Requirements Specification

Abstract

The purpose of this system is to create and implement a functional incident management system for the customer, Fox Valley Special Recreation Association. Their current system is utilized with paper forms and needs to be replaced with a web application for all employees and designated admins to use.

1. Introduction

A. The overall goal of this project is to create a functioning Incident Management System for the Fox Valley Special Recreation Association. This project is being continued from last semester, meaning the project is starting off half completed. The main styling of the front end is almost fully complete. Two of the five forms are fully functional in terms of submitting the data to the database. The remaining forms that are not functional need variable changes on the front end, and the backend to be created to post them to the database. The next major functionality would be the exporting of the form data from the database to an Excel file. Another aspect that needs to be corrected is the form submission page. We had perceived this functionality to be completed from our previous work, however, this turned out to be a problem. The submission page usually does not come up after successful form submission, which needs to be corrected. This form submission page is useful because it gives a clean visual indication of the status of the form they attempted to submit.

Another main aspect of the program that needs to be fixed is the login function. Currently the login page does not have a functioning back end. The user has an option of clicking the login button or the admin button to reach their perspective pages. This needs to be corrected so there are two buttons, one for employees and one for admins. The employee button will link directly to the form creation page, which they have access to four of the five forms, excluding form 04. The admin button will link to a login page where their credentials will need to be entered and verified by the backend comparing to the database. The admin login will be preset by us and changed before deployment to meet the customer's needs. After credential verification, it will link to the admin page where all of the admin functions will be available. These functions include form creation, form viewing, and security. The form creation function for the admin is the same as the

employee, with the addition of form 04. The form viewing function will allow the admin to download the current form data in the database to an excel file. The security function will allow the admin to create users(giving them a username, and password), change the main admin password, change any user's password, and delete any user(this feature will need to have a layer of protection via a master password that is separate from the admin password).

B. The people that will use this system fall into two main categories. The first being all FVSRA employees. The employees are able to login to the system easily and create a form for the designated incident that occurred. The second category is the admins. The admins are able to view any created forms as well as create forms.

C. The figure below is the workflow diagram that describes how the customer will use the system. The user will start on the initial page. This page gives the user two options in the form of buttons. The admin button and employee button. If the user clicks the employee button they will be redirected to the form selection page.

On the form selection page the employee will have access to 4 of the 5 forms, excluding form 04. The user would be able to click the button correlating to the form they wish to fill out. After filling out the form they would click the submit button. Upon clicking this button it will either inform the user of input errors, or if errorless, will redirect to the status page. The status page will have a confirmation message of the form submission. There will be a button to link back to the home page if they wish to submit another form.

If the user selects the admin button, they would be redirected to the login page. On this page the user would enter their username and password, and then click submit. If the username and password are incorrect, a message would display informing the user and they would be prompted to reenter. If the username and password are correct, the user will be redirected to the admin page.

