previous screen |

|  |  |
| --- | --- |
| **Use Case** | **Return to Project** |
| Situation | The user clicks the “Return to Project” button from either the phase or task details page. |
| Purpose | Test the ability to return the user to the project overview page. |
| Result | The user clicks the button and is immediately returned to the project overview page. |

|  |  |
| --- | --- |
| **Use Case** | **Return to Home** |
| Situation | The user clicks the “Return to Home” button from the project overview page. |
| Purpose | Test the ability to return the user to the homepage displaying all of the projects. |
| Result | The user clicks the button is immediately returned to the homepage. |

|  |  |
| --- | --- |
| **Use Case** | **View Account** |
| Situation | The user clicks on their username from the homepage. |
| Purpose | Test the ability to view a user’s account information |
| Result | The user is taken to a new page that displays their username, role (manager/employee), user id#, password (starred \*\*\* out), date of birth, address, email address, phone number, and ability to manage other users (managers only). |

|  |  |
| --- | --- |
| **Use Case** | **Edit User Account** |
| Valid Parameters | User’s first and last name, username, date of birth, address, email address, and phone number must be left unchanged or changed to something else, but not left blank. |
| Invalid Parameters | Any of the fields mentioned above are changed and left blank. |
| Purpose | Test the ability to change a user’s account details. |
| Expected Valid Result | The user’s details are changed in the database and the user is brought back to their account details. |
| Expected Invalid Result | The user is shown which fields are not properly filled out and the changes are not saved to the database. |
| Boundary Conditions | Invalid form data results in error. |

|  |  |
| --- | --- |
| **Use Case** | **Delete User** |
| Situation | User clicks the delete user button on the user view page |
| Purpose | Test the ability to delete a user’s information |
| Result | The user is prompted for confirmation twice. After agreeing on each alert the users account is deleted |

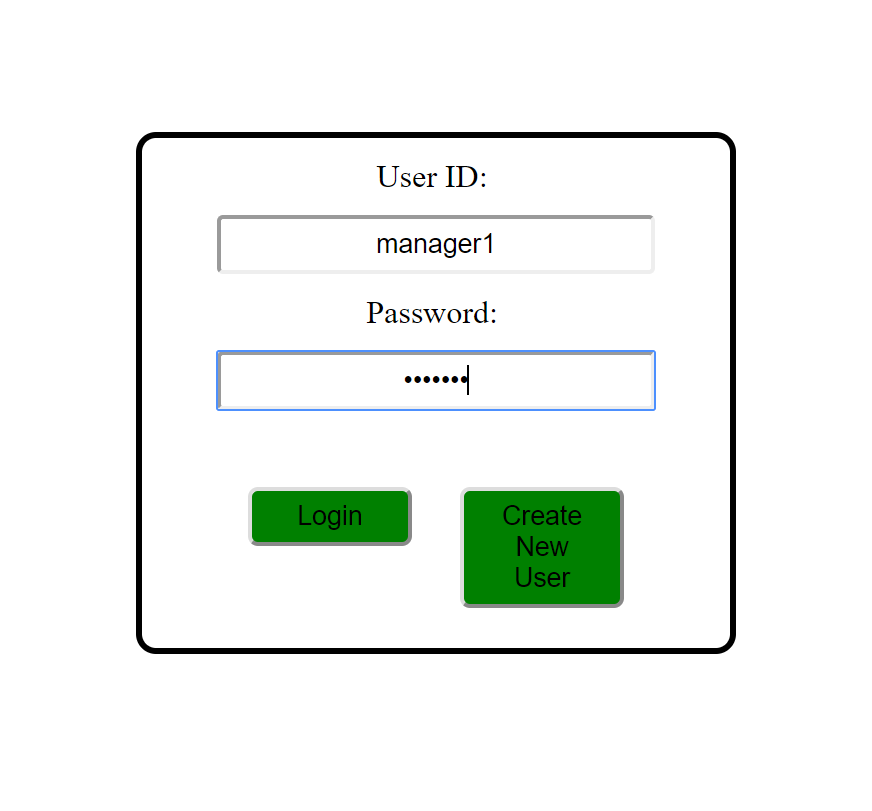
|  |  |
| --- | --- |
| **Use Case** | **Create New User** |
| Valid Parameters | User’s first and last name, username, password, date of birth, address, email address, and phone number must not be left blank. |
| Invalid Parameters | Any of the fields mentioned above are left blank. |
| Purpose | Test the ability to create a new user account. |
| Expected Valid Result | The user’s account is created, added to the database, and the current manager is returned to the account details page. |
| Expected Invalid Result | The user (manager) is shown which fields are not filled in correctly and the account is not created. |
| Boundary Conditions | Invalid form data results in error. |

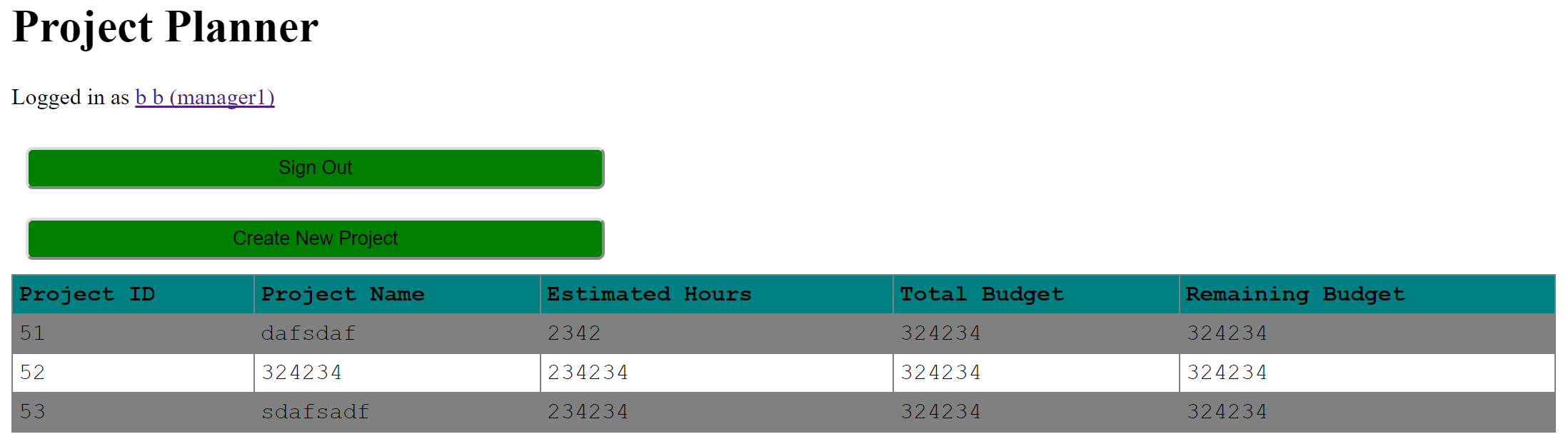
3.2 Test Results

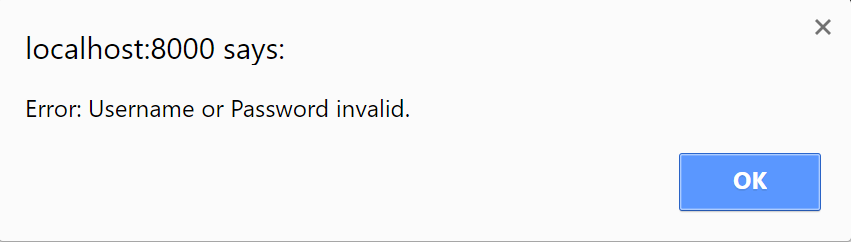
The following are the results from performing each of the tests outlined in the Test Suite. The screenshots displayed do not reflect the final look of our software. The updated visuals and design are completed but have not been merged with the actual functionality, and will not affect any testing results once implemented.

