A blue and white background

Description automatically generatedText Box 

Contents

[Executive Summary 5](#_Toc172569440)

[Functional Requirements Document 7](#_Toc172569441)

[1.Introduction 7](#_Toc172569442)

[1.1 Purpose 7](#_Toc172569443)

[1.2 Scope 7](#_Toc172569444)

[2. Functional Requirements 7](#_Toc172569445)

[2.1 User Authentication: 7](#_Toc172569446)

[2.2 Alumni Page: 7](#_Toc172569447)

[2.3 Address Information: 8](#_Toc172569448)

[2.4 Degree Information: 8](#_Toc172569449)

[2.5 Employment History: 8](#_Toc172569450)

[2.6 Donation Information: 8](#_Toc172569451)

[2.7 Skillset Information: 8](#_Toc172569452)

[2.8 Data Validation: 9](#_Toc172569453)

[2.9 Database Relationships: 9](#_Toc172569454)

[2.10 Navigation: 9](#_Toc172569455)

[2.11 User Roles and permissions 9](#_Toc172569456)

[Technical Requirements Document 10](#_Toc172569457)

[1. Overview 10](#_Toc172569458)

[2. Authentication and Authorization 10](#_Toc172569459)

[2.1 User Authentication: 10](#_Toc172569460)

[2.2 Session Management: 10](#_Toc172569461)

[2.3 User Permissions: 10](#_Toc172569462)

[3. Database Requirements 10](#_Toc172569463)

[3.1 Database System 10](#_Toc172569464)

[3.2 Tables and Relationships 11](#_Toc172569465)

[2.3 Entity Relationship 11](#_Toc172569466)

[2.4 Relationships: 12](#_Toc172569467)

[3. Files and File Types 12](#_Toc172569468)

[3.1 Project Structure 12](#_Toc172569469)

[3.2 File Types 13](#_Toc172569470)

[4. Connectivity 13](#_Toc172569471)

[4.1 Database Connectivity 13](#_Toc172569472)

[5.User Interface 13](#_Toc172569473)

[5.1 Navigation and UI Components: 13](#_Toc172569474)

[5.2 Alumni List and Search: 13](#_Toc172569475)

[6. Data Management 14](#_Toc172569476)

[6.1 CRUD Operations for Alumni: 14](#_Toc172569477)

[6.2 Address Management: 14](#_Toc172569478)

[6.3 Degree Management: 14](#_Toc172569479)

[6.4 Employment Management: 14](#_Toc172569480)

[6.5 Skill Management: 14](#_Toc172569481)

[7. Implementation 15](#_Toc172569482)

[7.1 Development Environment 15](#_Toc172569483)

[7.2 Security: 15](#_Toc172569484)

[7.3 Logging and Monitoring: 15](#_Toc172569485)

[7.4 Testing: 15](#_Toc172569486)

[7.5 Documentation: 16](#_Toc172569487)

[Technical Process Flow: 16](#_Toc172569488)

[Test plan 18](#_Toc172569489)

[Test Scripts 20](#_Toc172569490)

[Log In Page 20](#_Toc172569491)

[Alumni Page 21](#_Toc172569492)

[Address Information 23](#_Toc172569493)

[Degree 24](#_Toc172569494)

[Employment 26](#_Toc172569495)

[Donation 28](#_Toc172569496)

[Skill Set 29](#_Toc172569497)

[User Documentation and User Manual for the Alumni Management System 31](#_Toc172569498)

[User Documentation and User Manual Table of Contents 31](#_Toc172569499)

[1. Introduction: 33](#_Toc172569500)

[2.System Requirements: 33](#_Toc172569501)

[3. Installation and Setup: 33](#_Toc172569502)

[5.Navigating the Dashboard 34](#_Toc172569503)

[6. Managing Alumni 34](#_Toc172569504)

[Viewing Alumni List 34](#_Toc172569505)

[Adding a New Alumni 34](#_Toc172569506)

[Viewing Alumni Profile 34](#_Toc172569507)

[Editing Alumni Information 34](#_Toc172569508)

[Deleting Alumni 34](#_Toc172569509)

[7. Managing Addresses 34](#_Toc172569510)

[Viewing Addresses 34](#_Toc172569511)

[Adding a New Address 34](#_Toc172569512)

[Editing an Address 35](#_Toc172569513)

[8. Managing Employment 35](#_Toc172569514)

[9. Managing Degrees 35](#_Toc172569515)

[Deleting a Degree 36](#_Toc172569516)

[10. Managing Donations 36](#_Toc172569517)

[Viewing Donations 36](#_Toc172569518)

[Adding a Donation 36](#_Toc172569519)

[Editing a Donation 36](#_Toc172569520)

[Deleting a Donation 36](#_Toc172569521)

[Viewing Skillsets 36](#_Toc172569522)

[Editing a Skillset 37](#_Toc172569523)

[Deleting a Skillset 37](#_Toc172569524)

[Admin Privileges 37](#_Toc172569525)

[User Privileges 37](#_Toc172569526)

## Executive Summary

Alumni management system by Group 9 whose group members are Nigel Kaganzi, Kayla Gresham, Andrew Light, and Joshua Meyers, aimed to create a practical and easy to use website to keep track of alumni information. This system is designed to store and allow edits to a wide range of information including, degrees, addresses, job histories, donations, and skillsets. It connects that data to the correct alumni and supplies their name and email address to allow KSU to stay connected to them.