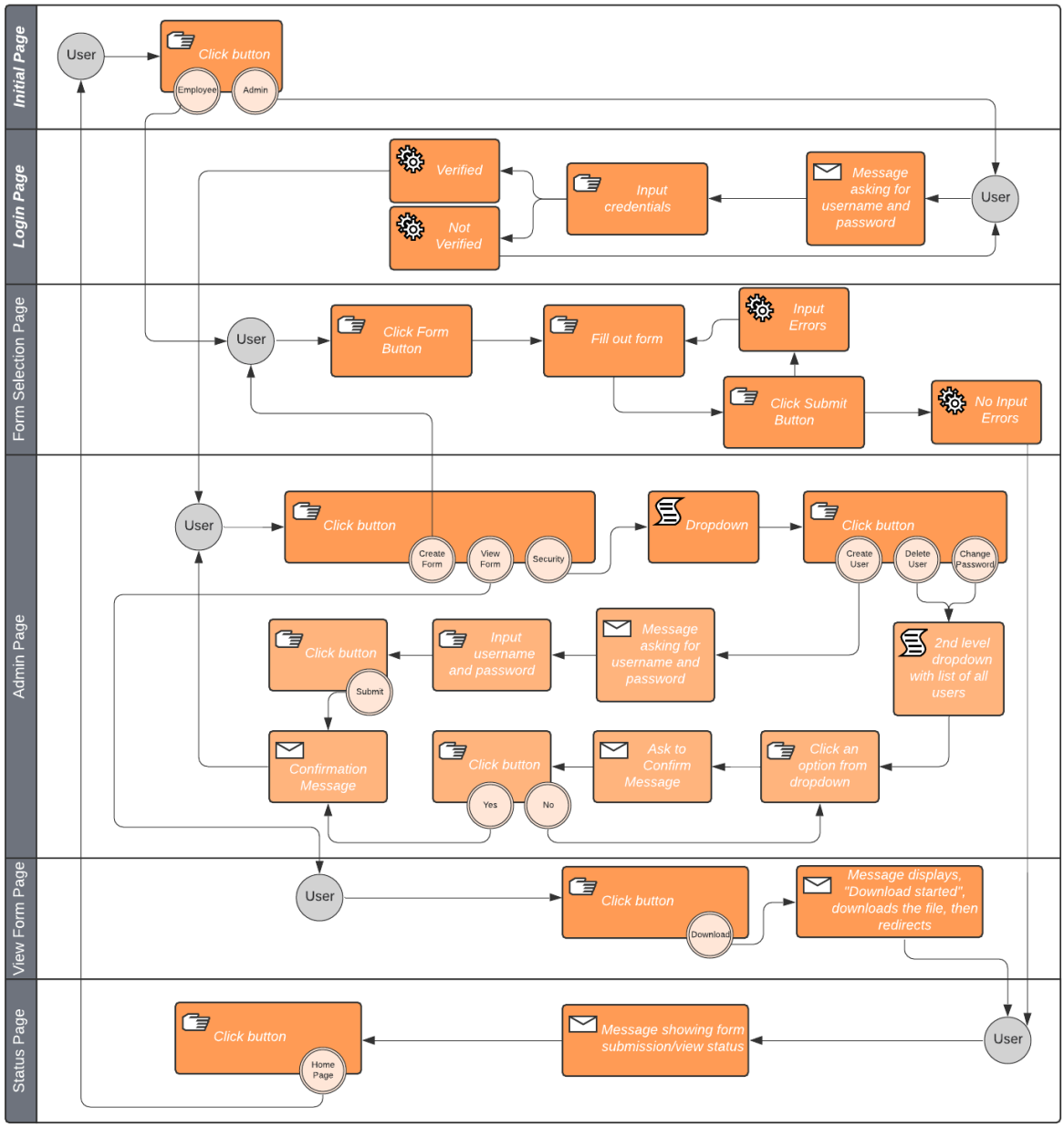
On the admin page the user has three options. If they press the create form button, they will be redirected to the form selection page. This page functions the same as it would for the employee with the addition of form 04. The other two options are the view form button and the security button.

If the user clicks the view form button they will be redirected to the view form page. The user would click the download button and a message would appear confirming the download. After the download has started they would be redirected to the status page which functions the same as with a form submission.

If the user were to click the security button a dropdown list would appear with three options. The first option is the create user button. If the user clicks the create user button a box message with a submit button will appear prompting the user to input a username and password. After submitting they would get a confirmation message and the page would reset to the base state with no buttons clicked.

The second option is the delete user option. If the user clicked this button a second dropdown list would appear with all of the users currently registered. The user would select a user and a confirmation message would appear. If yes was selected the user would be deleted and a confirmation would appear, then resetting the page. If no was selected the message would disappear and the deletion would be canceled.

The third option is the change password option. This option functions the same way as the delete user option, instead in the confirm prompt, there would be a text box to enter the new password.


Initial Page
User
Click button
Employee
Admin
Login Page
Verified
Not Verified
Input credentials
Message asking for username and password
User
Form Selection Page
User
Click Form Button
Fill out form
Input Errors
Click Submit Button
No Input Errors
Admin Page
User
Click button
Create Form
View Form
Security
Dropdown
Click button
Create User
Delete User
Change Password
Click button
Submit
Input username and password
Message asking for username and password
2nd level dropdown with list of all users
Confirmation Message
Click button
Yes
No
Ask to Confirm Message
Click an option from dropdown
View Form Page
User
Click button
Download
Message displays, "Download started", downloads the file, then redirects
Status Page
Click button
Home Page
Message showing form submission/view status
User

2. Must Have Requirements

Standard user:

S1. Login –

Standard users, being FVSRA staff, need to be able to log in to a general employee account that does not have a password. This will be tested by attempting to enter the employee username into the login page.

S2. Form Creation –

Standard users need to be able to create forms. This information needs to be displayed mimicking the pdf forms provided by FVSRA. The information needs to be stored in a database. This will be tested by running a local server and attempting to populate information into a form for submission.