**Use Case: Login (Manager)**

Conducted by: Aidan Gray

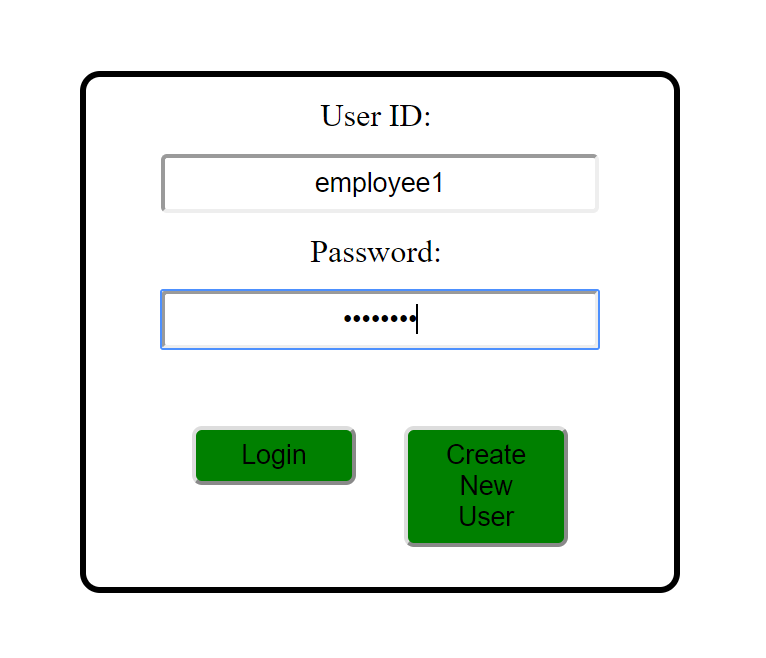
Valid Username & Password:  


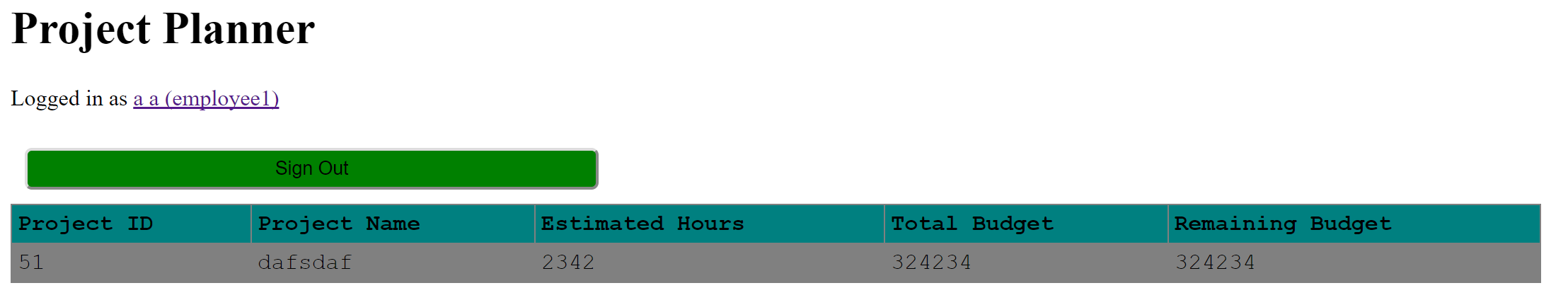


Invalid Username & Password:**

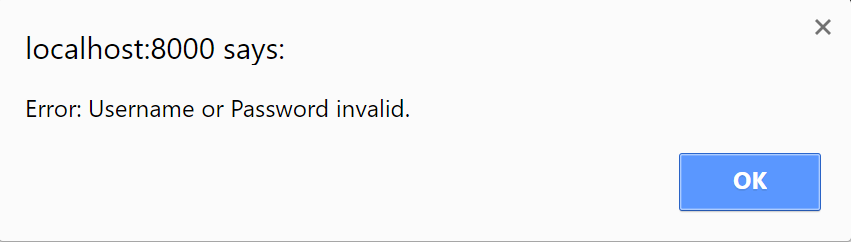
**Use Case: Login (Employee)**

Conducted by: Aidan Gray

Valid Username & Password:  




Invalid Username & Password:

**

**Use Case: Logout**

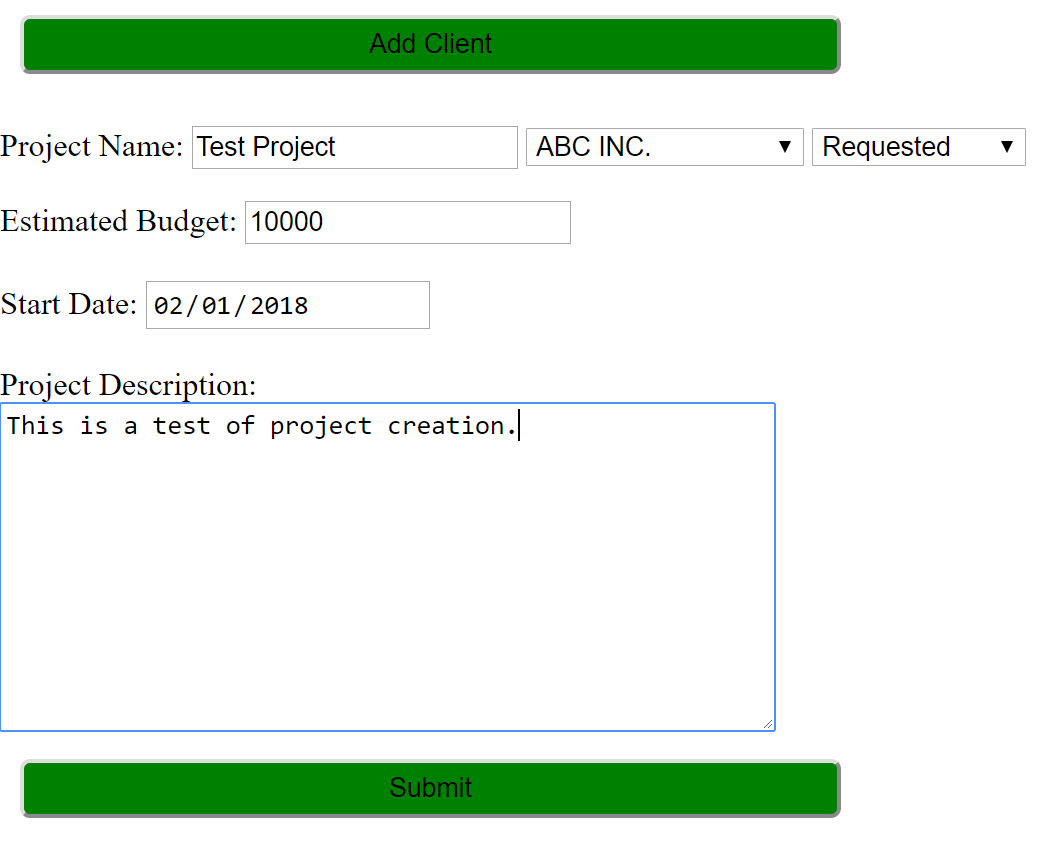
Conducted by: Aidan Gray

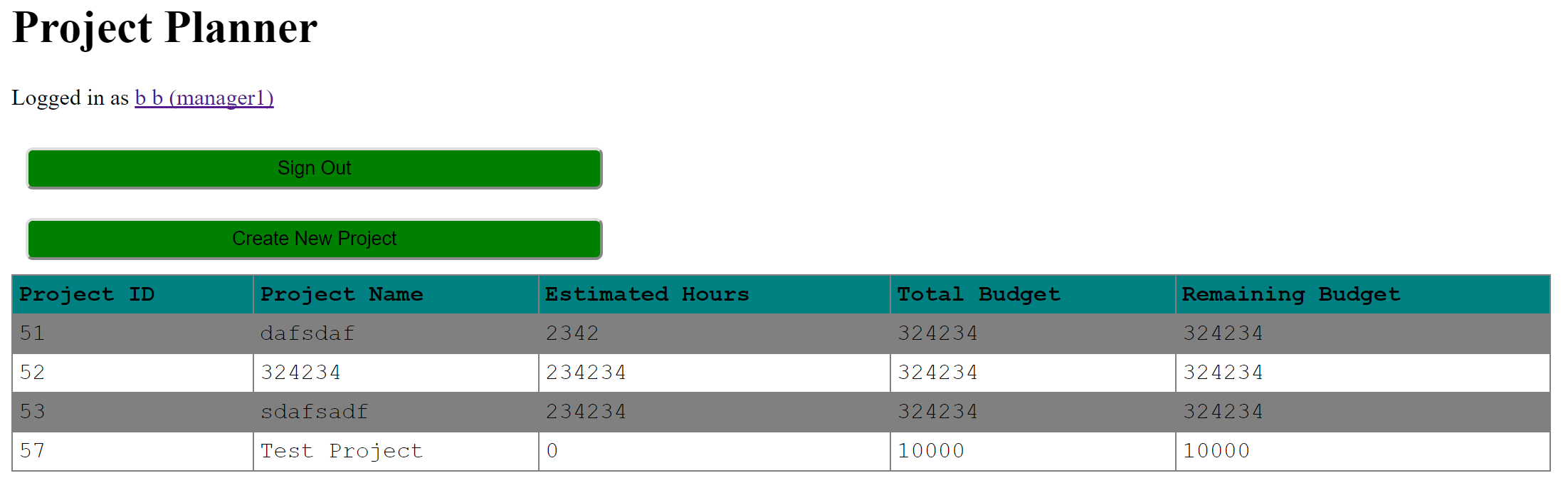
Situation: Click the “Sign Out” button.