Our main goal was to make sure the data was accurate and secure. It is also important that the data is easily accessible and the website easy to navigate. The key features that allow this are the secure login which allows only approved individuals to edit data. The ability to add and delete alumni as well as edit existing alumni information which allows the data to stay up to date. Finally, a simple UI allows users to easily browse our website and find the data they need.

We used a mix of technologies to build this system. The backend uses flask which is a flexible python web framework, and MySQL is used to manage the database. The front end is built using CSS and HTML to give it that clean look as well as make it reactive.

Following the waterfall methodology, we took a step-by-step approach to this project with distinct times set aside for gathering requirements, designing and implementing those requirements, testing the product and finally deploying the product. Each phase was carefully planned out to meet the deadlines provided to us and give us time to make it as good as possible.

We developed a detailed testing plan to make sure the website worked on both the front end and back end. Each team member was assigned specific testing duties with a focus on testing parts they didn’t work on. We focused our testing to ensure functionality, Usability, and security as those were our main focuses during this project.

During all of this we regularly met to make sure everything was on track and working as intended. In conclusion by focusing on user experience, data accuracy, and security, holding ourselves to our schedule and keeping communication a priority, we have successfully delivered our product.

Functional Requirements Document

#### 1.Introduction

#### 1.1 Purpose

This document outlines the functional requirements for the Alumni Management System. The system is designed to manage and maintain alumni data.

#### 1.2 Scope

The system will allow authorized users to perform Create Read Update and Delete Operations on Alumni data such as their address, degree, employment history, donation and skills.

#### 2. Functional Requirements

#### 

##### 2.1 User Authentication:

* + Create a login page for user authentication.
  + Verify user credentials during login.
  + Create and manage session variables for tracking user permissions across pages (view, insert, update, delete).

##### 2.2 Alumni Page:

* + Display a list of alumni.
  + Allow the addition of new alumni.
  + Enable updating of existing alumni information.
  + Provide options to delete alumni records.
  + Direct users to pages or pop-up forms for entering related information (addresses, degrees, employment, donations, skills).

##### 2.3 Address Information:

* + Allow users to enter multiple addresses for each alumni.
  + Mark one address as the primary address.
  + Manage active flags to indicate valid addresses.
  + Maintain a history of old addresses without deleting them.

##### 2.4 Degree Information:

* + Track all degrees obtained by the alumni, including those before and after their association with KSU.
  + Allow entry of degree details such as degree type (bachelor’s, master’s, PhD), institution, and date obtained.

##### 2.5 Employment History:

* + Capture the employment history of alumni.
  + Create a new record for any job title changes within the same company.
  + Use a flag (currentYN) to indicate the alumni’s current employment position.

##### 2.6 Donation Information:

* + Record donations made by alumni.
  + Maintain a history of all donations, including amount and date of donation.

##### 2.7 Skillset Information:

* + Allow freeform entry of alumni skills.
  + Track various specialized skills like programming, database management, project management, etc.

##### 2.8 Data Validation:

* + Implement validation to limit typing mistakes and ensure data integrity.

##### 2.9 Database Relationships:

* + Ensure all data entry and updates respect the database relationships and constraints.

##### 2.10 Navigation:

* + Design a user-friendly interface to navigate between different pages and pop-up forms for data entry.
  + Ensure seamless flow between viewing, adding, updating, and deleting information.

##### 2.11 User Roles and permissions

* + Support user roles with different permissions, such as admin and regular user roles.

A diagram of a process flow

Description automatically generated

Technical Requirements Document

#### 1. Overview

This document outlines the technical requirements needed to develop the Alumni Management System. It includes detailed information on the systems database, files conectivity, architecure and implementation methods

#### 2. Authentication and Authorization

##### 2.1 User Authentication:

* + Implement a login form to collect username and password.
  + Flask-Login for user session management
  + Store passwords securely using hashing algorithms like bcrypt.

##### 2.2 Session Management:

* + Implement session management to track logged-in users.
  + Use cookies or tokens to manage user sessions.
  + Implement session timeout (e.g., 30 minutes of inactivity) to enhance security.

#### 2.3 User Permissions:

* + Define user roles (e.g., admin, editor, viewer) and permissions for each role.
  + Implement role-based access control (RBAC) to enforce permissions on different actions (view, edit, delete).

#### 3. Database Requirements

##### 3.1 Database System

* Database Management System: MySQL.
* Database Name: ‘capstoneproject’.

