Vaccination Tracker



Bilal Asghar

Advanced Higher Computing Science Project

Contents

[Analysis](#_gjdgxs) **3**

[Description of problem:](#_30j0zll) 3

[Scope:](#_1fob9te) 4

[Boundaries:](#_3znysh7) 4

[Constraints:](#_2et92p0) 5

[UML: Use Case Diagram](#_tyjcwt) 6

[Requirements Specifications:](#_3dy6vkm) 7

[Project plan](#_1t3h5sf) 7

[Design](#_4d34og8)10

[Entity Relationship Diagram](#_kwufpmet3d3s) 10

[Data Dictionaries](#_2s8eyo1) 11

[Query Designs](#_17dp8vu) 13

[PHP Pseudocode](#_35nkun2) 19

[User Interface Design](#_1ksv4uv) 29

[Implementation](#_44sinio) **36**

[01 Tables](#_f4lop6tz5swv) 36

[02 Patients Sample Data](#_3j2qqm3) 39

[Creating Patients Sample Data:](#_1y810tw) 40

[03 Vaccination Types Sample Data](#_4i7ojhp) 41

[Creating Vaccination Types Sample Data:](#_2xcytpi) 42

[04 Medical Persons Sample Data](#_1ci93xb) 43

[Creating Medical Persons Sample Data:](#_3whwml4) 44

[05 Vaccination Centres Sample Data](#_2bn6wsx) 45

[Creating Vaccination Centres Sample Data:](#_qsh70q) 46

[06 Patient Vaccinations Sample Data](#_3as4poj) 47

[Creating Patient Vaccinations Sample Data:](#_1pxezwc) 48

[07 Stored Procedures](#_49x2ik5) 49

[PHP Code](#_147n2zr) 65

[Testing the Final Solution](#_3o7alnk) 122

[Final Test Plan](#_23ckvvd) 122

[Test Table](#_3fwokq0) 124

[Requirements Testing](#_1v1yuxt) 129

[Evaluation of the solution](#_2u6wntf) 145

[Fitness for Purpose](#_19c6y18) 145

[Maintainability](#_3tbugp1) 146

[Robustness](#_28h4qwu) 149

[Overall Evaluation](#_nmf14n) 152

# Analysis

## Description of problem:

I’m making a database design and development project that integrates with a web user interface. The application I’m creating allows a medical person to insert a patient vaccination into the system. They can also view their vaccination history which includes their patients historical vaccination data. There is also an Administrator that can view Administrator Reports. There are 3 reports that can be selected. Report 1 displays the total number of covid vaccinations carried out by area. Report 2 is a report on the number of patients for each type of vaccination. Report 3 displays the number of vaccinations carried out by each centre. For this I needed a relational database with 5 tables. When a patient gets a vaccination, it is done at a vaccination centre, for this I created a VaccinationCentres table to store the details of the centre such as the Name, Address, Postcode and Telephone because a patient may need to contact their vaccination centre. At the vaccination centre the medical person will need to confirm with the patient for the type of vaccination they are having, I created a VaccinationTypes table to store the name of the vaccination type. I also needed a table to store the details and contact information of the medical person who was carrying out the vaccination named MedicalPersons. This table stored the medical persons First Name, Last Name, Address, Postcode, Telephone and Profession. The details of the patient also have to be recorded so I created a table named Patients which stored a PatientUniqueId to identify each patient uniquely, patient First Name, Last Name, DateOfBirth, Address, Postcode and Telephone number. There can be more than one patient in fact there can be many, so I had to create a PatientVaccinations table where it stores the date and time of the vaccination, VaccinationCentreId for which centre the vaccination was given, PatientId for what patient received the vaccination. The MedicalPersonId pointing to whatever medical person carried out the vaccination and finally the VaccinationTypeId for the type of vaccination that was given. My project meets all the advanced higher concepts as it has 5 tables which is more than the minimum required 4 tables, it uses a query across 5 tables to get a list of vaccination history, it uses the between operator for Report 2 to show number of patients for each type of vaccination between 01/01/2020 and the current date.

## Scope:

The scope of my project will include:

* Fully completed in depth analyses with detailed description of problem, completed UML Use Case Diagram, requirements specification of project and detailed project plan.
* Fully completed design
  + Entity Relationship Diagram
  + Data Dictionaries
  + Query Designs
  + PHP Pseudocode
  + Wireframes for user interface
* Fully working implementation
* Fully completed test plan with descriptions of a test persona, test cases and expected outputs.
* The results of the final testing that was done.
* Report on overall evaluation of my project including fitness for purpose and robustness.

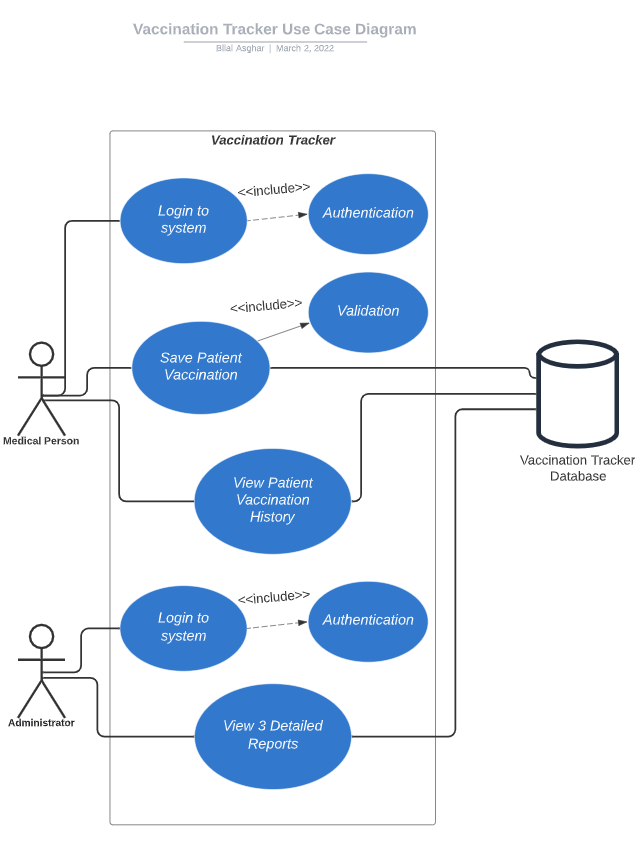
## Boundaries:

* Due to the time available for this project, the Medical Persons will already be registered and their details will be already inserted into the database. Also, there will be a large number of Patient details will also be in the database already with 32 Vaccination Types, 5 Vaccination Centres.
  + All fields in database must be greater than 0
* I will have a login system but it won’t be a full login system
  + Only have to select a user from the drop down list to log in as the implementation of a password is not part of my project and therefore not required for logging in.
* The login system will be validated so if there is no selected user or if something goes wrong the form should redirect back to the login page
* If the user logged in is valid and is the Demo Administrator then they would be directed to Administration Reports page and will be able to run reports on data in the database
* If the user logged in is valid and is a medical person/vaccinator then they would be directed to Vaccinate Patient page.
* When filling out the Vaccinate Patient form all fields will have a required attribute so the user has to enter a character. Also, if an empty value is submitted the database should not insert the data as it is not greater than 0 and should redirect the user to the patient vaccinate page with the query string of failed and display an error message.
* If entering the details successfully the data should be inserted into the database and redirected to patient vaccinate page with a success message.