Admin:

A1. Login –

Admin users, being those deemed in charge of administrating this system via management at FVSRA, need to be able to log in to their accounts linked with the system. This will be tested by storing a username and password in a locally run database instance and attempting to enter said username and password into the login page.

A2. Form Creation –

Admin users need to be able to create forms. This information needs to be displayed mimicking the pdf forms provided by FVSRA. The information needs to be stored in a database. Admin users have an additional form that the standard user does not, form 4E. This will be tested by running a local server and attempting to populate information into a form for submission.

A3. Security –

Admins need to be able to use security feature. This feature includes creating a user, deleting a user, and changing passwords of a user. This will allow the admin to create other admin accounts that can access the admin forms and security feature. This will be tested with a locally run server, checking for the proper population or change in the database.

A4. View Form –

Admin users need to be able to view/download forms on file. This will be a feature listed in the admin panel and will allow the admin user to download the form in excel format. This will be tested by populating a form, submitting it, then attempting to download it, all ran on a local server.

Backend System:

D1. Database –

The system needs a functioning database to store all the data that goes along with the system. This data includes usernames and passwords for both admins and standard users. Each username must be unique. This data also includes all the data in relation to the forms.

3. Stretch Requirements

Str1. Form View –

Admin users could have the ability of viewing the forms in the web application in addition to being able to down them in excel format. This would be tested by populating a form and then attempting to view the newly created form in the web application run on a local server.

Str2. Form 4E PDF –

Admin users could have the ability of transferring any given Form 4E in the database directly onto the exact form 4E pdf file provided by FVSRA. This would be tested by populating a new form 4E and then attempting to download the newly created form in the proper pdf format, run on a local server.

Str3. Form Submission Page –

All users could have the feature of redirecting them from submitting a form to the submission page where there would be a message depending upon the status of their submission. This would be tested by populating a form, and then attempting to submit the form. Checking if the page properly reroutes on submission.

Str4. Additional Testing –

Additional testing could be done using a framework such as Mocha Chai. This would be done if the schedule allows.

4. Design Overview of the Product

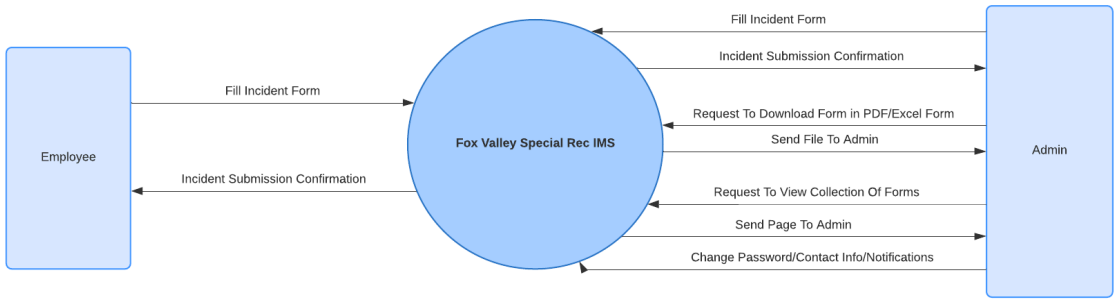
Data Design:

- This team shall utilize a SQL database that shall store the application data submitted via the forms filled by the user, as well as login information & contact info of said admin. Admins can also change who receives a notification when a form is submitted. For admins, they shall be required to have login credentials to access all the forms from

the employees that submitted the forms. Admins shall have the ability to change their contact information as well.

Context Diagram

This diagram shows a general overview of what the web-app can do.



Minor Injury Log	
minor_injury_id	PK
injury_date	
injury_time	
name_of_injured	
injury_location	
treatment	
how_injury_occurred	
facility_where_injury_occurred	
full_name_of_staff	

Acident Incident Report	
accident_incident_id	PK
Agency_Name	
Todays_Date	
Date_of_Incident	
Time_of_incident	
Name_of_the_person_Completing_the_report	
Title_Of_Person_Completing_the_report	
Business_Phone	
55+ More Fields	

Employee Injury Report	
employee_injury_id	PK
Agency_Name	
Todays_Date	
Time_of_Incident	
Name_of_person_completing_report	
Title_of_person_completing_report	
Business_phone	
How_did_the_incident_occur_provide_online_online_description	
40+ More Fields	