##### 3.2 Tables and Relationships

* Alumni: Store Basic Alumni Information.
* Address: Stores Multiple addresses for each Alumni with active and primary markers.
* Degree: Tracks degrees achieved by Alumni.
* Employment: Tracks employment history with current marker.
* Donation: Logs donations made by Alumni.
* Skillset: List of skills that Alumni have.

##### 3.3 Entity Relationship

Alumni:

- alumniID (PK), fName, lName, phone, email, DOB, gender, ethnicity, website, linkedIn\_link, twitter\_link, facebook\_link, instagram\_link, guestSpeakerYN, newsLetterYN, imageThumb, imageNormal, description, deceasedYN, deceasedDT, deceasedNotes

Address:

- addressID (PK), alumniID (FK), address, city, state, zipCode, activeYN, primaryYN

User:

- userID (PK), password, fName, lName, jobDescription, viewPriveledgeYN, insertPriveledgeYN, updatePriveledgeYN, deletePriveledgeYN

Employment:

- employmentID (PK), alumniID (FK), company, city, state, zip, jobTitle, startDate, endDate, currentYN, notes

Degree:

- degreeID (PK), alumniID (FK), major, minor, graduationDT, university, city, state

Skillset:

- skillsetID (PK), alumniID (FK), skill, proficiency, description

Donation:

- donationID (PK), alumniID (FK), donationAmt, donationDT, reason, description

##### 3.4 Relationships:

* Alumni to Address: One-to-Many (One Alumni can have multiple addresses)
* Alumni to Employment: One-to-Many (One Alumni can have multiple employments)
* Alumni to Degree: One-to-Many (One Alumni can have multiple degrees)
* Alumni to Skillset: One-to-Many (One Alumni can have multiple skills)
* Alumni to Donations: One-to-Many (One Alumni can make multiple donations)

##### 3.5 Database Entity Relationship Diagram

A diagram of a computer

Description automatically generated with medium confidence

#### 4. Files and File Types

##### 4.1 Project Structure

* **Main Project Directory**:
  + app.py: Main application file.
  + config.py: Configuration settings.
  + requirements.txt: Project dependencies.
  + run.py: Run script for the application.
* **App Directory (app)**:
  + \_\_init\_\_.py: Initialize the Flask app and database.
  + models.py: Database models.
  + views.py: Application routes.
  + forms.py: Forms for data input.
  + templates/: HTML templates.
  + static/: Static files (CSS, JS, images).

##### 4.2 File Types

* **Python Files (.py)**: Used for views, models, forms, and configuration.
* **HTML Files (.html)**: Used for templates.
* **CSS Files (.css)**: Used for styling.
* **JavaScript Files (.js)**: Used for client-side interactivity.

#### 5. Connectivity

##### 5.1 Database Connectivity

* Database Engine: ‘SQLAlchemy’ with MySQL
* Database Client: ‘pymysql’

Database connection configuration located in ‘config.py’ file.

#### 6.User Interface

##### 6.1 Navigation and UI Components:

* Create a responsive user interface with navigation links or buttons to different sections (home, alumni list, addresses, degrees, employment, donation, skillset).
* Use Adobe Dreamweaver for building the UI components.

##### 6.2 Alumni List and Search:

* Implement a view to display the list of alumni with basic information.
* Implement search and filter functionality for the alumni list using query parameters.
* Use pagination to handle large datasets.

#### 7. Data Management

##### 7.1 CRUD Operations for Alumni:

* Implement API endpoints to create, read, update, and delete alumni records.
* Use a relational database (MySQL) to store alumni data.
* Implement data validation for all CRUD operations to ensure data integrity.

##### 7.2 Address Management:

* Implement functionality to add multiple addresses for each alumnus.
* Allow marking one address as the primary address.
* Implement status flags (active, inactive) for addresses and prevent deletion.

##### 7.3 Degree Management:

* Implement functionality to add, view, and update multiple degrees for each alumnus.
* Store degree details (type, institution, graduation year) in the database.

##### 7.4 Employment Management:

* Implement functionality to add, view, and update multiple job records for each alumnus.
* Store job details (company name, job title, start and end dates) in the database.
* Track current employment status and handle changes within the same company.

##### 7.5 Skill Management:

* Implement functionality to add, view, and update skills for each alumnus.
* Store skills as text or use a taxonomy for predefined skill categories.
* Implement search functionality to find alumni with specific skills.

#### 8. Implementation

##### 8.1 Development Environment

* **Operating System**: Cross-platform (Windows, macOS, Linux)
* **Programming Language**: Python
* **Framework**: Flask
* **Database**: MySQL
* **Version Control**: Git

##### 8.2 Security:

* **Authentication**: Flask-Login
* **Authorization**: Role Based access control
* Implement HTTPS to encrypt data in transit.
* Implement input validation and sanitation to prevent SQL injection and XSS attacks.
* Regularly update dependencies to mitigate security vulnerabilities.

##### 8.3 Logging and Monitoring:

* Implement logging for key actions (e.g., login, CRUD operations) for audit purposes.
* Use monitoring tools to track application performance and detect anomalies.