## Constraints:

* My projected must be fully completed by the deadline 4th March 2021.
* This project must comply with Copyright, Design and Patents Act 1988 and with General Data Protection Data Regulation as it will use only test data.
* I will be coding the front end with PHP, html and css and will be using EasyPHP as my database server for phpMyAdmin using the MySQL language.
* There will be no costs in making this project.

## UML: Use Case Diagram



## Requirements Specifications:

End User Requirements:

* If the end user is a medical person:
  + They will be able to add a patient vaccination to the system.
  + Be able to view their vaccination history on all their patients.
* If the end user is a administrator:
  + They will be able to access administrator reports on data in the database.
* The user will be able to easily navigate throughout the site .
* The user will be to read everything displayed on the site with ease.

Functional Requirements:

* The user will need to log in to have any access to the site.
* If the end user is a medical person:
  + My site will store the patients vaccination details into a database
  + All form fields will be validated on the site and in the database
* My site will have already generated reports on the administrator reports page using data from database.
* The user are only allowed to add patient vaccination if they are given the authority, i.e. on the system as a medical person/vaccinator

## Project plan

Detailed Project Plan of each stage:

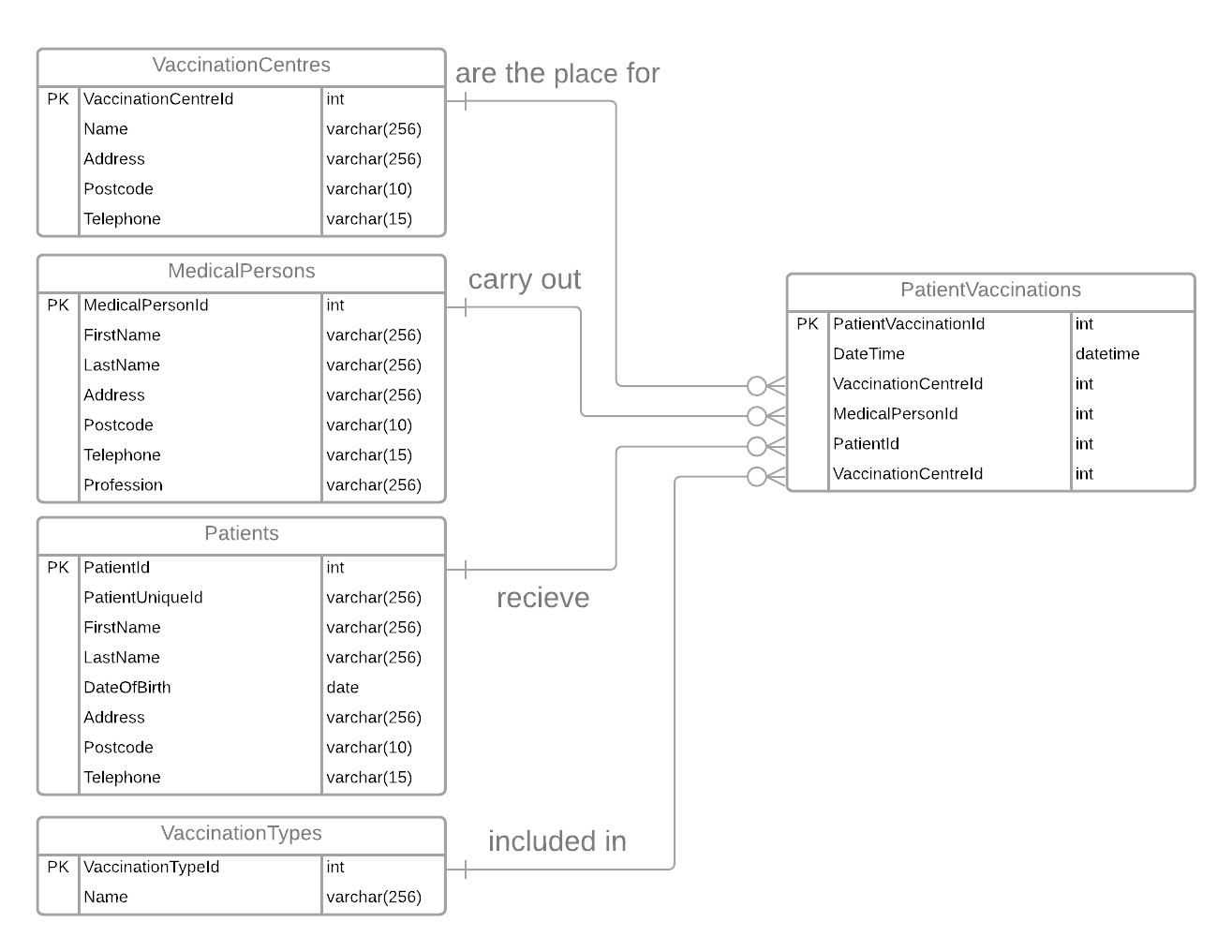
| **Task** | **Section Task** | **Time it took** | **Begin Date** | **Should Finish By Date** |
| --- | --- | --- | --- | --- |
| Analysis | Go through specification looking at all requirements for project and make project idea. | 0.5 Days | 3rd December | 3rd December |
| Start designing solution of project problem and start write up of documentation | 0.5 Days | 3rd December | 3rd December |
| Write up scope, boundaries and constraints | 1 Days | 10th December | 10th December |
| Draw out UML Case Diagram | 0.5 Days | 17th December | 17th December |
| End-User and functional requirements specification | 0.5 Days | 17th December | 17th December |
| Create Project Plan for remaining stages. | 1 Days | 22nd December | 22nd December |
| Design | Create designs and wireframes for User Interface | 2 Days | 5th January | 7th  January |
| Design Entity Relationship Diagram | 0.5 Days | 11th January | 11th January |
| Design data dictionaries | 0.5 Days | 11th January | 11th January |
| Query Designs | 0.5 Days | 12th January | 12th January |
| Web design | 0.5 Days | 12th January | 12th January |
| Implementation | Create Database Structure | 2 Days | 14th January | 18th January |
| Create and insert test data | 1 Days | 19th January | 19th January |
| Write up queries | 3 Days | 21st January | 26th January |
| Create user interface | 2 Days | 28th January | 1st February |
| Write up new skill learned during development | 0.5 Days | 2nd February | 2nd February |
| Log of testing throughout implementation | Is done throughout implementation | 14th January | 3rd February |
| Testing | Test plan | 2 Days | 5th February | 9th February |
| Begin testing solution | 2 Days | 10th February | 12th February |
| Write up any problems highlighted by testing | 1 Days | 15th February | 15th February |
|  |  |  |  |
| Evaluation  Finalise All Evidence | Write up report on evaluation of solution | 1 Days | 16th February | 16th February |
| Check over Analysis | 0.5 Days | 18th February | 18th February |
| Check over Design | 0.5 Days | 18th February | 18th February |
| Check over Implementation | 1 Days | 22nd February | 22nd February |
| Check over Testing and Evaluation | 1 Days | 23nd February | 23nd February |
| Submit final version of plan as evidence | 1 Days | 25nd February | 25nd February |