Notification Of Injury To Employer Report	
employee_injury_id	PK
Employee_Name	
Date_of_Incident	
Time_of_Incident	
Specific_location_of_accident	
Are_you_reporting_the_injury_for_the_first_time_using_this_form	
If_no_when_did_you_first_report_the_injury_And _to_whom_reported	
Describe_how_the_injury_occurred.	
10+ More Fields	

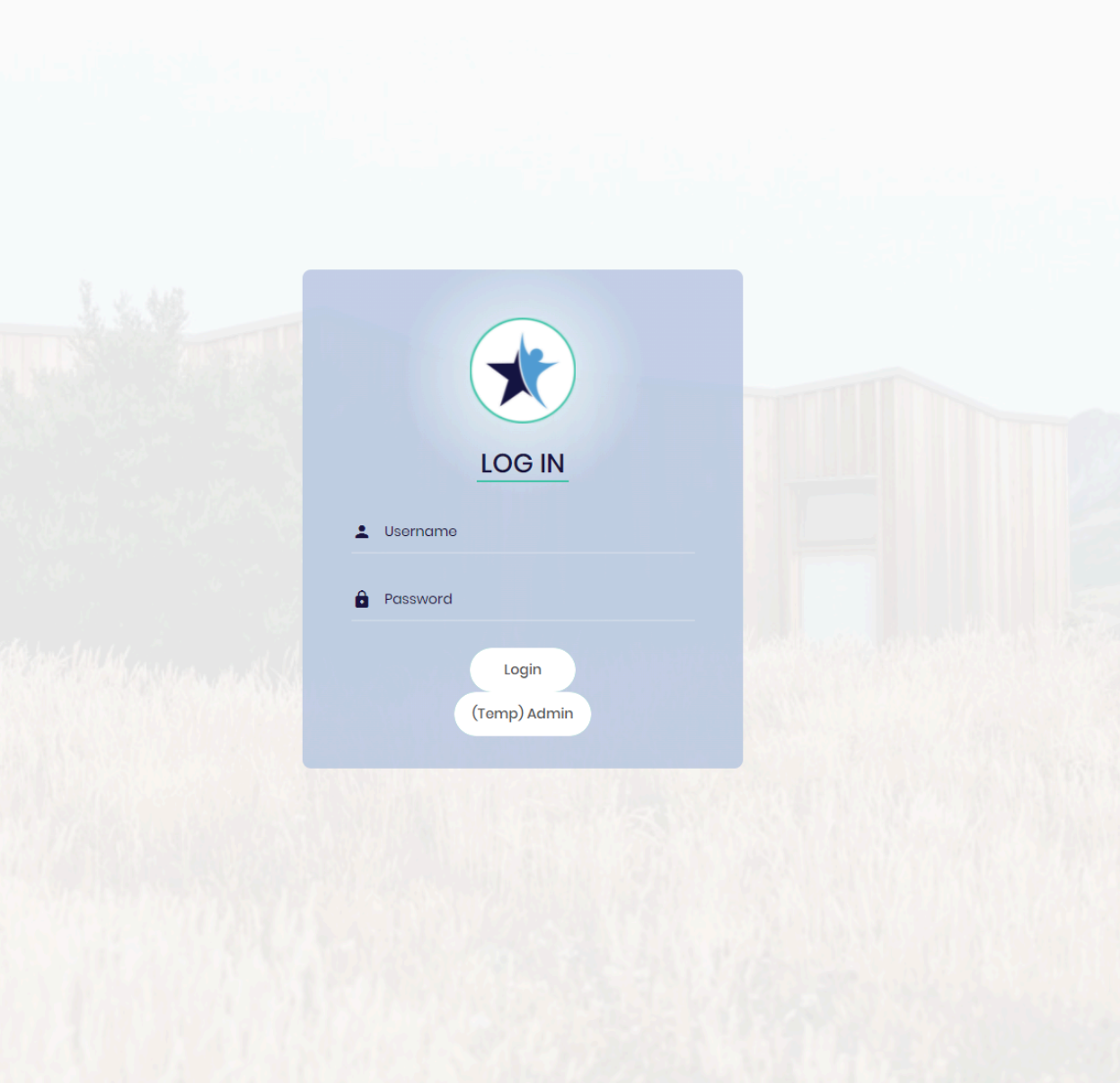
Property Loss Report	
property_loss_id	PK
Name_of_the_Agency	
Todays_Date	
Date_of_Incident	
Time_of_Incident	
Name_of_the_person_Completing_the_report	
Title_Of_Person_Completing_the_report	
How_did_the_incident_occur_and_what_property_was_damaged	
21+ More Fields	