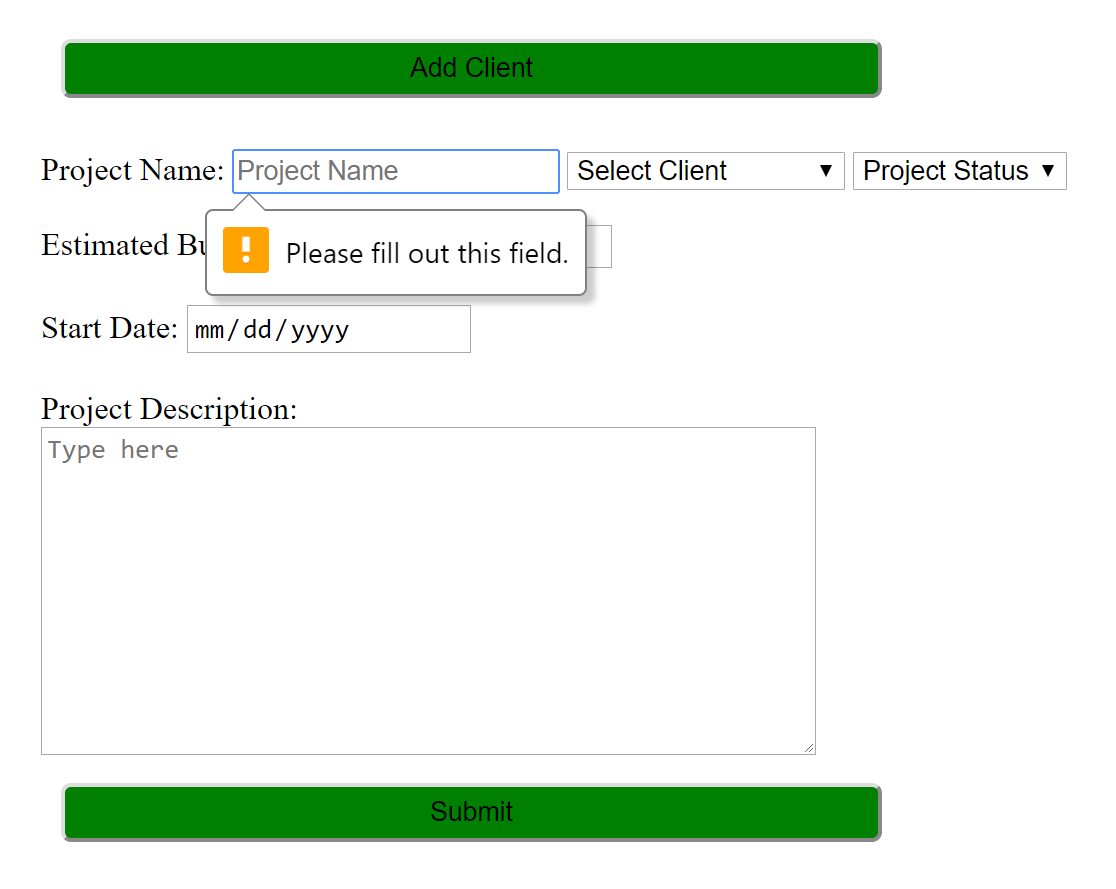
Result: Successfully logout.

**Use Case: Create Project**

Conducted by: Aidan Gray

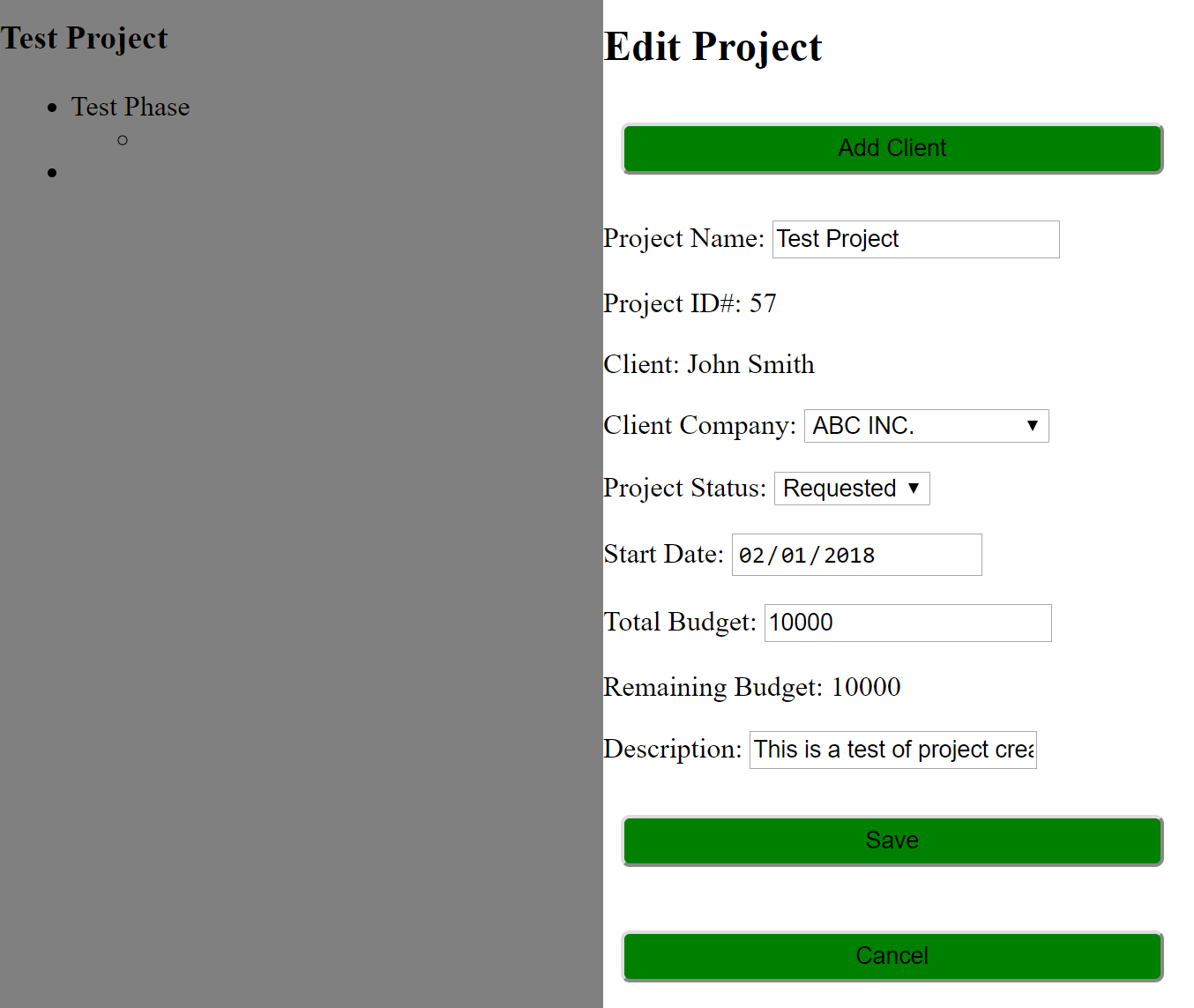
Valid Parameters:  


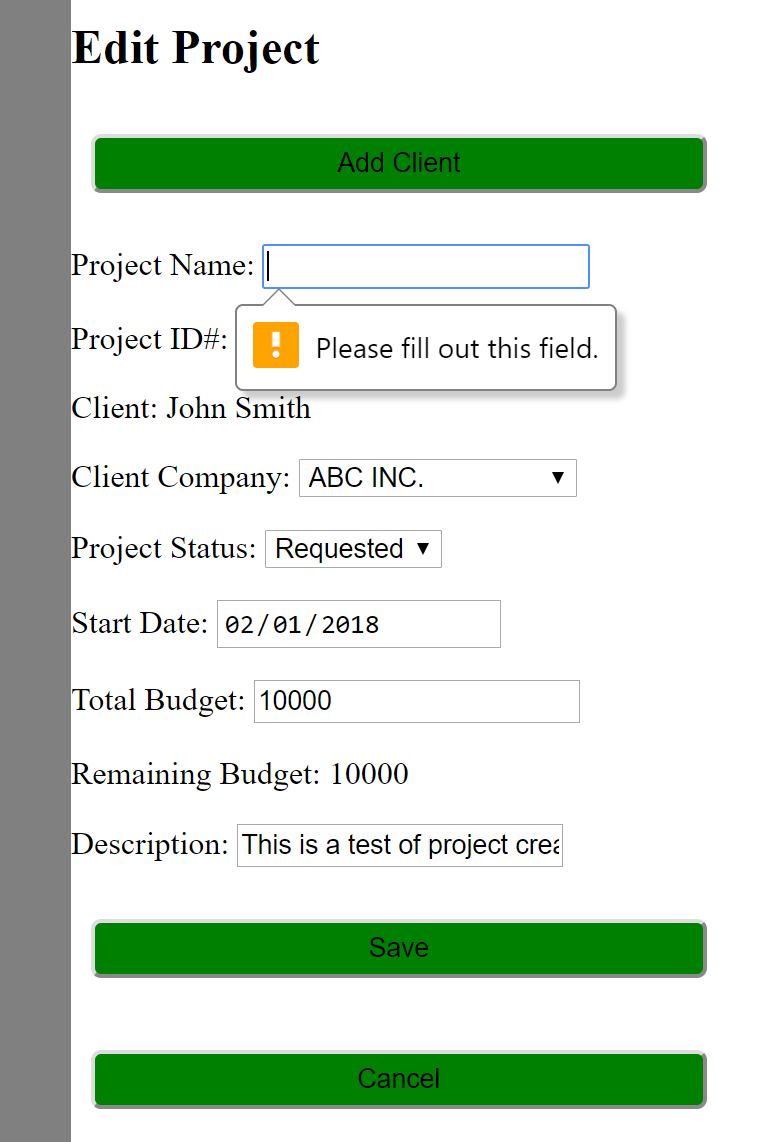


Invalid Parameters:  