##### 8.4 Testing:

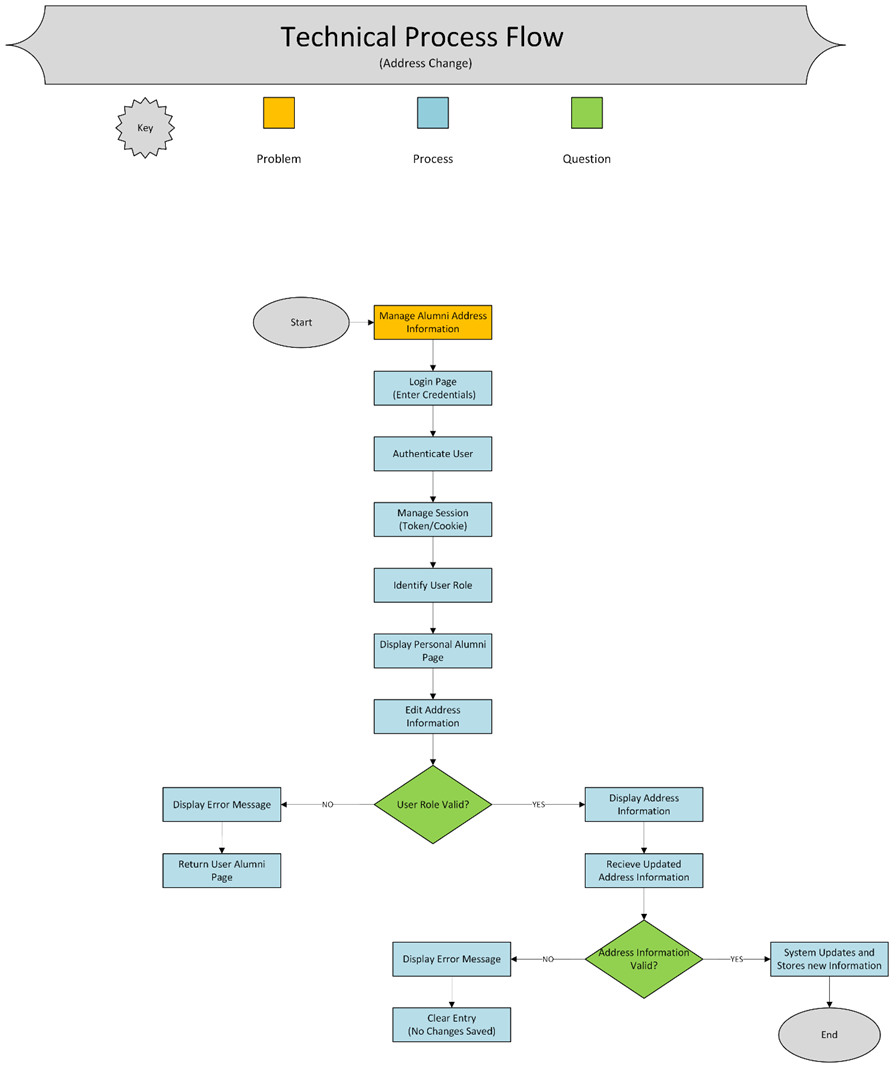
* Implement unit tests for individual components and functions.
* Implement integration tests to verify the interaction between different parts of the application.

#### 8.5 Documentation:

* Provide user manuals and technical documentation for developers.

#### Technical Process Flow:

“This flow shows an example of how our system would operate if a user wanted to change their Address information.”



## Test plan

* Executive summary:

This testing plan involves frontend testers Kayla Gresham and Andrew Light, and backend testers Nigel Kaganzi and Josh Meyers. Using the waterfall methodology, we will perform functional and usability testing on the login page, alumni page, address information, degree, employment, donations, and skillset, excluding front end reporting, newsletters, and reports. Testing runs from July 4 to July 16, 2024, with specific days for each component and time for bug fixes. The main risk is the rigidity of the waterfall method, mitigated by constant communication to promptly address issues and adjust the schedule.

* Testing resources:

Frontend testers: Kayla Gresham, Andrew Light.

Backend testers: Nigel Kaganzi, Josh Meyers.

* Scope of testing

In Scope: Login page, Alumni page, Address information, Degree, Employment, Donations, Skillset

Out of Scope: Front end reporting, Newsletter, Reports

* Testing Approach

Methodology: The goals of this testing plan are to ensure that the application works as intended and that the user experience is easy and enjoyable. The best testing approach to meet these goals is the waterfall method. As we are all very new at dealing with projects of this scale a strict plan that we all have to stick to will keep us on track and on schedule. The types of testing we will be doing during this test plan are functional and usability.