Vehicle Accident Report	
vehicle_accident_id	PK
Name_of_the_Agency	
Todays_Date	
Date_of_Incident	
Time_of_Incident	
Name_of_the_person_Completing_the_report	
Title_Of_Person_Completing_the_report	
Business_Phone	
114+ More Fields	

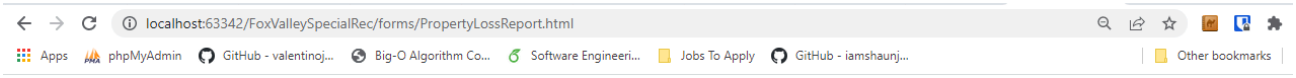
Architecture: This web application will be designed using a RESTful backend server with Node.js enabling forms to be stored persistently in an online database and viewed dynamically on a web page. The client will be able to communicate with the server by filling out a form, which the front-end will collect all the info from the required fields, store it into a JSON object, then using an AJAX call send it to the server.

- *Prototype –*

Login Page:



Form Example:



PDRMA
Park District Risk Management Agency

Property Loss Report
(For damage to agency property)
Attorney/Client Privileged Document

Form 03

1 Agency name

Today's date mm/dd/yyyy

2 Date of incident (mm/dd/yyyy) mm/dd/yyyy

Time of incident (hh/mm, a.m./p.m.) --:-- --

3 Name of person completing the report

Title of person completing report

4 Business phone 000-000-0000

Business email

5 How did the incident occur and what property was damaged? (Provide a brief factual summary.)

6 Name of the location (park, pool, community center; *Ex. Smith Pool, Johnson Community Center*) or nearest intersection where the incident occurred.

7 Is there an address for incident location? If yes, please provide the following:

Street address

City | State | Zip code 00000

8 Location (Specify the exact type of location/facility damaged, listing multiple locations/facilities if necessary. *Ex. maintenance garage, sports field*)

9 Primary location (Identify the exact area of damage. *Ex. tool storage, batting cage*)

10 Estimate of loss

11 Contact person at facility

12 Contact person's email

13 Contact person's phone number 000-000-0000

14 Was damage caused by third-party (non-agency) individual? Yes○ No○ Unknown◉

15 Has the party responsible for damage been identified? If yes, provide the following contact information for the person or persons identified:

Name | Street address

City | State | Zip code 00000

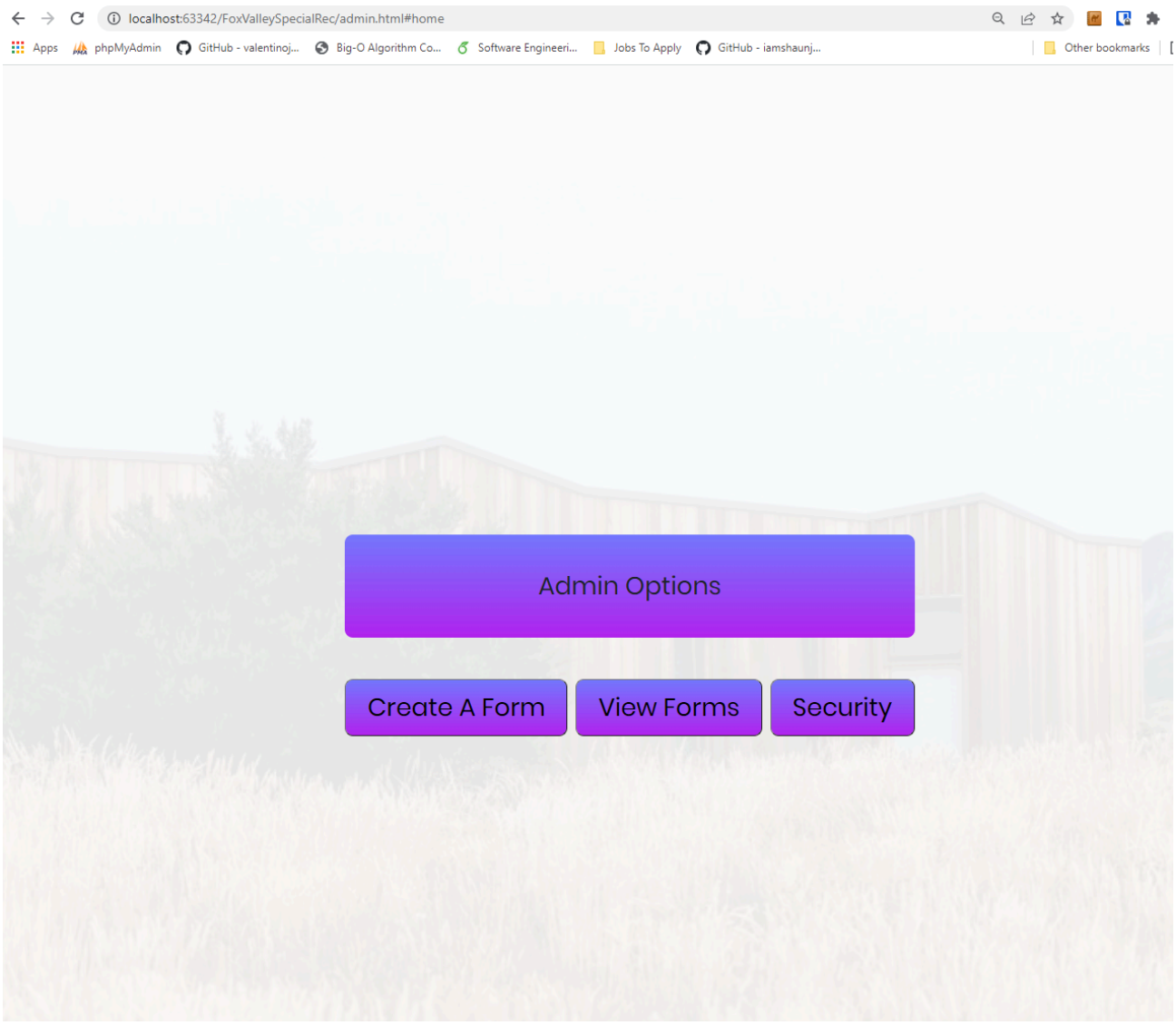
16 Has a police agency conducted an investigation? Yes○ No○ Unknown◉

17 What police agency investigated the incident? | What is the police report number?

18 Were criminal charges brought against the responsible party? If yes, what were the charges?

Submit

Admin Controls:



5. Verification

- ***Demo:***
- S1. Login - Access the web app on a device
- S2. Form Creation - Using postman, send a post request with JSON response containing all the required fields and return a status 200 (OK), as well as not including all the required fields and return a value of 400
- A1. Access the web app on a device and login with admin credentials
- A2. Form Creation - Using postman, send a post request with JSON response containing all the required fields and return a status 200 (OK), as well as not including all the required fields and return a value of 400
- A3. Access the webpage using admin credentials and be able to fill out a form to create a new user. An admin shall also be able to pick out a user from a table to delete its account. An admin shall also be able to fill out a form to change its password.
- Postman can also be used to send a POST request to verify a user can be added, and using a DELETE request to delete it, and a PUT request to change the password.
- A4. Access the webpage using admin credentials, be able to see a dynamic table with multiple form entries which can be viewed more in detail or be downloaded as a PDF/Excel file. Postman can be used using a GET request to verify that there are entries in the database.
- D1. Using Postman, and in the following section “Functional Testing”, using a variety of tests by using GET/POST/DELETE methods, we can verify that the database is able to view/create/store form entries properly. If the test returns a status of 200, it’s working properly.

- **Functional Testing:**

Postman Test

R#	TC #	Test Case Description	Input Data	Expected Result	Results

R1	T1	Get All Minor Injuries Records	Submit a GET request w/ the following url: {{url}}/fvsra/employeeInjuryReport	Return a status 200	PASS
	T2	Get One Minor Injury Log Record	Submit a GET request w/ the following url: {{url}}/fvsra/minorInjuryLog/11	Return a status 200	PASS
	T3	Create One Minor Injury Log Entry	Submit a POST request w/ all the fields filled out w/ the following url: {{url}}/fvsra/minorInjuryLog	Return a status 200	PASS
	T4	Create One Minor Injury Log Entry w/o all required data	Submit a POST request w/o all the fields filled out w/ the following url: {{url}}/fvsra/minorInjuryLog	Return a status 400	FAIL
	T5	Delete One Minor Injury Log Record	Submit a DELETE request w/ the following url: {{url}}/fvsra/minorInjuryLog/11	Return a status 200	PASS
R2	T6	Get All Form 01 Records	Submit a GET request w/ the following url: {{url}}/fvsra/accidentIncidentReport	Return a status 200	PASS
	T7	Get One Form 01 Record	Submit a GET request w/ the following url: {{url}}/fvsra/accidentIncidentReport/11	Return a status 200	PASS
	T8	Create One Form 01 Entry	Submit a POST request w/ all the fields filled out w/ the	Return a status 200	PASS