**Use Case: Edit Project**

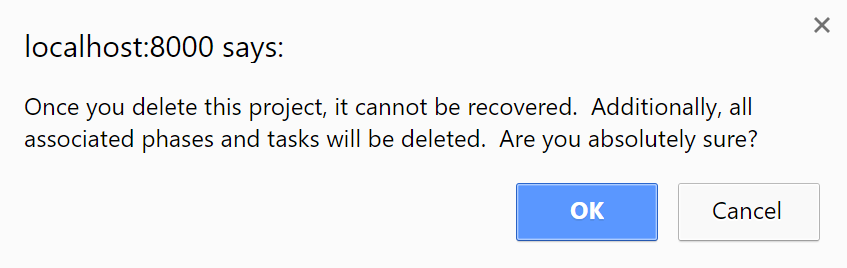
Conducted by: Aidan Gray

Valid Parameters: 

Invalid Parameters:   


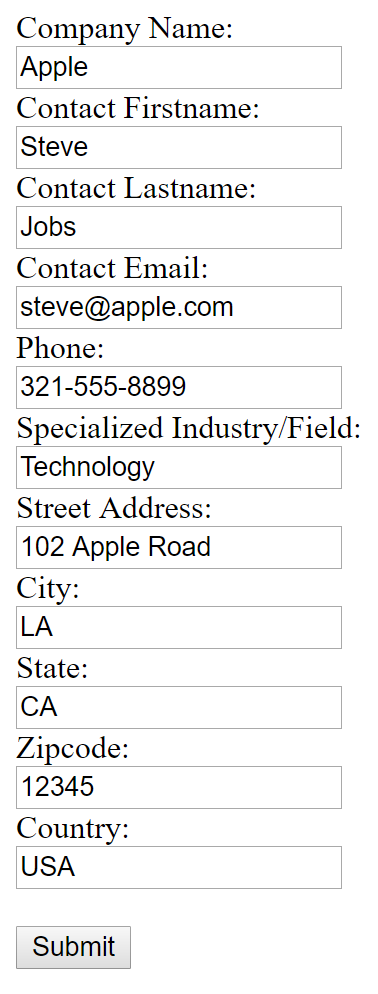
**Use Case: Delete Project**

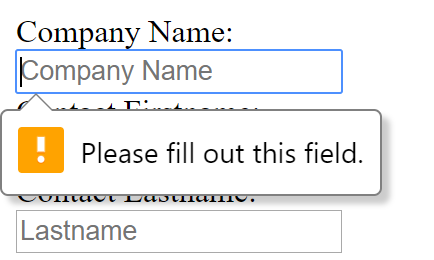
Conducted by: Aidan Gray

Result (Confirmation):  


**Use Case: Add Client**

Conducted by: Aidan Gray

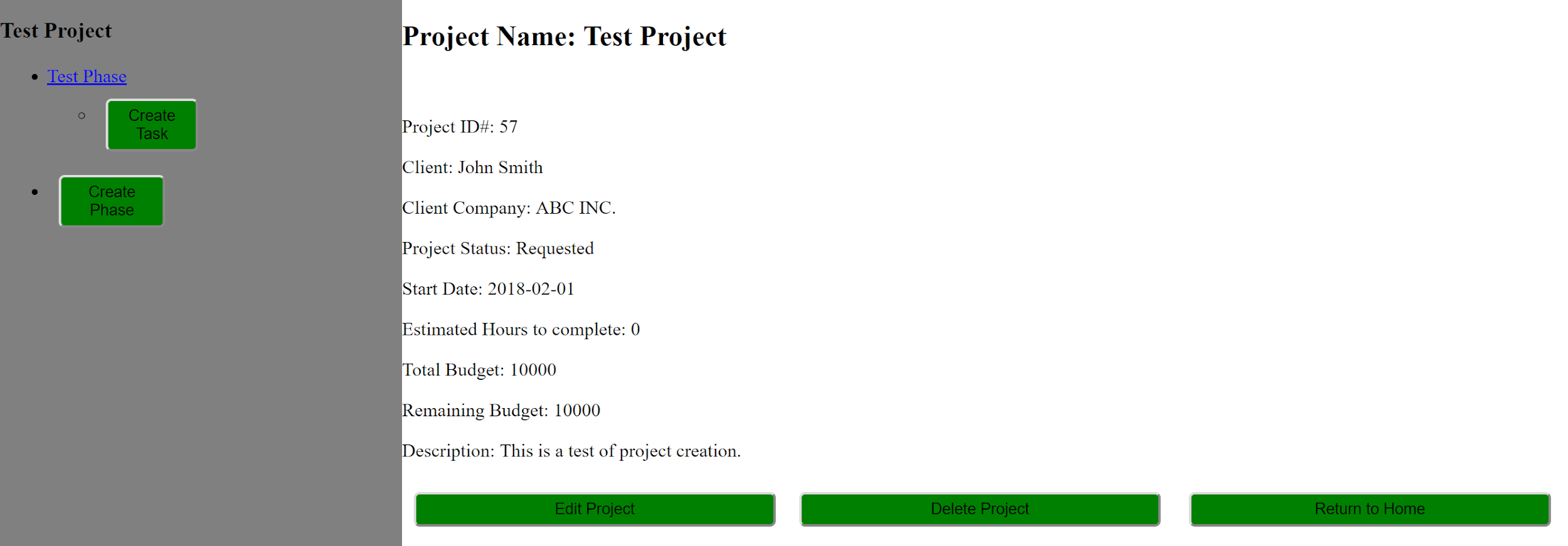
Valid Parameter:  


Invalid Parameters:  


**Use Case: View Project**

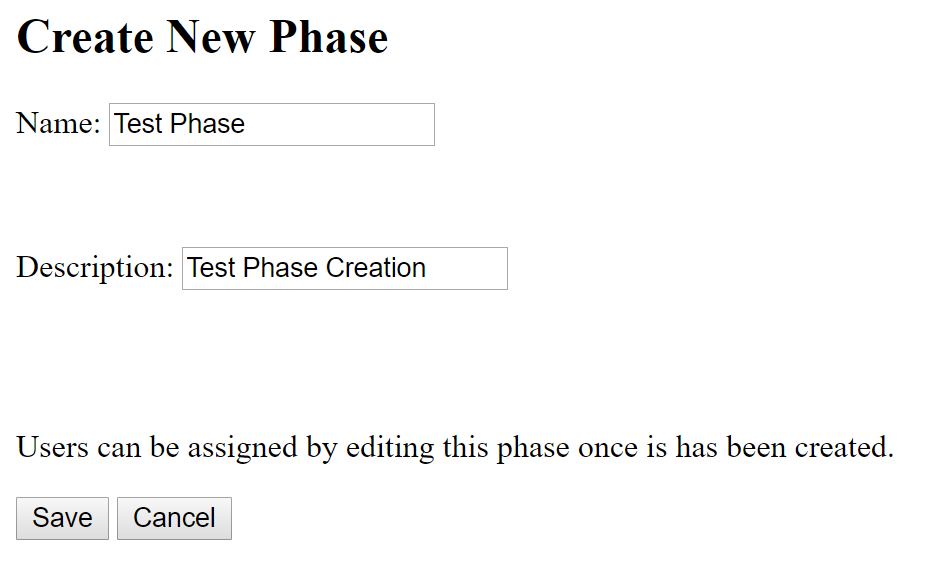
Conducted by: Aidan Gray

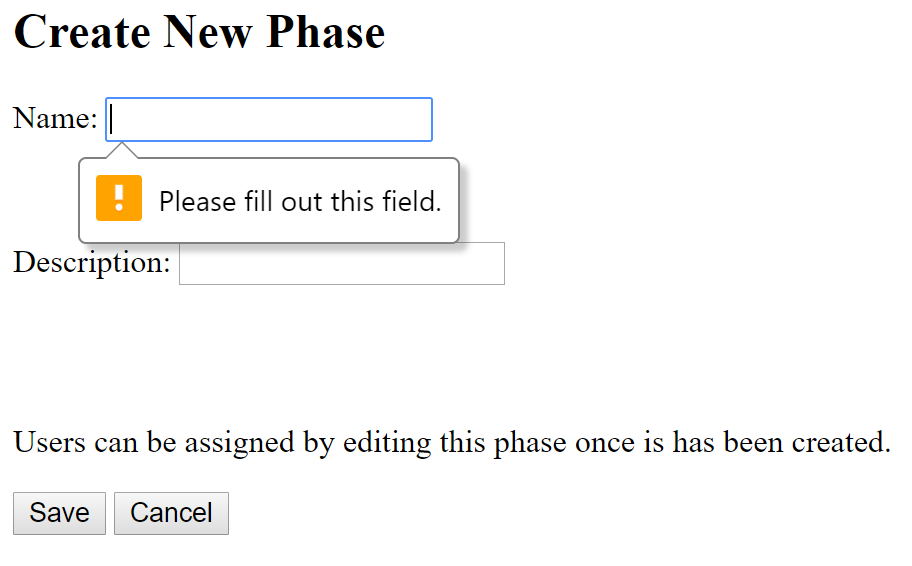
Result:



**Use Case: Create Phase**

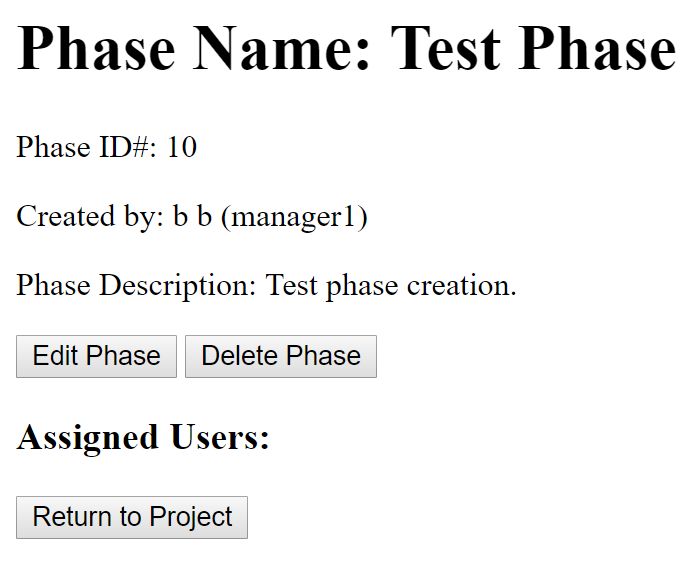
Conducted by: Aidan Gray

Valid Parameters:  


Invalid Parameters:  


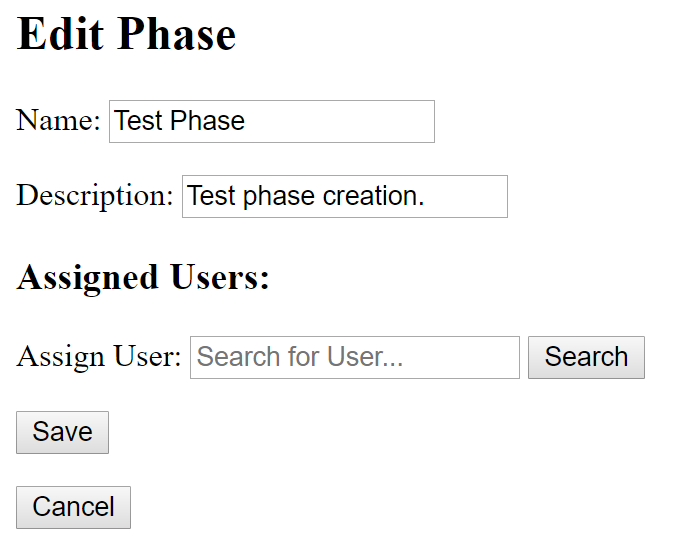
**Use Case: View Phase**

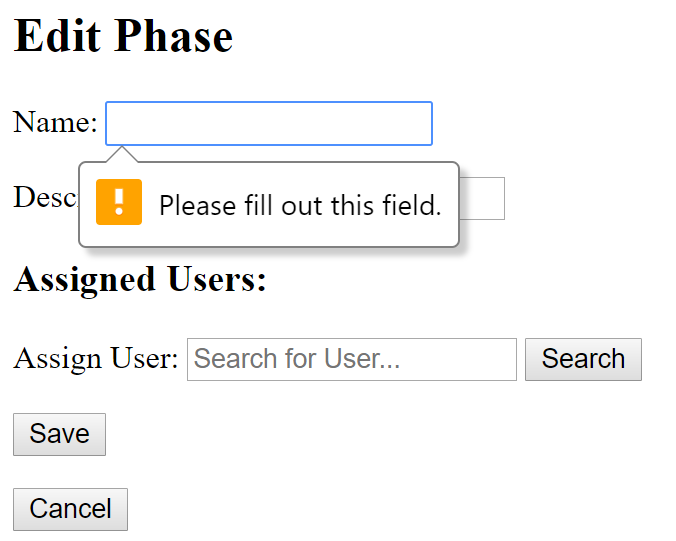
Conducted by: Aidan Gray

Result:  


**Use Case: Edit Phase**

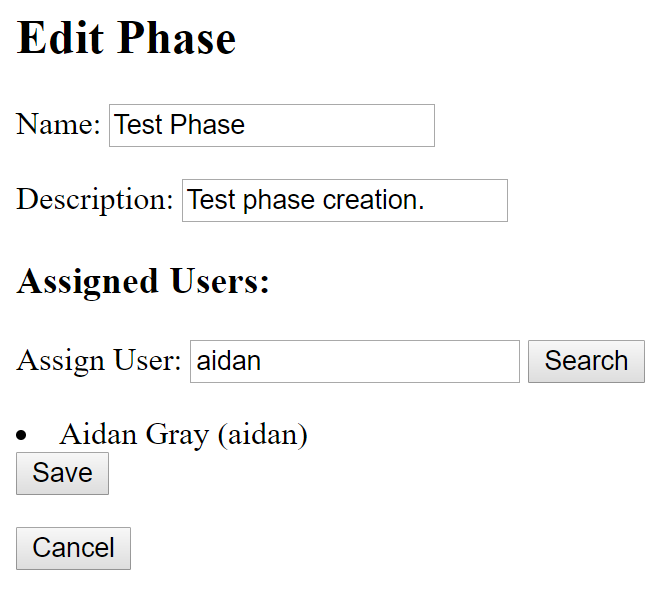
Conducted by: Aidan Gray

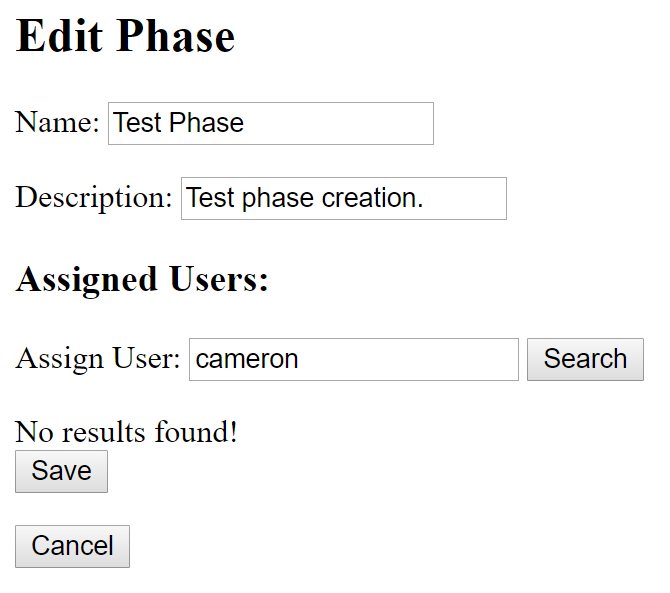
Valid Parameters:  


Invalid Parameters:   


**Use Case: Search for User**

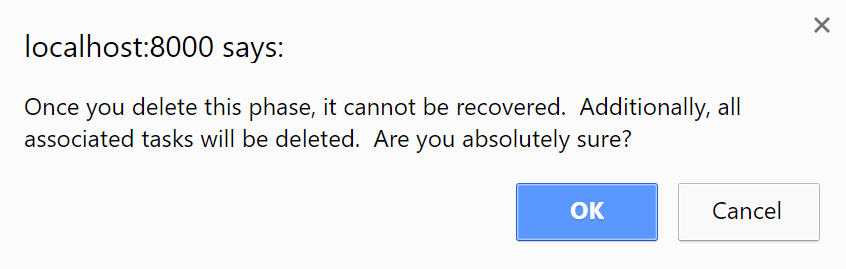
Conducted by: Aidan Gray

Valid User:  


Invalid User:  


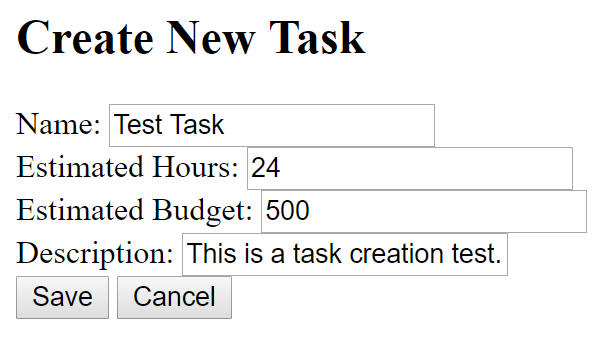
**Use Case: Delete Phase**

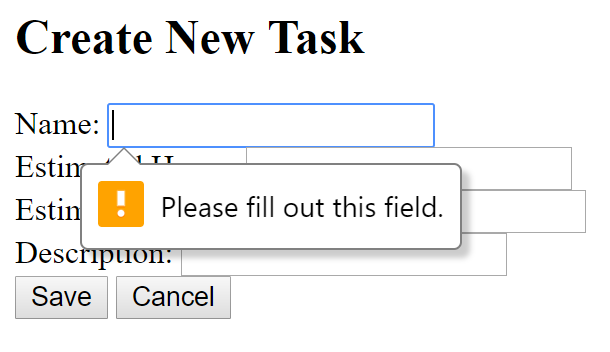
Conducted by: Aidan Gray

Result (Confirmation):  