* Test schedule

7/4/24: test front end login page

7/5/24: test back end system for login page

7/6/24: test front end alumni page

7/7/24: test back end system for alumni page

7/8 through 7/11: fixing bugs and retesting

7/12/24: test back end address information

7/13/24: test back end degree

7/14/24: test back end donation

7/15/24: test back end employment

7/16/24: test back end skill set

7/17 through 7/21: bug fixes and retesting

* Risks

The risks associated with this testing plan is how rigid the system is. The way to mitigate this by staying in near constant communication whenever testing is being done and when bug fixing is being done. This will give the team plenty of time to address issues before the affect the schedule or to make changes to the schedule if it is needed.

## Test Scripts

#### Log In Page

|  |  |  |  |
| --- | --- | --- | --- |
| Descriptions | Expected result | Actual Result | Comments/ Pass Fail |
| Successful Log in  1. Enter correct username into login page.  2. Enter correct password into login page.  3. Click the login button. | Successfully logged in and directs user to alumni page | User is logged in and is directed to the home page | Pass |
| Incorrect Username  1. Enter incorrect username into login page.  2. Enter correct password into login page.  3. Click the login button. | Site says “incorrect user name or password” and keeps user on log in page | Message Flashed Incorrect username or password. User is kept on the login page | Pass |
| Incorrect Password  1. Enter correct username into login page.  2. Enter incorrect password into login page.  3. Click the login button. | Site says “incorrect user name or password” and keeps user on log in page | Message Flashed  Incorrect username or password.  User is kept on the login page | Pass |
| Incorrect Username/Password  1. Enter incorrect username into login page.  2. Enter incorrect password into login page.  3. Click the login button. | Site says “incorrect user name or password” and keeps user on log in page | Message Flashed  Incorrect username or Password.  User is kept on the login Page | Pass |

#### Alumni Page

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Expected Result | Actual Result | Comments/ Pass Fail |
| Verify Alumni Page Load  1. Navigate to the alumni page.  2. Ensure the page loads completely without errors. | Alumni page loads successfully with all UI elements visible and functional. | Alumni Page Loads and is Visible | Pass |
| Add New Alumni Entry  1. Navigate to the alumni page.  2. Click on the "Add New Alumni" button.  3. Fill in the required fields (e.g., name, graduation year, degree).  4. Submit the form. | New alumni entry is added and visible in the alumni list. | New Alumni Entry is visible | Pass |
| Edit Existing Alumni Entry  1. Navigate to the alumni page.  2. Select an existing alumni entry from the list.  3. Click on the "Edit" button.  4. Modify the necessary fields (e.g., update contact information).  5. Submit the form. | Alumni entry is updated and the changes are reflected in the alumni list. | Alumni entry updated and changes are reflected in the alumni List | Pass |
| Delete Alumni Entry  1. Navigate to the alumni page.  2. Select an existing alumni entry from the list.  3. Click on the "Delete" button.  4. Confirm the deletion. | Alumni entry is removed from the alumni list. | Alumni is removed from the list | Pass |
| Navigate to Detailed Information Pages  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on a link/button to navigate to a detailed information page (e.g., employment history, donations). | User is redirected to the detailed information page and the correct alumni data is displayed. | User is navigated to Alumni Profile page | Pass |
| Search Alumni  1. Navigate to the alumni page.  2. Use the search functionality to look for a specific alumni (e.g., by name or graduation year). | Alumni matching the search criteria are displayed in the results list. | Alumni with matching criteria are found for name degree employment and skill | Pass |
| Data Validation and Error Handling  1. Navigate to the alumni page.  2. Try to submit the form with invalid data (e.g., missing required fields, incorrect data format). | Appropriate error messages are displayed, and form submission is prevented until the errors are corrected. | Fields with Data Validators flash appropriate validator messages | Pass |
| Verify Database Relationships  1. Navigate to the alumni page  2. Add or update alumni information that affects related tables (e.g., adding a new degree).  3. Check the related tables in the database. | Data is correctly updated in the related tables according to the database relationships. | Data is updated accordingly | Pass |
| User Permissions and Access Control  1. Log in as a user with different permission levels (e.g., admin, regular user).  2. Navigate to the alumni page and attempt to perform various actions (e.g., add, edit, delete). | Actions available to the user match their permission level, preventing unauthorized access. | User is not allowed to update or delete Alumni Data | Pass |

#### Address Information

|  |  |  |  |
| --- | --- | --- | --- |
| Descriptions | Expected Result | Actual Result | Comments/ Pass Fail |
| Add Multiple Addresses  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Address" button.  4. Enter a home address and mark it as primary and active.  5. Submit the form.  6. Add a work address and mark it as active, but not primary. | Both addresses are added to the alumni entry, with the home address marked as primary and both addresses marked as active. | Both Addresses are added and ordered by primary address, and primary address is blue | Pass |
| Set Primary Address  1. Navigate to the alumni page.  2. Select an alumni entry with multiple addresses.  3. Edit the work address and mark it as primary.  4. Submit the form. | The work address is now marked as the primary address, and the home address is no longer primary. | Both Addresses remain as primary | Fail |
| Deactivate Old Address  1. Navigate to the alumni page.  2. Select an alumni entry with an existing home address marked as primary.  3. Add a new home address and mark it as primary and active.  4. Submit the form. | The new home address is added and marked as primary and active. The old home address is marked as not active but remains in the record. | A new address is added and both addresses are marked as primary | Fail |
| Ensure Only One Primary Address  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Attempt to mark both home and work addresses as primary. | The system prevents setting both addresses as primary, ensuring only one address is primary at a time. | Both are marked as primary address | Pass |
| Validation of Address Fields  1. Navigate to the alumni page.  2. Add a new address with invalid data (e.g., missing required fields, incorrect format). | The system displays appropriate error messages and prevents the form submission until the data is corrected. | Address Form does not submit without valid data, and flashes messages to validate Data | Pass |