During the development of this project I will need access to resources to implement my solution:

| Analysis | * Microsoft Word 2016 * Google Chrome |
| --- | --- |
| Design | * Microsoft Word 2016 * Google Chrome * wireframes & Entity Relationship Diagram   + <https://www.lucidchart.com/pages/>   + <https://wireframe.cc/> |
| Implementation | * EasyPHP Server * MyPHPAdmin (MySQL web software tool) * Brackets (Integrated Development Environment) * Bootstrap * Font Awesome * SQL Server * Notepad ++ * <https://www.tailorbrands.com/> * https://www.doogal.co.uk/RandomAddresses.php * <https://spottedtiger.tripod.com/D_Language/D_Randomly_Gen_Data_XP.html> * https://www.cdc.gov/vaccines/programs/iis/code-sets/vis-barcode-lookup-table.html |
| Final Testing | * EasyPHP Server * MyPHPAdmin(MySQL web software tool) * Brackets (Integrated Development Environment) * Microsoft Word 2016 |
| Evaluation | * Microsoft Word 2016 |

# Design

## Entity Relationship Diagram



## Data Dictionaries

| **Patients Table** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Field Name | Key | Type | Size | Required | Validation |
| PatientId | PK | Int |  | Yes | Auto\_Increment |
| PatientUniqueId |  | Varchar | 256 | Yes | Length > 0 |
| FirstName |  | Varchar | 256 | Yes | Length > 0 |
| LastName |  | Varchar | 256 | Yes | Length > 0 |
| DateOfBirth |  | Date | 256 | Yes |  |
| Address |  | Varchar | 256 | Yes | Length > 0 |
| Postcode |  | Varchar | 10 | Yes | Length > 0 |
| Telephone |  | Varchar | 15 | Yes | Length > 0 |

| **MedicalPersons Table** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Field Name | Key | Type | Size | Required | Validation |
| MedicalPersonId | PK | Int |  | Yes | Auto\_Increment |
| FirstName |  | Varchar | 256 | Yes |  |
| LastName |  | Varchar | 256 | Yes |  |
| Address |  | Varchar | 256 | Yes |  |
| Postcode |  | Varchar | 10 | Yes |  |
| Telephone |  | Varchar | 15 | Yes |  |
| Profession |  | Varchar | 256 | Yes |  |

| **VaccinationCentres Table** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key | Type | Size | Required | Validation |
| VaccinationCentreId | PK | Int |  | Yes | Auto\_Increment |
| Name |  | Varchar | 256 | Yes |  |
| Address |  | Varchar | 256 | Yes |  |
| Postcode |  | Varchar | 10 | Yes |  |
| Telephone |  | Varchar | 15 | Yes |  |

| **VaccinationTypes Table** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key | Type | Size | Required | Validation |
| VaccinationTypeId | PK | Int |  | Yes | Auto\_Increment |
| Name |  | Varcchar | 256 | Yes |  |

| **PatientVaccinations Table** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Key | Type | Size | Required | Validation |
| PatientVaccinationId | PK | Int |  | Yes | Auto\_Increment |
| DateTime |  | DateTime |  | Yes |  |
| VaccinationCentreId | FK | Int |  | Yes | References VaccinationCentres(VaccinationCentreId) |
| MedicalPersonId | FK | Int |  | Yes | References MedicalPersons(MedicalPersonId) |
| PatientId | FK | Int |  | Yes | References Patients(PatientId) |
| VaccinationTypeId | FK | Int |  | Yes | References VaccinationTypes(VaccinationTypeId) |

## Query Designs

spGetUsernames:

This Query design returns a list of medical person usernames ordered by first and last name

| Fields / Calculations | MedicalPerson,  CONCAT(FirstName + “ ” + LastName + “ (” + Profession + “) “) AS Username |
| --- | --- |
| Tables | MedicalPersons |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort | FirstName, LastName |

spGetMedicalPerson:

This Query design gets the MedicalPersonId of the user logged in

| Fields / Calculations | \* |
| --- | --- |
| Tables | MedicalPersons mp |
| Criteria | mp.MedicalPersonId = < P\_medicalPersonId value from php session variable > |
| Grouping |  |
| Having |  |
| Sort |  |

spGetVaccinationCentres:

This Query design returns all vaccination centres ordered by name

| Purpose | Returns all vaccination centres |
| --- | --- |
| Fields / Calculations | VaccinationCentreId, Name |
| Tables | VaccinationCentres |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort | Name |

spGetVaccinationTypes:

This Query design returns all vaccination types ordered by name

| Fields / Calculations | VaccinationTypeId, Name |
| --- | --- |
| Tables | VaccinationTypes |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort | Name |

spGetVaccinationHistory:

This Query design is used within Temp\_VaxHistory temporary table and returns all vaccination types ordered by pv.DateTime DESC

| Purpose | Return all vaccination types |
| --- | --- |
| Fields / Calculations | DATE\_FORMAT(pv.DateTime, "%d %b %Y at %h:%i %p") AS DateTime, vc.Name AS VaccinationCentre, CONCAT(p.FirstName , ' ' , p.LastName) AS PatientName, vt.Name AS VaccinationType |
| Tables | medicalpersons mp, patientvaccinations pv, vaccinationcentres vc, patients p, vaccinationtypes vt |
| Criteria | mp.MedicalPersonId = < p\_MedicalPersonId value from php session variable > |
| Grouping |  |
| Having |  |
| Sort | pv.DateTime DESC |

This query design selects all the data from Temp\_VaxHistory and assigns a row number

| Fields / Calculations | \*, (@row\_number:=@row\_number + 1) AS RowNum |
| --- | --- |
| Tables | Temp\_VaxHistory |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort |  |

spSavePatientVaccination:

This Query design checks if a patientId exists, if it does then it will update the patient details otherwise it will create a new patient in the Patients table and insert the details. After this check it inserts the patient vaccinations details into PatientVaccinations table.

| Fields / Calculations | PatientId INTO @PatientId |
| --- | --- |
| Tables | Patients |
| Criteria | PatientUniqueId = < p\_PatientUniqueId value from form > |
| Grouping |  |
| Having |  |
| Sort |  |

IF @PatientId IS NULL THEN

| INSERT INTO Patients | PatientUniqueId, FirstName, LastName, DateofBirth, Address, Postcode, Telephone |
| --- | --- |
| VALUES | < p\_PatientUniqueId, p\_PatientFirstName, p\_PatientLastName, p\_PatientDOB, p\_PatientAddress, p\_PatientPostcode, p\_PatientTelephone  values from form > |

SET @PatientId := LAST\_INSERTED\_ID();

ELSE

| UPDATE | Patients |
| --- | --- |
| SET | FirstName = p\_PatientFirstName, LastName = p\_PatientLastName, DateofBirth = p\_PatientDOB, Address = p\_PatientAddress, Postcode = p\_PatientPostcode, Telephone = p\_PatientTelephone  < update existing patient details with valuess from form > |
| Criteria | PatientId = @PatientId |

END IF;

| INSERT INTO PatientVaccinations | DateTime, VaccinationCentreId, MedicalPersonId, PatientId, VaccinationTypeId |
| --- | --- |
| VALUES | < p\_DateTime, p\_VaccinationCentreId, p\_MedicalPersonId, @PatientId, p\_VaccinationTypeId  values from form > |

spGetReportVaccinationsByCentre:

This query design gets a list of vaccinations carried out by each centre

| Fields / Calculations | vc.Name, vc.Address, vc.Postcode, vc.Telephone, COUNT(pv.VaccinationCentreId) AS NumberOfVaccinations |
| --- | --- |
| Tables | Patient Vaccinations pv, Vaccination Centres vc |
| Criteria |  |
| Grouping | vc.VaccinationCentreId |
| Having |  |
| Sort | NumberOfVaccinations DESC |

spGetReportPatientsByVaccinationTypes:

This query design selects the total number of patients into @TotalPatients variable

| Fields / Calculations | COUNT(\*) INTO @TotalPatients |
| --- | --- |
| Tables | Patients |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort |  |

This query design will use the above query design to use @TotalPatients variable

| Purpose | This will get a list of vaccinations carried out by each centre |
| --- | --- |
| Fields / Calculations | vt.Name, COUNT(pv.VaccinationTypeId) AS NumberOfPatients, COUNT(pv.VaccinationTypeId) / @TotalPatients \* 100 AS PercentOfPatients |
| Tables | Patient Vaccinations pv, Vaccination Centres vc |
| Criteria | pv.DateTime BETWEEN DATE("2020-01-1") AND DATE(CURDATE()) |
| Grouping | vt.VaccinationTypeId |
| Having |  |
| Sort | NumberOfPatients DESC |

spGetReportCovidVaccinationsByArea:

Create a temporary table called Temp\_AreaVaxDetails

| Purpose | Return all vaccination types |
| --- | --- |
| Fields / Calculations | SUBSTRING\_INDEX(p.Address, ', ', -1) AS Area,  vt.name AS Vaccine,  COUNT(\*) AS NumVaxByArea,  0 AS TotalVax |
| Tables | Patients p, PatientVaccinations pv, vaccinationtypes vt |
| Criteria | vt.name LIKE "COVID-19%" |
| Grouping | Area, vt.VaccinationTypeId |
| Having |  |
| Sort |  |

Create a temporary table called Temp\_AreaVaxQty

| Purpose | Return all vaccination types |
| --- | --- |
| Fields / Calculations | Area, SUM(NumVaxByArea) AS Qty |
| Tables | Temp\_AreaVaxDetails |
| Criteria |  |
| Grouping | Area |
| Having |  |
| Sort |  |

| UPDATE | Temp\_AreaVaxDetails t1, Temp\_AreaVaxQty t2 |
| --- | --- |
| SET | t1.TotalVax = t2.Qty |
| Criteria |  |

| Fields / Calculations | \* |
| --- | --- |
| Tables | Temp\_AreaVaxDetails |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort | TotalVax DESC, Area,  NumVaxByArea DESC,  Vaccine |

Drop temporary tables that are no longer required:

DROP TEMPORARY TABLE Temp\_AreaVaxDetails;

DROP TEMPORARY TABLE Temp\_AreaVaxQty;

spGetTotalPatients:

This query design gets the total number of patients from Patients table

| Fields / Calculations | COUNT(\*) AS TotalPatients |
| --- | --- |
| Tables | Patients |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort |  |

spGetTotalVaccinations:

This query design gets the total number of vaccinations from PatientVaccinations table

| Fields / Calculations | SELECT COUNT(\*) AS TotalVaccinations |
| --- | --- |
| Tables | PatientVaccinations |
| Criteria |  |
| Grouping |  |
| Having |  |
| Sort |  |

spGetTotalCovidVaccinations:

This query design gets the total number of covid vaccinations from PatientVaccinations table

| Fields / Calculations | SELECT COUNT(\*) AS TotalCovidVaccinations |
| --- | --- |
| Tables | PatientVaccinations pv, VaccinationTypes vt, |
| Criteria | vt.name LIKE "COVID-19%" |
| Grouping |  |
| Having |  |
| Sort |  |

spGetTotalMedicalPersonVaccinations:

This query design gets the total number of vaccinations for a medical person

| Fields / Calculations | SELECT COUNT(\*) AS TotalMedicalPersonVaccinations |
| --- | --- |
| Tables | MedicalPersons mp, PatientVaccinations pv, |
| Criteria | mp.MedicalPersonId = < p\_MedicalPersonId value from php session variable > |
| Grouping |  |
| Having |  |
| Sort |  |

## PHP Pseudocode

On Page Load Process:

* 1. When project is first loaded always redirect to Login.php page

Login Page Process:

1.1 Start Session

1.2 If $\_SESSION[“medicalPersonId”] is null and current page is Login.php

1.3 Then redirect to Login.php page

1.4 Include DatabaseHelpers.php

1.5 If current page is not Login.php then show log out button

1.6 Include Login.php page $page\_header

1.7 Include Login\_Content.php $page\_content

1.8 $usernames = GetUsernames() which get an array of medical person usernames from calling SQL stored procedure spGetUsernames

1.9 In select element foreach loop through $usernames

1.10 Select option value = <? $username[‘MedicalPersonId’] ?> <?= $username[‘Username’]

1.12 After pressing Login button the form points to Login\_Authenticate.php

1.13 Now going throught Login\_Authenticate.php

1.14 $username = $\_POST[“txtUsername”]

1.15 $\_SESSION[“medicalPersonId”] = $username

1.16 IF $username == “DemoAdministrator” THEN

1.17 Redirect to AdminReports.php

1.18 ELSE IF $username is not null

1.19 Redirect to VaccinatePatient.php

1.20 ELSE

1.21 Invalid username redirect back to Login.php page

1.22 END IF

1.23 Die() to exit Login\_Authenticate.php script

Reports Page Process:

1.1 Then redirect to Login.php page

1.2 Include DatabaseHelpers.php

1.3 If current page is note Login.php then show log out button

1.4 Include Login.php page $page\_header

1.5 Include Login\_Content.php $page\_content

1.6 $reportType = $\_GET[‘ReportType’] from query string

1.7 SWITCH ($reportType)

1.8 Case $reportType = “1” then include ReportType\_1.php file

1.9 Case $reportType = “2” then include ReportType\_2.php file

1.10 Case $reportType = “3” then include ReportType\_3.php file

1.11 END SWITCH

Logout Process:

1.1 Start Session

1.2 If $\_SESSION[“medicalPersonId”] is null and current page is Login.php

1.3 Then redirect to Login.php page

1.4 Include DatabaseHelpers.php

1.5 If current page is not Login.php then show log out button

1.6 Include Login.php page $page\_header

1.7 Include Login\_Content.php $page\_content

1.8 Starts Session

1.9 Unset all session variables

1.10 Destroy all data associated with session variables

1.11 Direct to Login.php page

Report 1 Page Process:

1.1 Then redirect to Login.php page

1.2 Include DatabaseHelpers.php

1.3 If current page is not Login.php then show log out button

1.4 Include Login.php page $page\_header

1.5 Include Login\_Content.php $page\_content

1.6 Set variable $totalCovidVaccinations = GetTotalCovidVaccinations() which gets the total number of covid vaccinations carried out by calling SQL stored procedure spGetTotalCovidVaccinations

1.7 Set variable $reportData = GetReportCovidVaccinationsByArea() which gets an array of data by calling SQL stored procedure

spGetReportPatientsByVaccinationType

1.8 In table body FOR $I = 0; $I < totalItems in $reportData array $I++

1.9 Assign $reportData[$i] to $item

1.10 Assign $item[‘Area’] to $currentArea

1.11 IF $currentArea not equal to $previous

1.12 SET $previous = $currentArea

1.13 Assign $item[‘Area’] and $item[TotalVax’] in two <td> elements

1.14 SET $p=$I

1.15 WHILE $currentArea == $reportData[$p][‘Area’]

1.16 Assign $item[Vaccine’] and $item['NumVaxByArea'] in third <td> element

1.17 Increment while loop index $p++

1.18 END WHILE

1.19 END IF

1.20 END FOR LOOP

Report 2 Page Process:

1.1 Then redirect to Login.php page

1.2 Include DatabaseHelpers.php

1.3 If current page is not Login.php then show log out button

1.4 Include Login.php page $page\_header

1.5 Include Login\_Content.php $page\_content

1.6 $totalPatients = GetTotalPatients() which gets the total number of patients from the patients table by calling SQL stored procedure spGetTotalPatients

1.7 $reportData = GetReportPatientsByVaccinationType() which gets an array of data by calling SQL stored procedure sp GetReportPatientsByVaccinationType

1.8 In table body FOREACH $reportData as $i

1.9 Inside each <td> there is a <span> element

1.10 Within the first <span> element the $i index of the $reportData Name is assigned

1.11 Within the second <span> element the $i index of the $reportData NumberOfPatients is assigned

1.12 Within the first <span> element the $i index of the $reportData PercentOfPatients is assigned

Report 3 Page Process:

1.1 Then redirect to Login.php page

1.2 Include DatabaseHelpers.php

1.3 If current page is not Login.php then show log out button

1.4 Include AdminReports.php.php page $page\_header

1.5 Include AdminReports\_Content.php $page\_content

1.6 $totalVaccinations = GetTotalVaccinations () which gets the total number of total vaccinations carried out by calling SQL stored procedure spGetTotalPatients

1.7 $reportData = GetReportVaccinationByCentre() which gets an array of data by calling SQL stored procedure sp GetReportPatientsByVaccinationType

1.8 In table body FOREACH $reportData as $i

1.9 Inside each <td> there is a <span> element

1.10 There are 2 <td>

1.11 Within the first <td> element 4 <span> elements have the $i index of the $reportData Name, Address, Postcode, Telephone is assigned

1.12 Within the secodn <td> the <span> element the $i index of the $reportData NumberOfVaccinations is assigned

Vaccinate Patient Page Process:

1.1 Start Session

1.2 If $\_SESSION[“medicalPersonId”] is null and current page is Login.php

1.3 Then redirect to Login.php page

1.4 Include DatabaseHelpers.php

1.5 If current page is note Login.php then show log out button

1.6 Include VaccinatePatient.php page $page\_header

1.7 Include VaccinatePatient\_Content.php $page\_content

1.8 $medicalPerson = GetMedicalPerson($\_SESSION[‘MedicalPersonId’])[0] which get the first value in the array of medical person details from calling SQL stored procedure spGetMedicalPerson

1.9 $vaccinationCentres = GetVaccinationCentres() this will get a list of all 5 vaccination centres by calling SQL stored procedure spGetVaccinationCentres

1.10 $vaccinationTypes= GetVaccinationTypes () this will get a list of all 32 vaccines types by calling SQL stored procedure spGetVaccinationTypes

1.11 $currentDateTime = date(“Y-m-d|TH:i”) this will get the current date and time

1.12 IF $\_GET[‘SavedPatient’] == “success” THEN

1.13 Show <div> element with success message with green background

1.14 ELSE IF $\_GET[‘SavedPatient’] == “failed” THEN

1.15 Show <div> element with error message with red background

1.16 Using $medicalPerson variable display medical person FirstName, LastName and Profession in <h3> tag

1.17 Using $medicalPerson variable display medical person Address, Postcode and Telephone in <div> tag with muted text style

1.18 For Vaccination Centre drop down list

1.19 FOREACH $vaccinationCentres as $vaccinationCentre

1.20 Set the value of each option to $vaccinationCentre[‘VaccinationCentreId’] to id the drop down option relating to centre id in VaccinationCentres table within in database.

1.21 Set the text to be displayed in drop down option to be $vaccinationCentre[‘Name’] for the name of the centre

1.20 For datetime input fill current date time in

1.21 Date time input value=”<?= $currentDatetIME; ?>”

1.22 For Vaccination type drop down list

1.23 FOREACH $vaccinationTypes as $vaccinationType

1.24 Set the value of each option to $vaccinationType [‘VaccinationTypeId’] to id the drop down option relating to vaccine id in VaccinationTypes table within in database.

1.25 Set the text to be displayed in drop down option to be $vaccinationType[‘Name’] for the name of the vaccine

1.26 Once save patient button is pressed it looks at VaccinatePatient\_Save.php

Vaccinate Patient Save Process:

1.1 Start Session

1.2 POST all data from form fields and assign to variables

1.3 Include DatabaseHelpers.php to access functions for database stored procedures

1.4 Call SavePatientVaccination() function from DatabaseHelpers.php that passes through all the form field data

1.5 Then after saving patient data redirecct back to VaccinatePatient.php page with a query string checking if ?SavePatient is a success or failed

Vaccination History Page Process:

1.1 Start Session

1.2 If $\_SESSION[“medicalPersonId”] is null and current page is Login.php

1.3 Then redirect to Login.php page

1.4 Include DatabaseHelpers.php

1.5 If current page is note Login.php then show log out button

1.6 Include VaccinationHistory.php page $page\_header

1.7 Include VaccinationHistory\_Content.php $page\_content

1.8 SET $medicalPerson TO GetMedicalPerson() function from DatabaseHelpers.php which takes in $\_SESSION[‘medicalPersonId’][0] as parameter

1.9 SET $totalMedicalPersonVaccinations TO GetTotalMedicalPersonVaccinations() function from DatabaseHelpers.php which takes in $\_SESSION[‘medicalPersonId’] as parameter

1.10 SET $vaccinationHistory TO GetVaccinationHistory() function from DatabaseHelpers.php which takes in $\_SESSION[‘medicalPersonId’] as parameter

1.11 In <h3> element display $medicalPerson FirstName, LastName, Professsion

1.12 In <div> element below <h3> element display $medicalPerson Address, Postcode, Telephone

1.13 Display Total Number of Vaccinations: $totalMedicalPersonVaccinations

1.14 In table body FOREACH $vaccinationHistory as $vh

1.9 Inside <tr> element there is 5 <td> elements

1.10 Within each <td> element display the following $vh RowNum, DateTime, VaccinationCentre, PatientName, VaccinationType

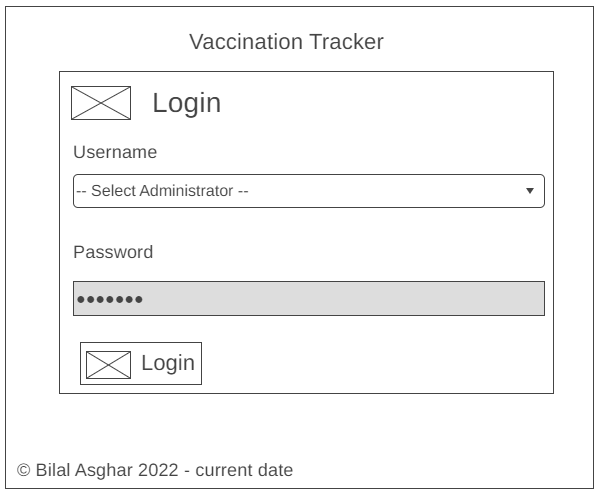
1.11 END FOREACH

## User Interface Design

Login Page Wireframe:





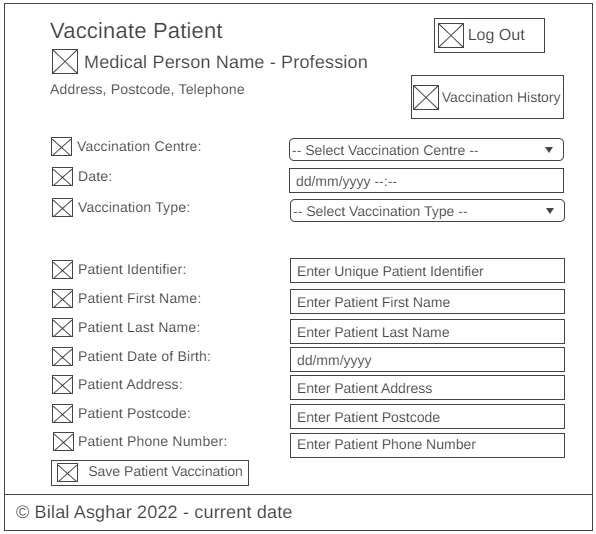




Vaccinate Patient Page Wireframe:



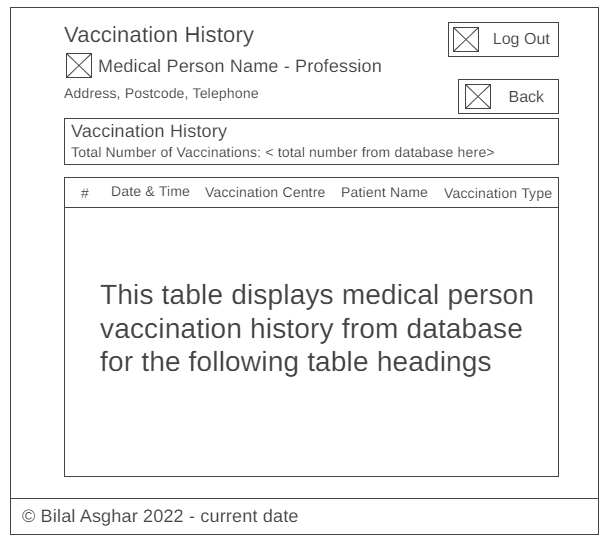






Vaccination History Page Wireframe:

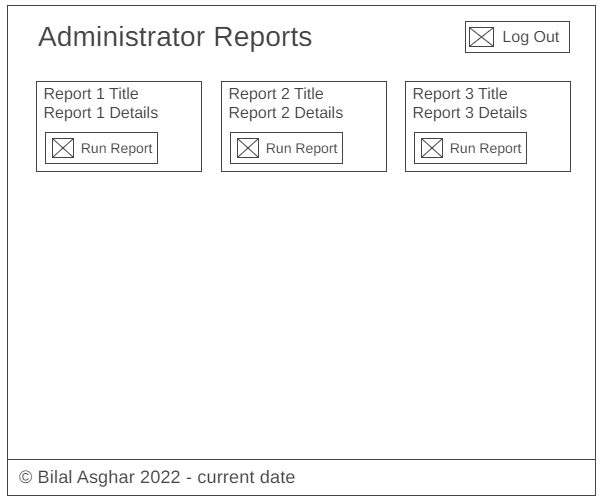






Administrator Reports Page Wireframe:





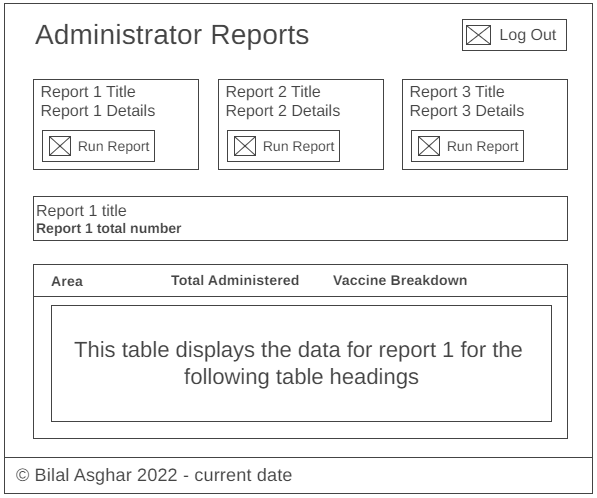


Administrator Reports Page Wireframe – When report 1 is clicked:

Query string **?ReportType=1** at the end of url









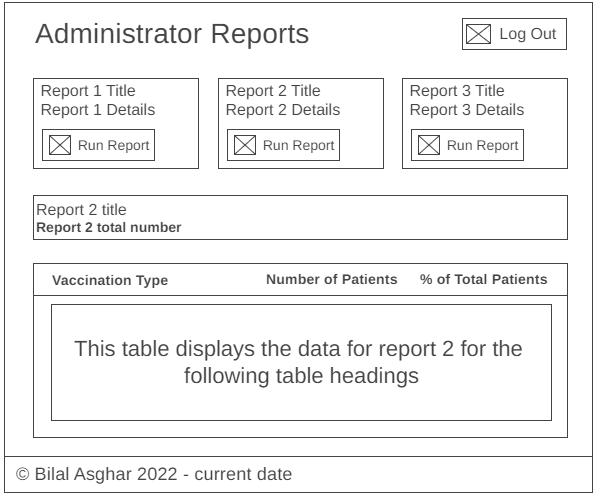
Administrator Reports Page Wireframe – When report 2 is clicked:

Query string **?ReportType=2** at the end of url



\



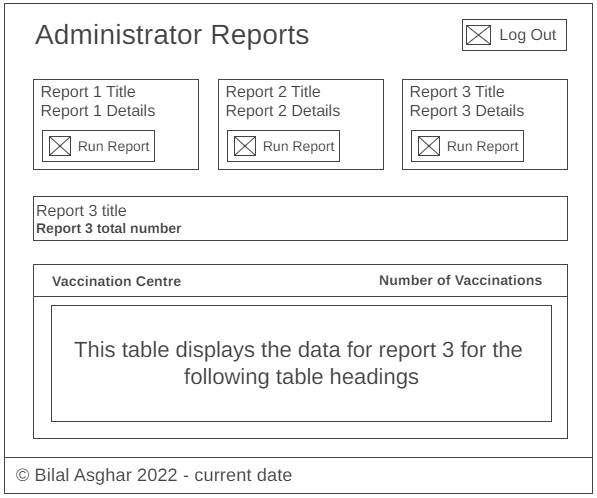




Administrator Reports Page Wireframe – When report 3 is clicked:

Query string **?ReportType=3** at the end of url







# Implementation

## 01 Tables

MedicalPersons Table:

CREATE TABLE `medicalpersons` (

`MedicalPersonId` int NOT NULL AUTO\_INCREMENT,

`FirstName` varchar(256) NOT NULL,

`LastName` varchar(256) NOT NULL,

`Address` varchar(256) NOT NULL,

`Postcode` varchar(10) NOT NULL,

`Telephone` varchar(15) NOT NULL,

`Profession` varchar(256) NOT NULL,

PRIMARY KEY (`MedicalPersonId`)

);

Patients Table:

CREATE TABLE `patients` (

`PatientId` int NOT NULL AUTO\_INCREMENT,

`PatientUniqueId` varchar(256) NOT NULL,

`FirstName` varchar(256) NOT NULL,

`LastName` varchar(256) NOT NULL,

`DateofBirth` date NOT NULL,

`Address` varchar(256) NOT NULL,

`Postcode` varchar(10) NOT NULL,

`Telephone` varchar(15) NOT NULL,

CHECK (LENGTH(`PatientUniqueId`) > 0),

CHECK (LENGTH(`FirstName`) > 0),

CHECK (LENGTH(`LastName`) > 0),

CHECK (LENGTH(`Address`) > 0),

CHECK (LENGTH(`Postcode`) > 0),

CHECK (LENGTH(`Telephone`) > 0),

PRIMARY KEY (`PatientId`)

);

VaccinationCentres Table:

CREATE TABLE `vaccinationcentres` (

`VaccinationCentreId` int NOT NULL AUTO\_INCREMENT,

`Name` varchar(256) NOT NULL,

`Address` varchar(256) NOT NULL,

`Postcode` varchar(10) NOT NULL,

`Telephone` varchar(15) NOT NULL,

PRIMARY KEY (`VaccinationCentreId`)

);

VaccinationTypes Table:

CREATE TABLE `vaccinationtypes` (

`VaccinationTypeId` int NOT NULL AUTO\_INCREMENT,

`Name` varchar(256) NOT NULL,

PRIMARY KEY (`VaccinationTypeId`)

);

PatientVaccinations Table:

CREATE TABLE `patientvaccinations` (

`PatientVaccinationId` int NOT NULL AUTO\_INCREMENT,

`DateTime` datetime(3) NOT NULL,

`VaccinationCentreId` int NOT NULL,

`MedicalPersonId` int NOT NULL,

`PatientId` int NOT NULL,

`VaccinationTypeId` int NOT NULL,

PRIMARY KEY (`PatientVaccinationId`),

KEY `FK\_PatientVaccinations\_MedicalPersons` (`MedicalPersonId`),

KEY `FK\_PatientVaccinations\_VaccinationCentres\_idx` (`VaccinationCentreId`),

KEY `FK\_PatientVaccinations\_Patients\_idx` (`PatientId`),

KEY `FK\_PatientVaccinations\_VaccinationTypes\_idx` (`VaccinationTypeId`),

CONSTRAINT `FK\_PatientVaccinations\_MedicalPersons` FOREIGN KEY (`MedicalPersonId`) REFERENCES `medicalpersons` (`MedicalPersonId`),

CONSTRAINT `FK\_PatientVaccinations\_Patients` FOREIGN KEY (`PatientId`) REFERENCES `patients` (`PatientId`),

CONSTRAINT `FK\_PatientVaccinations\_VaccinationCentres` FOREIGN KEY (`VaccinationCentreId`) REFERENCES `vaccinationcentres` (`VaccinationCentreId`),

CONSTRAINT `FK\_PatientVaccinations\_VaccinationTypes` FOREIGN KEY (`VaccinationTypeId`) REFERENCES `vaccinationtypes` (`VaccinationTypeId`)

);

## 02 Patients Sample Data

INSERT INTO patients (`PatientId`, `PatientUniqueId`, `FirstName`, `LastName`, `DateofBirth`, `Address`, `Postcode`, `Telephone`) VALUES

(1, N'02D246C4', N'Papillong', N'Cornwell', CAST(N'1974-01-03' AS Date), N'116 North Walk, New Addington', N'CR0 9ES', N'(01689) 582460')

, (2, N'A9A075E1', N'Uval', N'Forget', CAST(N'2008-01-03' AS Date), N'Prentice Grange, 2 Prentice Hall Lane, Tollesbury', N'CM9 8RN', N'(01621) 581216')

, (3, N'3BF2F060', N'Jacky', N'Fels', CAST(N'1992-01-03' AS Date), N'4 Little Acre, St Albans', N'AL3 5SG', N'(01727) 036208')

, (4, N'CC29BC73', N'Markus', N'Dumm', CAST(N'1942-01-03' AS Date), N'287 Rooley Lane, West Bowling', N'BD5 8JZ', N'(01274) 176847')

, (5, N'D9FD0E7B', N'Cortney', N'Dales', CAST(N'1967-01-03' AS Date), N'22 Hamilton Road, Newmarket', N'CB8 0NY', N'(01638) 412652')

, (6, N'5CE66BF9', N'Yan', N'Dozar', CAST(N'1956-01-03' AS Date), N'4 Japonica Drive, Merthyr Tydfil', N'CF48 3HG', N'(01685) 364785')

, (7, N'B3B9B976', N'Fernand', N'Mannschre', CAST(N'1987-01-03' AS Date), N'36 Main Road, Denholme', N'BD13 4DD', N'(01535) 363718')

, (8, N'7341DA04', N'Tomas', N'Wheeland', CAST(N'1983-01-03' AS Date), N'18 Melbourne Avenue, Fleetwood', N'FY7 8AY', N'(01253) 860070')

, (9, N'E14BE1BF', N'Parsifal', N'Bellemy', CAST(N'1935-01-03' AS Date), N'45 Beryl Road, Prenton', N'CH43 9RS', N'(0151) 137 4713')

, (10, N'E0BA81E1', N'Delbert', N'Mckahan', CAST(N'1938-01-03' AS Date), N'2 Roscow Avenue, Bolton', N'BL2 6HU', N'(01204) 030241')

, (11, N'1B9D8003', N'Wilder', N'Alvine', CAST(N'1972-01-03' AS Date), N'29 Warple Road, Birmingham', N'B32 1RL', N'(0121) 152 4536')

, (13, N'19783B4C', N'Hewes', N'Absher', CAST(N'1957-01-03' AS Date), N'81 Mayflower Way, Ongar', N'CM5 9BB', N'(01277) 472656')

NOTE: This continues for another 10484 rows. Total number of insert statements 10500

## Creating Patients Sample Data:

The Patients table took a first and last name so I went to this site:

<https://spottedtiger.tripod.com/D_Language/D_Randomly_Gen_Data_XP.html>

I was able to download an xml file that contained a large list of random male and female first and last name data. I needed to convert this xml into a csv to import into SQL server. I used this site to convert the xml csv:

<https://www.convertcsv.com/xml-to-csv.htm>

I migrated the csv date into a PatientSample table.

I generated a random PatientUniqueId for all patients using a Globally Unique Identifier (GUID) with the substring function to make the length of the PatientUniqueId 8 characters long.

To get the address, postcode and phone number I used a random address generator that included random phone numbers:

https://www.doogal.co.uk/RandomAddresses.php

Once I had all the data that I needed in the PatientSample table, I wrote a query to insert data from PatientSample table into the Patients table. I then generated the Patient table SQL script so I had all the insert statements so I could test the data in the table without worrying about losing it.

I then deleted the PatientSample table because I now had the same data within the Patients table meaning that I no longer needed to keep the PatientSample table.