**Use Case: Create Task**

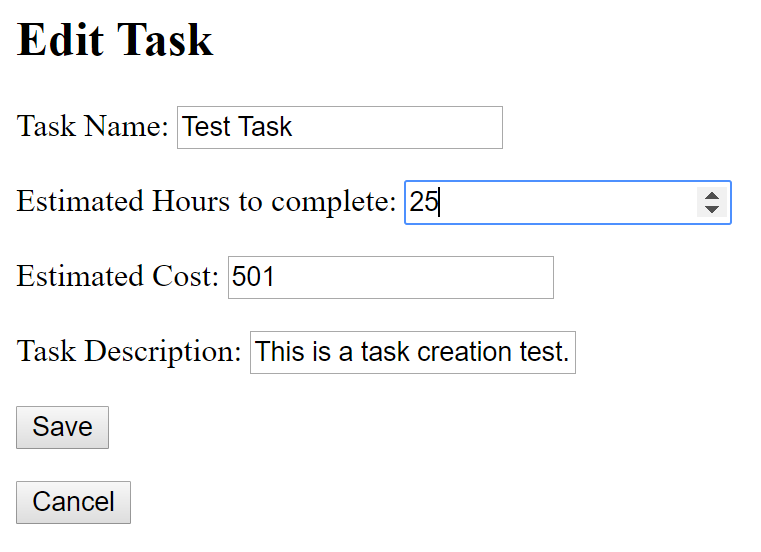
Conducted by: Aidan Gray

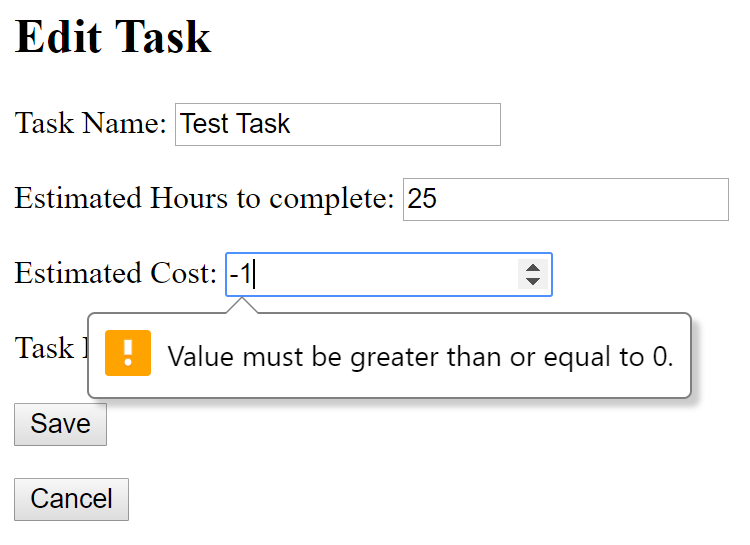
Valid Parameters:  


Invalid Parameters:  


**Use Case: Edit Task**

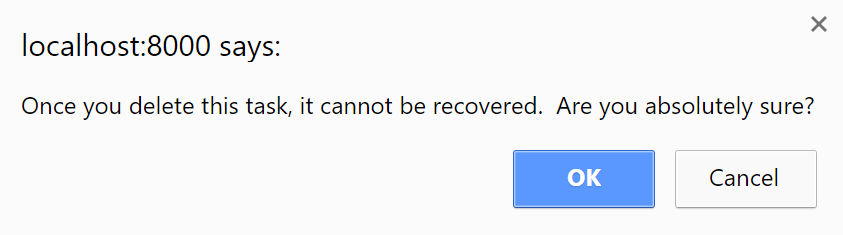
Conducted by: Aidan Gray

Valid Parameters:  


Invalid Parameters:  


**Use Case: Delete Task**

Conducted by: Aidan Gray

Result (Confirmation):  


**Use Case: Return to Project**

Conducted by: Aidan Gray

Situation: Click the “Return to Project” button.

Result: Successfully returned to the project overview.

**Use Case: Return to Home**

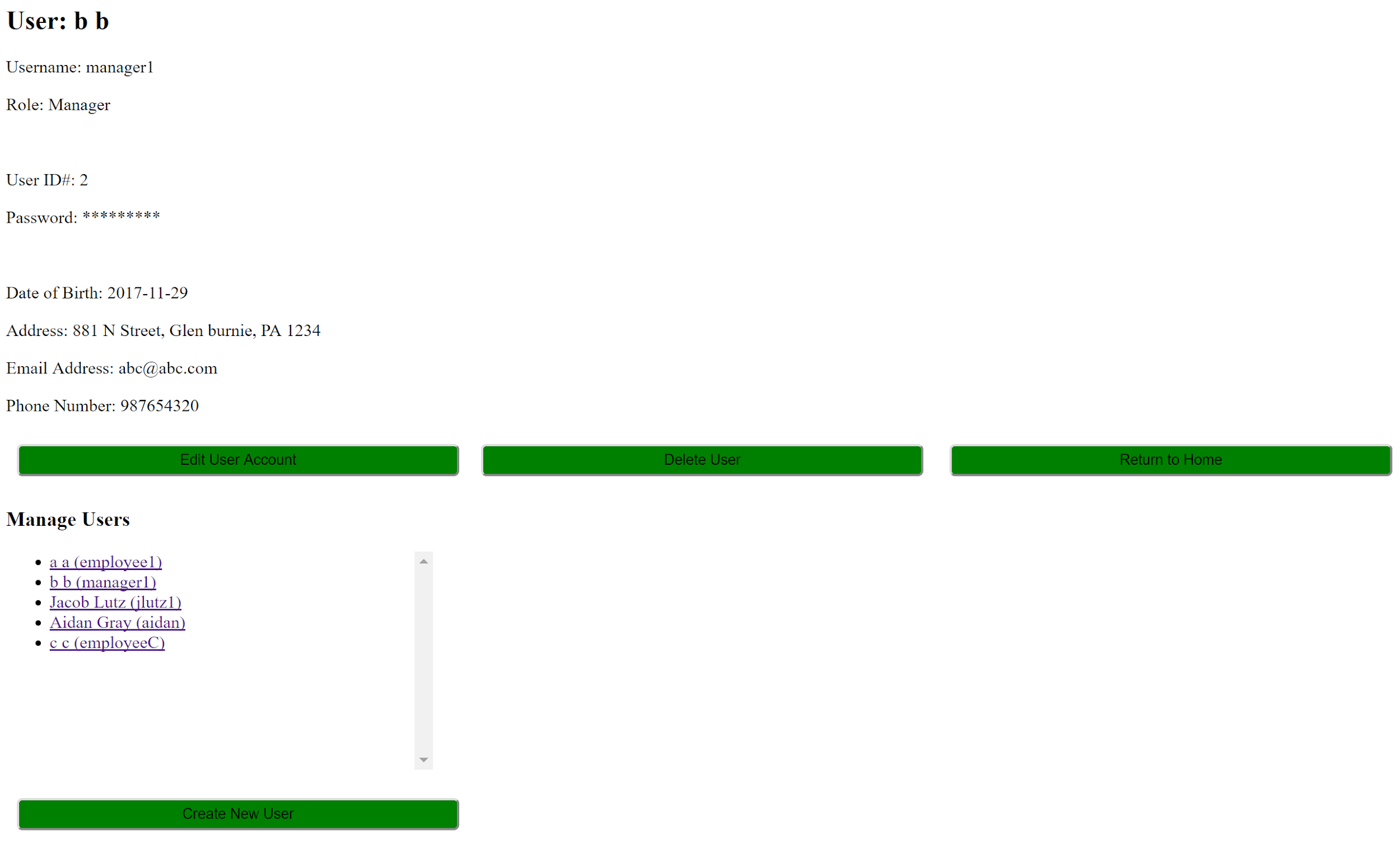
Conducted by: Aidan Gray

Situation: Click the “Return to Home” button.

Result: Successfully returned to the homepage.