#### Degree

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Expected Result | Actual Result | Comments/ Pass Fail |
| Add Degree Information  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Degree" button.  4. Enter details for a degree obtained at KSU (e.g., bachelor's degree).  5. Submit the form. | The degree information is added and visible in the alumni entry. | Degree information is visible and new degree is added | Pass |
| Add Multiple Degrees  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Degree" button.  4. Enter details for a degree obtained prior to KSU (e.g., associate degree).  5. Submit the form.  6. Repeat to add another degree obtained after leaving KSU (e.g., master's degree). | All degrees are added to the alumni entry, with each degree correctly displayed. | Degree information is visible and multiple entries are available | Pass |
| Edit Degree Information  1. Navigate to the alumni page.  2. Select an alumni entry with existing degree information.  3. Click on the "Edit" button next to a degree.  4. Modify the degree details (e.g., change the degree title or institution).  5. Submit the form. | The degree information is updated successfully and the changes are reflected in the alumni entry. | Degree information is updated with modified information visible | Pass |
| Delete Degree Information  1. Navigate to the alumni page.  2. Select an alumni entry with existing degree information.  3. Click on the "Delete" button next to a degree.  4. Confirm the deletion. | The degree information is removed from the alumni entry. | Degree information for selected entry is deleted | Pass |
| Verify Degree History  1. Navigate to the alumni page.  2. Select an alumni entry with multiple degrees.  3. Ensure all added degrees are listed correctly in chronological order. | All degrees are displayed in the correct order, showing comprehensive degree history. | Degrees are displayed in correct order | Pass |
| Validation of Degree Fields  1. Navigate to the alumni page.  2. Add a new degree with invalid data (e.g., missing required fields, incorrect format). | The system displays appropriate error messages and prevents the form submission until the data is corrected. | Degree Form does not submit without valid data, and flashes messages to validate data | Pass |

#### Employment

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Expected Result | Actual Result | Comments/ Pass Fail |
| Add Employment Record  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Employment" button.  4. Enter details for an employment record (e.g., company name, job title, start date).  5. Submit the form. | The employment information is added and visible in the alumni entry. | Employment information is visible. | Pass |
| Add Multiple Employment Records  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Employment" button.  4. Enter details for an initial employment record.  5. Submit the form.  6. Add another employment record for a different company. | All employment records are added to the alumni entry, showing the entire employment history. | Employment information is visible. Multiple entries are added | Pass |
| Edit Employment Record  1. Navigate to the alumni page.  2. Select an alumni entry with existing employment information.  3. Click on the "Edit" button next to an employment record.  4. Modify the employment details (e.g., update job title or company name).  5. Submit the form. | The employment information is updated successfully and the changes are reflected in the alumni entry. | Employment information is updated and new data is displayed. | Pass |
| Delete Employment Record  1. Navigate to the alumni page.  2. Select an alumni entry with existing employment information.  3. Click on the "Delete" button next to an employment record.  4. Confirm the deletion. | The employment record is removed from the alumni entry. | Selected employment entry is deleted and no longer available on the page | Pass |
| Verify Employment History  1. Navigate to the alumni page.  2. Select an alumni entry with multiple employment records.  3. Ensure all added employment records are listed correctly in chronological order. | All employment records are displayed in the correct order, showing comprehensive employment history. | Employment records are displayed in correct order | Pass |
| Validation of Employment Fields  1. Navigate to the alumni page.  2. Add a new employment record with invalid data (e.g., missing required fields, incorrect format). | The system displays appropriate error messages and prevents the form submission until the data is corrected. | Employment Form does not submit without valid data, and flashes messages to validate data | Pass |