			following url: {{url}}/fvsra/accidentIncidentReport		
	T9	Create One Form 01 Entry w/o all required data	Submit a POST request w/o all the fields filled out w/ the following url: {{url}}/fvsra/accidentIncidentReport	Return a status 400	FAIL
	T10	Delete One Form 01 Record	Submit a DELETE request w/ the following url: {{url}}/fvsra/accidentIncidentReport	Return a status 200	PASS
R3	T11	Get All Form 02 Records	Submit a GET request w/ the following url: {{url}}/fvsra/vehicleAccidentReport	Return a status 200	PASS
	T12	Get One Form 02 Record	Submit a GET request w/ the following url: {{url}}/fvsra/vehicleAccidentReport/11	Return a status 200	PASS
	T13	Create One Form 02 Entry	Submit a POST request w/ all the fields filled out w/ the following url: {{url}}/fvsra/vehicleAccidentReport	Return a status 200	PASS
	T14	Create One Form 02 Entry w/o all required data	Submit a POST request w/o all the fields filled out w/ the following url: {{url}}/fvsra/vehicleAccidentReport	Return a status 400	FAIL
	T15	Delete One Form 02 Record	Submit a DELETE request w/ the following url: {{url}}/fvsra/vehicleAccidentReport	Return a status 200	PASS

R4	T16	Get All Form 03 Records	Submit a GET request w/ the following url: {{url}}/fvsra/propertyLossReport	Return a status 200	PASS
	T17	Get One Form 03 Record	Submit a GET request w/ the following url: {{url}}/fvsra/propertyLossReport/11	Return a status 200	PASS
	T18	Create One Form 03 Entry	Submit a POST request w/ all the fields filled out w/ the following url: {{url}}/fvsra/propertyLossReport	Return a status 200	PASS
	T19	Create One Form 03 Entry w/o all required data	Submit a POST request w/o all the fields filled out w/ the following url: {{url}}/fvsra/propertyLossReport	Return a status 400	FAIL
	T20	Delete One Form 03 Record	Submit a DELETE request w/ the following url: {{url}}/fvsra/propertyLossReport	Return a status 200	PASS
R5	T21	Get All Form 04 Records	Submit a GET request w/ the following url: {{url}}/fvsra/employeeInjuryReport	Return a status 200	PASS
	T22	Get One Form 04 Record	Submit a GET request w/ the following url: {{url}}/fvsra/employeeInjuryReport/11	Return a status 200	PASS
	T23	Create One Form 04 Entry	Submit a POST request w/ all the fields filled out w/ the following url: {{url}}/fvsra/employeeInjuryReport	Return a status 200	PASS

	T24	Create One Form 04 Entry w/o all required data	Submit a POST request w/o all the fields filled out w/ the following url: {{url}}/fvsra/employeeInjuryReport	Return a status 400	FAIL
	T25	Delete One Form 04 Record	Submit a DELETE request w/ the following url: {{url}}/fvsra/employeeInjuryReport	Return a status 200	PASS
R6	T26	Get All Form 04E Records	Submit a GET request w/ the following url: {{url}}/fvsra/NotificationOfInjuryEmployerReport	Return a status 200	PASS
	T27	Get One Form 04E Record	Submit a GET request w/ the following url: {{url}}/fvsra/NotificationOfInjuryEmployerReport/11	Return a status 200	PASS
	T28	Create One Form 04E Entry	Submit a POST request w/ all the fields filled out w/ the following url: {{url}}/fvsra/NotificationOfInjuryEmployerReport	Return a status 200	PASS
	T29	Create One Form 04E Entry w/o all required data	Submit a POST request w/o all the fields filled out w/ the following url: {{url}}/fvsra/NotificationOfInjuryEmployerReport	Return a status 400	FAIL
	T30	Delete One Form 04E Record	Submit a DELETE request w/ the following url: {{url}}/fvsra/NotificationOfInjuryEmployerReport	Return a status 200	PASS