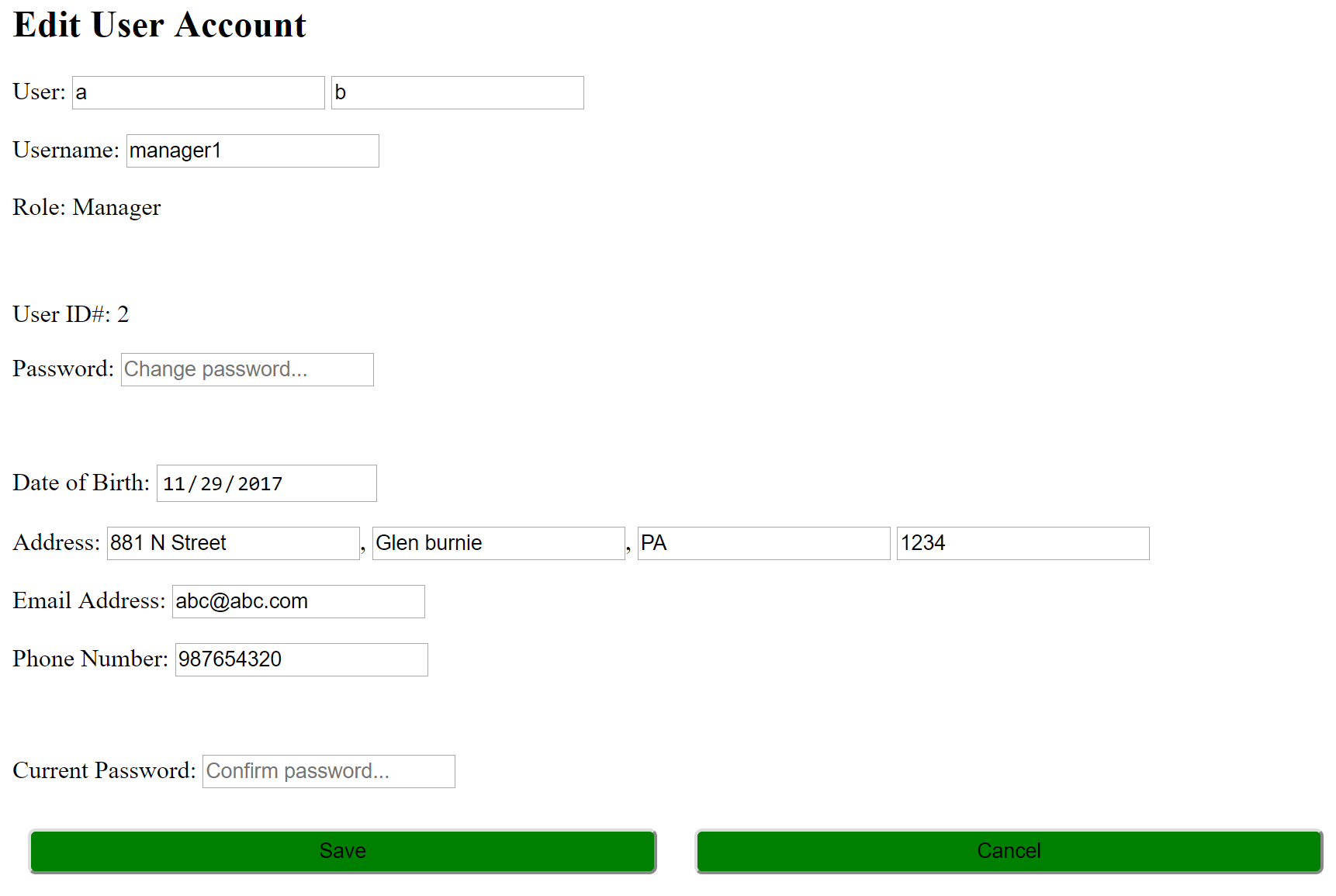
**Use Case: View Account**

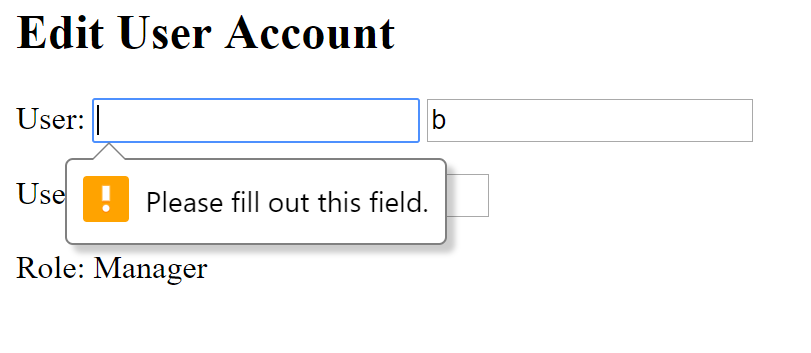
Conducted by: Aidan Gray

Result:  


**Use Case: Edit Account**

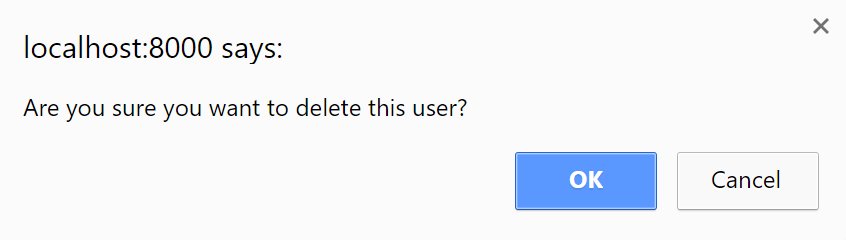
Conducted by: Aidan Gray

Valid Parameters:  


Invalid Parameters:  


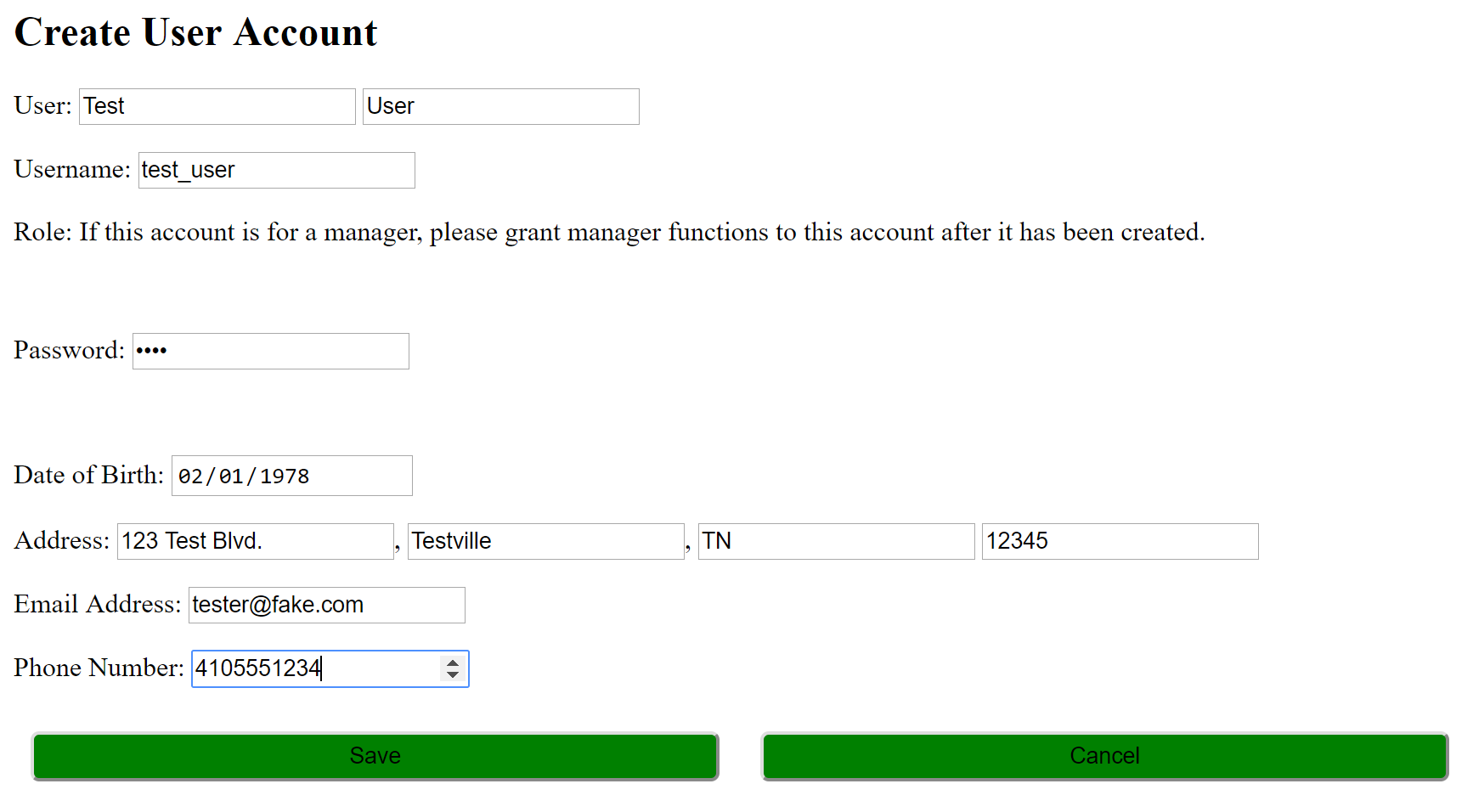
**Use Case: Delete User**

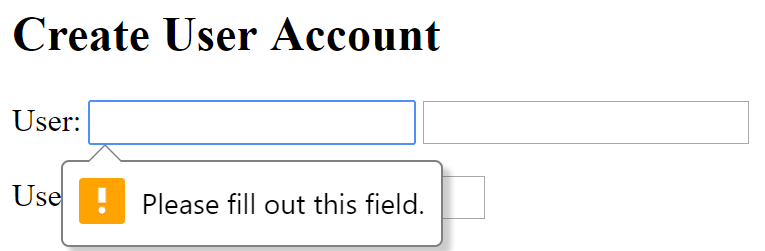
Conducted by: Aidan Gray

Result (Confirmation):  