#### Donation

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Expected Result | Actual Result | Comments/ Pass Fail |
| Add Donation Record  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Donation" button.  4. Enter details for a donation (e.g., amount, date, purpose).  5. Submit the form. | The donation information is added and visible in the alumni entry. | Donation information is visible | Pass |
| Add Multiple Donation Records  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Donation" button.  4. Enter details for an initial donation.  5. Submit the form.  6. Add another donation record with different details. | All donation records are added to the alumni entry, showing the entire donation history. | All donation entries are visible | Pass |
| Edit Donation Record  1. Navigate to the alumni page.  2. Select an alumni entry with existing donation information.  3. Click on the "Edit" button next to a donation record.  4. Modify the donation details (e.g., change the amount or purpose).  5. Submit the form. | The donation information is updated successfully, and the changes are reflected in the alumni entry. | Updated donation information is reflected in the donation page | Pass |
| Delete Donation Record  1. Navigate to the alumni page.  2. Select an alumni entry with existing donation information.  3. Click on the "Delete" button next to a donation record.  4. Confirm the deletion. | The donation record is removed from the alumni entry. | Selected donation entry is deleted and no longer available | Pass |
| Verify Donation History  1. Navigate to the alumni page.  2. Select an alumni entry with multiple donation records.  3. Ensure all added donation records are listed correctly in chronological order. | All donation records are displayed in the correct order, showing comprehensive donation history. | Donation entries are displayed in the correct order | Pass |
| Validation of Donation Fields  1. Navigate to the alumni page.  2. Add a new donation record with invalid data (e.g., missing required fields, incorrect format). | The system displays appropriate error messages and prevents the form submission until the data is corrected. | Donation Form does not submit without valid data, and flashes messages to validate data | Pass |

#### Skill Set

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Expected Result | Actual Result | Comments/ Pass Fail |
| Add Skillset Entry  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Skill" button.  4. Enter details for a skill (e.g., "Programming").  5. Submit the form. | The skillset information is added and visible in the alumni entry. | New skillset entry is added | Pass |
| Add Multiple Skillset Entries  1. Navigate to the alumni page.  2. Select an alumni entry.  3. Click on the "Add Skill" button.  4. Enter details for an initial skill (e.g., "Database Management").  5. Submit the form.  6. Add another skillset record with different details (e.g., "Project Management"). | All skillset records are added to the alumni entry, showing the complete set of skills. | Multiple entries are visible | Pass |
| Edit Skillset Entry  1. Navigate to the alumni page.  2. Select an alumni entry with existing skillset information.  3. Click on the "Edit" button next to a skill.  4. Modify the skill details (e.g., change "Programming" to "Advanced Programming").  5. Submit the form. | The skillset information is updated successfully and the changes are reflected in the alumni entry. | Skillset information is updated and new information is edited | Pass |
| Delete Skillset Entry  1. Navigate to the alumni page.  2. Select an alumni entry with existing skillset information.  3. Click on the "Delete" button next to a skill.  4. Confirm the deletion. | The skillset record is removed from the alumni entry. | Selected entry is deleted and no longer available | Pass |
| Verify Skillset Listing  1. Navigate to the alumni page.  2. Select an alumni entry with multiple skillset records.  3. Ensure all added skillset records are listed correctly. | All skillset records are displayed correctly, showing the complete skillset of the alumni. | All records displayed correctly | Pass |
| Validation of Skillset Fields  1. Navigate to the alumni page.  2. Add a new skillset record with invalid data (e.g., empty field). | The system displays appropriate error messages and prevents the form submission until the data is corrected. | Donation Form does not submit without valid data, and flashes messages to validate data | Pass |

## User Documentation and User Manual for the Alumni Management System

#### User Documentation and User Manual Table of Contents

1. Introduction
2. System Requirements
3. Installation and Setup
4. Loggin In
5. Navigating the Dashboard
6. Managing Alumni
7. Viewing Alumni List
8. Adding a new Alumni
9. Viewing Alumni Profile
10. Editing Alumni Information
11. Deleting Alumni
12. Managing Addresses
13. Viewing Addresses
14. Adding a new Address
15. Editing an Address
16. Deleting an Address
17. Managing Employment
18. Viewing Employment History
19. Adding Employment Record
20. Editing Employment Record
21. Deleting Employment Record
22. Managing Degrees
23. Viewing Degrees
24. Adding a new Degree
25. Editing a Degree
26. Deleting an Address
27. Managing Donations
28. Viewing Donations
29. Adding a Donation
30. Editing a Donation
31. Deleting a Donation
32. Managing Skillset
33. Viewing Skillset
34. Adding a Skillset
35. Editing a skillset
36. Deleting a Skillset
37. Role-Based Access Control
38. Admin Privileges
39. User Privileges
40. Logging Out
41. Troubleshooting

#### 1. Introduction:

This Alumni Management System is designed to help the Information Systems & Security department efficiently manage and maintain alumni data. This system includes features for handling alumni information, address updates, degree records, employment history, donations, and skillsets. This document serves as a comprehensive guide for users to understand and effectively utilize the system.

#### 2.System Requirements:

* **Operating System:** Any OS that supports a web browser
* **Web Browser:** Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge
* **Backend:** Python with Flask framework
* **Database:** MySQL
* **Hosting:** PythonAnywhere (or any compatible web server)
* **Additional Libraries:** Listed in requirements.txt

#### 3. Installation and Setup:

**Local Setup**

Clone the Repository

git clone https://github.com/your-repo-url/IS-Capstone-Group-9.git cd IS-Capstone-Group-9

 Install Dependencies

pip install -r requirements.txt

Setup database

In MySQL Workbench, select server> Data import>Import> choose cp\_db>start import

Set environment variables in config.py

SECRET\_KEY=your\_secret\_key DATABASE\_URL=mysql+mysqlconnector://user:password@host/dbname

Run the application

Flask run

**4. Logging In**

1. **Navigate to the Login Page**.
2. **Enter your User ID and Password**.
3. **Click the Login Button**

#### 5.Navigating the Dashboard

After logging in, you will be directed to the dashboard where you can see a welcome message along with your role privileges.

#### 6. Managing Alumni

##### Viewing Alumni List

1. **Click on 'Alumni List'** from the navigation bar.
2. **Search for Alumni** by name or email using the search bar.

##### Adding a New Alumni

1. **Click on 'Add Alumni'** button.
2. **Fill in the Alumni Details** and click Submit.

##### Viewing Alumni Profile

1. **Click on an Alumni's Name** from the alumni list.
2. **View detailed information** about the alumni.

##### Editing Alumni Information

1. **Navigate to the Alumni's Profile**.
2. **Click on 'Edit'**.
3. **Update the necessary information** and click Submit.

##### Deleting Alumni

1. **Navigate to the Alumni's Profile**.
2. **Click on 'Delete'**.
3. **Confirm the deletion**.

#### 7. Managing Addresses

##### Viewing Addresses

1. **Navigate to the Alumni's Profile**.
2. **Click on 'View Addresses'**.

##### Adding a New Address

1. **In the Addresses page**, click on 'Add Address'.
2. **Fill in the Address Details** and click Submit.

##### Editing an Address

1. **In the Addresses page**, click on 'Edit' next to the address.
2. **Update the Address Details** and click Submit.

**Deleting an Address**

1. **In the Addresses page**, click on 'Delete' next to the address.
2. **Confirm the deletion**.

#### 8. Managing Employment

**Viewing Employment History**

1. **Navigate to the Alumni's Profile**.
2. **Click on 'View Employment'**.

**Adding Employment Record**

1. **In the Employment page**, click on 'Add Employment'.
2. **Fill in the Employment Details** and click Submit.

**Editing Employment Record**

1. **In the Employment page**, click on 'Edit' next to the employment record.
2. **Update the Employment Details** and click Submit.

**Deleting Employment Record**

1. **In the Employment page**, click on 'Delete' next to the employment record.
2. **Confirm the deletion**.

#### 9. Managing Degrees

**Viewing Degrees**

1. **Navigate to the Alumni's Profile**.
2. **Click on 'View Degrees'**.

**Adding a New Degree**

1. **In the Degrees page**, click on 'Add Degree'.
2. **Fill in the Degree Details** and click Submit.

**Editing a Degree**

1. **In the Degrees page**, click on 'Edit' next to the degree.
2. **Update the Degree Details** and click Submit.

##### Deleting a Degree

1. **In the Degrees page**, click on 'Delete' next to the degree.
2. **Confirm the deletion**.

#### 10. Managing Donations

##### Viewing Donations

1. **Navigate to the Alumni's Profile**.
2. **Click on 'View Donations'**.

##### Adding a Donation

1. **In the Donations page**, click on 'Add Donation'.
2. **Fill in the Donation Details** and click Submit.

##### Editing a Donation

1. **In the Donations page**, click on 'Edit' next to the donation.
2. **Update the Donation Details** and click Submit.

##### Deleting a Donation

1. **In the Donations page**, click on 'Delete' next to the donation.
2. **Confirm the deletion**.

**11. Managing Skillsets**

##### Viewing Skillsets

1. **Navigate to the Alumni's Profile**.
2. **Click on 'View Skillsets'**.

**Adding a Skillset**

1. **In the Skillsets page**, click on 'Add Skillset'.
2. **Fill in the Skillset Details** and click Submit.

##### Editing a Skillset

1. **In the Skillsets page**, click on 'Edit' next to the skillset.
2. **Update the Skillset Details** and click Submit.

##### Deleting a Skillset

1. **In the Skillsets page**, click on 'Delete' next to the skillset.
2. **Confirm the deletion**.

**12. Role-Based Access Control**

##### Admin Privileges

* **Admins** can view, add, edit, and delete all records.
* **Admins** can manage user roles and permissions.

##### User Privileges

* **Users** have limited access based on their role settings.
* **Users** can typically view and edit their own profile.

**13. Logging Out**

1. **Click on 'Logout'** in the navigation bar to log out of the system.

**14. Troubleshooting**

##### Common Issues

* **Cannot log in**: Ensure your User ID and Password are correct.
* **Cannot add/edit/delete records**: Ensure you have the necessary permissions.

##### Getting Help

* **Check the Documentation**: Refer to this user manual for guidance.
* **Contact Support**: Reach out to the IS Capstone Group 9 Team

**15. Contact Support**

For further assistance, please contact the IT support team at Kennesaw State University:

* **Email**: support@kennesaw.edu
* **Phone**: (470) 578-6999