**Use Case: Create User**

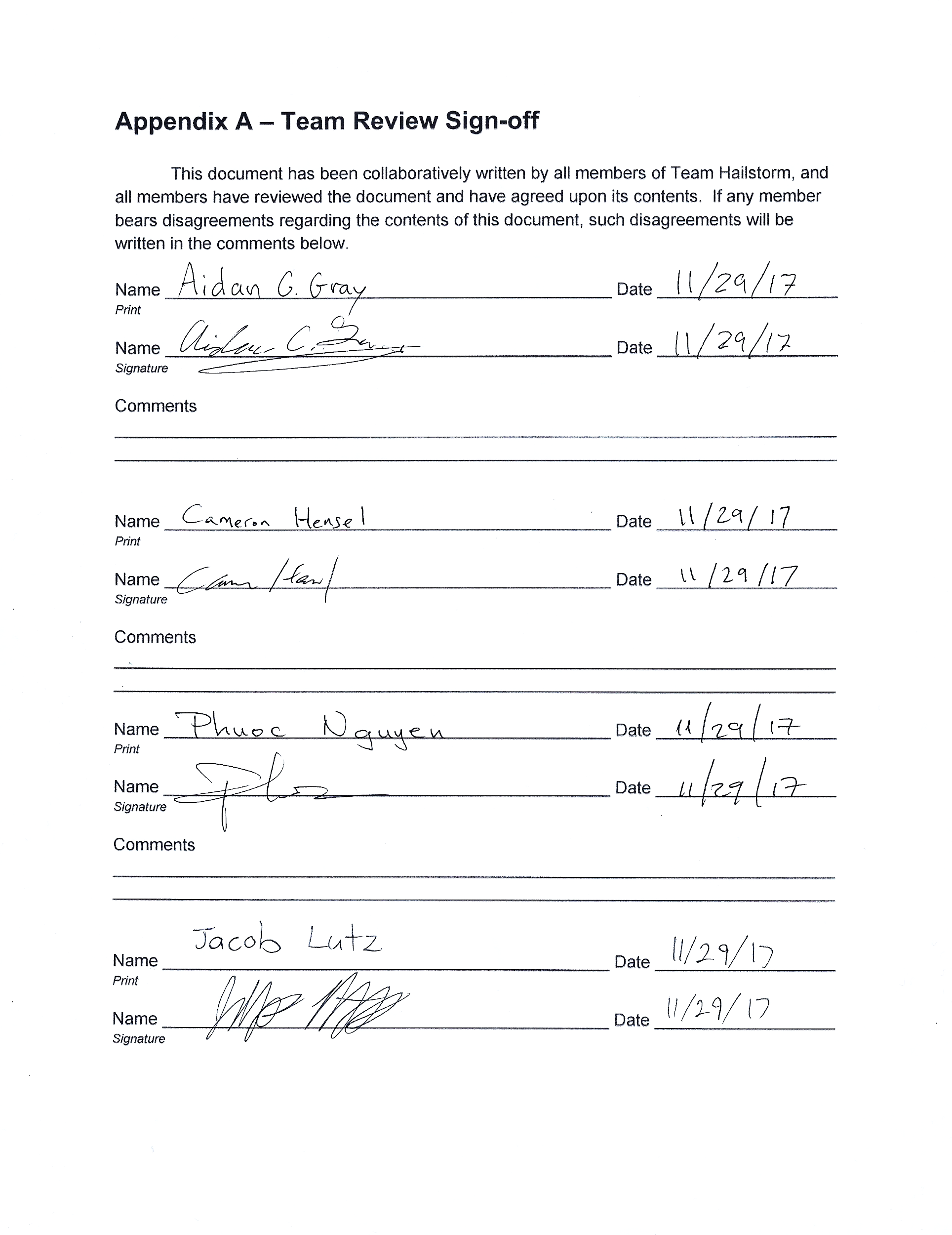
Conducted by: Aidan Gray

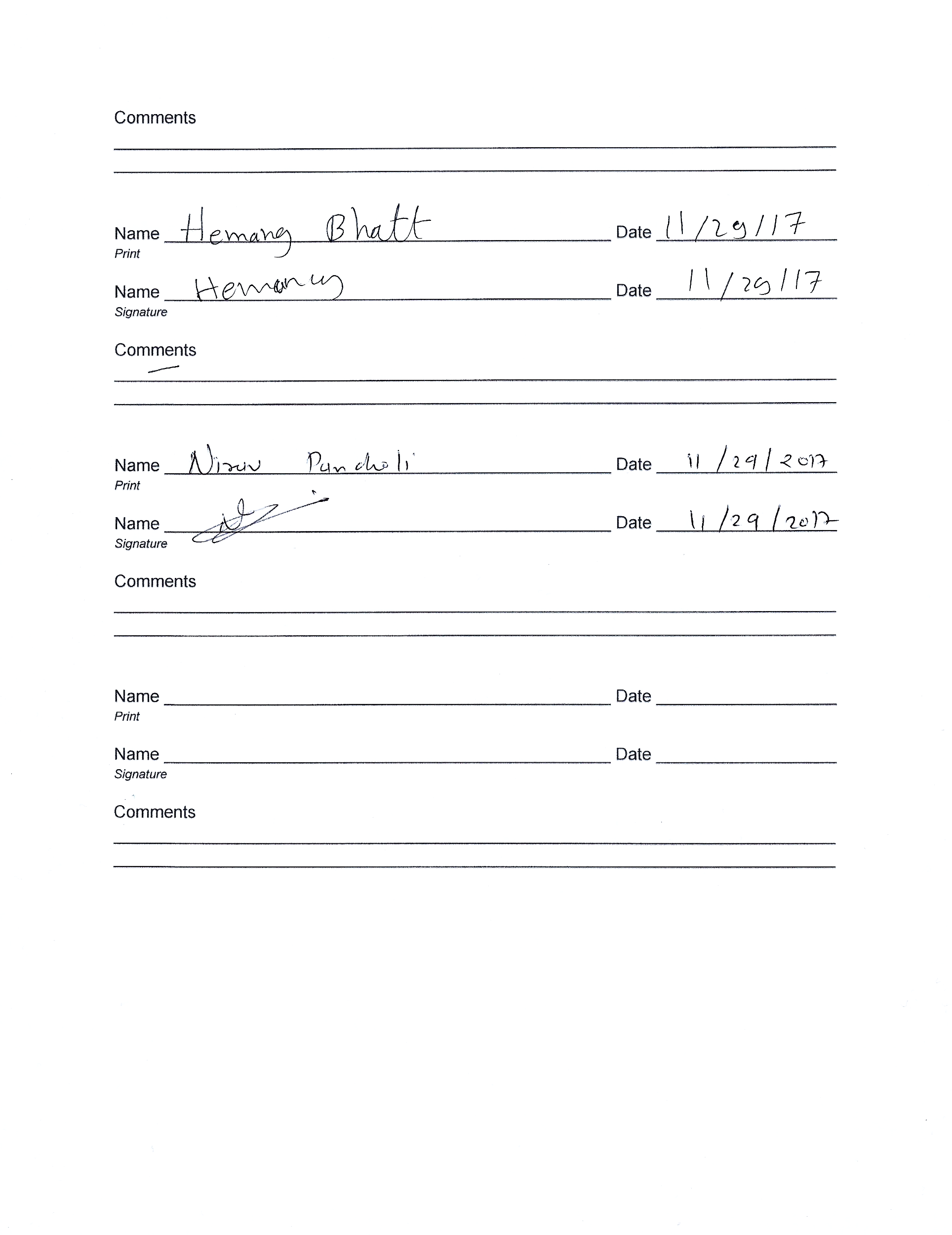
Valid Parameters:  


Invalid Parameters:  


**Appendix A – Team Review Sign-off**

This document has been collaboratively written by all members of Team Hailstorm, and all members have reviewed the document and have agreed upon its contents. If any member bears disagreements regarding the contents of this document, such disagreements will be written in the comments below.





**Appendix B – Document Contributions**

All team members contributed to the creation of the test case suite and review of the document. Hemang Bhatt, Jacob Lutz, Cameron Hansel, and Aidan Gray contributed to the written portion of the document, and Aidan Gray performed all testing.