## 03 Vaccination Types Sample Data

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (1, N'Adenovirus Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (2, N'Anthrax Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (3, N'Cholera Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (4, N'COVID-19 Janssen Vaccine');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (5, N'COVID-19 Moderna Vaccine');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (6, N'COVID-19 Pfizer BioNTech Vaccine');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (7, N'COVID-19 AstraZeneca Vaccine');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (8, N'DTaP (Diphtheria, Tetanus, Pertussis) Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (9, N'Haemophilus Influenzae type b (Hib) Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (10, N'Hepatitis A Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (11, N'Hepatitis B Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (12, N'HPV (Human Papillomavirus) Vaccine VIS');

INSERT INTO vaccinationtypes (`VaccinationTypeId`, `Name`) VALUES (13, N'Influenza (Flu) Vaccine (Inactivated or Recombinant) VIS');

NOTE: This continues for another 19 rows. Total number of insert statements 32

## Creating Vaccination Types Sample Data:

I found online a site that had Vaccine Information Statement(VIS) Lookup Table:

<https://spottedtiger.tripod.com/D_Language/D_Randomly_Gen_Data_XP.html>

This had a list of all vaccines current and historic. I downloaded the flat file version of the table and imported it into SQL server.

The data was then migrated to VaxSample table where I created a query to insert only current vaccine types.

I then migrated all the data from VaxSample table to VaccinationTypes table.

To keep the data for future testing I generated VaccinationTypes table SQL script so I always have the insert statements.

I then deleted the VaxSample table because I now had the same data within the VaccinationTypes table meaning that I no longer needed to keep the VaxSample table.

## 04 Medical Persons Sample Data

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) Values (1, N'Nonyameko', N'Kirk', N'18 Sadleir Road, St. Albans', N'CB6 2WT', N'(01900) 245563', N'Midwife');

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) VALUES (2, N'Renia', N'Cammarn', N'1 Portway Close, Solihull', N'AL7 3TJ', N'(01223) 411725', N'Physician');

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) VALUES (3, N'Thompson', N'Demara', N'3 Pembroke Gardens, Wellesbourne', N'CT4 5PG', N'(01274) 505245', N'Surgeon');

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) VALUES (4, N'Merima', N'Stonehous', N'11 Ghyll Close, Steeton', N'AL4 9DP', N'(01204) 812207', N'Nurse Anaesthetist');

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) VALUES (5, N'Gabriel', N'Lonneman', N'45 Knowle Avenue, Thornton-Cleveleys', N'CO7 6EU', N'(01282) 181235', N'Pediatrician');

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) VALUES (6, N'Ryan', N'Ferrigno', N'7 Pittmore Road, Burton', N'CH3 5NW', N'(01727) 101053', N'Medical Assistant');

INSERT INTO medicalpersons (`MedicalPersonId`, `FirstName`, `LastName`, `Address`, `Postcode`, `Telephone`, `Profession`) VALUES (7, N'Hao', N'Farnworth', N'20 Willow Trees Drive, Blackburn', N'CB10 1NR', N'(01443) 832676', N'Surgeon');

NOTE: This continues for another 18 rows. Total number of insert statements 25

## Creating Medical Persons Sample Data:

I searched online for medical professions and selected a list of 9 as each medical person has to be registered with a medical profession:

<https://www.google.com/search?q=medical+professions&rlz=1C1ONGR_en-GBGB931GB931&oq=medical+professions+&aqs=chrome..69i57j69i59.10800j0j7&sourceid=chrome&ie=UTF-8>

After this I made a loop that created 25 medical persons and each medical persons were given a random medical profession from my list of 9 that I got online. The rest of the data for the medical persons was created using the random data from the Patients table.

Then generated VaccinationTypes table SQL script to keep all the insert statements in case anything went wrong with the data.

## 05 Vaccination Centres Sample Data

INSERT INTO vaccinationcentres (`VaccinationCentreId`, `Name`, `Address`, `Postcode`, `Telephone`) VALUES (1, N'A R K Healthcare - Sparkhill', N'566-568 Stratford Road, Sparkhill, Birmingham', N'B11 4AN', N'0121 772 7381');

INSERT INTO vaccinationcentres (`VaccinationCentreId`, `Name`, `Address`, `Postcode`, `Telephone`) VALUES (2, N'Lytham Road Pharmacy - Blackpool', N'South Shore Primary Care Centre, Lytham Road, Blackpool', N'FY4 1TJ', N'01253 403038');

INSERT INTO vaccinationcentres (`VaccinationCentreId`, `Name`, `Address`, `Postcode`, `Telephone`) VALUES (3, N'Boots - Boots Colchester Lion Walk', N'5-6 Lion Walk Colchester, Essex', N'CO1 1LX', N'01206 577303');

INSERT INTO vaccinationcentres (`VaccinationCentreId`, `Name`, `Address`, `Postcode`, `Telephone`) VALUES (4, N'Rowlands Pharmacy - Shipley', N'Windhill Green Med Centre, 2 Thackley Old Road, Shipley, West Yorkshire', N'BD18 1QB', N'01274 531214');

INSERT INTO vaccinationcentres (`VaccinationCentreId`, `Name`, `Address`, `Postcode`, `Telephone`) VALUES (5, N'Avonmouth Pharmacy', N'205 Avonmouth Road, Avonmouth, Bristol', N'BS11 9EG', N'0117 982 3158');

Note: Total number of insert statements 5

## Creating Vaccination Centres Sample Data:

I got a list of vaccination centres from:

<https://www.england.nhs.uk/coronavirus/publication/vaccination-sites/>

Then took 1 vaccination centre for every 500 in Microsoft excel file. A total of 5 vaccination centres were chosen.

After this I created insert statements to insert the name, address, postcode and telephone for each of the 5 vaccination centres into VaccinationCentres table.

## 06 Patient Vaccinations Sample Data

INSERT INTO patientvaccinations (`PatientVaccinationId`, `DateTime`, `VaccinationCentreId`, `MedicalPersonId`, `PatientId`, `VaccinationTypeId`) VALUES

(1, CAST(N'2021-08-08T14:44:11.000' AS DateTime(3)), 1, 9, 1, 1)

, (2, CAST(N'2020-10-24T23:37:12.000' AS DateTime(3)), 1, 9, 2, 1)

, (3, CAST(N'2021-01-28T23:08:20.000' AS DateTime(3)), 1, 8, 3, 1)

, (4, CAST(N'2020-04-11T18:08:07.000' AS DateTime(3)), 1, 4, 4, 1)

, (5, CAST(N'2021-03-04T15:24:26.000' AS DateTime(3)), 1, 5, 5, 1)

, (6, CAST(N'2020-12-14T02:33:54.000' AS DateTime(3)), 1, 3, 6, 1)

, (7, CAST(N'2021-04-19T18:20:51.000' AS DateTime(3)), 1, 9, 7, 1)

, (8, CAST(N'2021-02-16T11:05:15.000' AS DateTime(3)), 1, 4, 8, 1)

, (9, CAST(N'2020-10-01T10:43:25.000' AS DateTime(3)), 1, 10, 9, 1)

, (10, CAST(N'2021-12-14T05:12:36.000' AS DateTime(3)), 1, 8, 10, 1)

, (11, CAST(N'2021-09-27T23:02:05.000' AS DateTime(3)), 1, 8, 11, 1)

, (12, CAST(N'2020-09-14T18:25:05.000' AS DateTime(3)), 1, 6, 12, 1)

, (13, CAST(N'2020-04-10T10:21:30.000' AS DateTime(3)), 1, 2, 13, 1)

, (14, CAST(N'2020-11-28T17:55:05.000' AS DateTime(3)), 1, 1, 14, 1)

, (15, CAST(N'2021-11-06T19:09:31.000' AS DateTime(3)), 1, 8, 15, 1)

, (16, CAST(N'2020-07-19T17:11:01.000' AS DateTime(3)), 1, 7, 16, 1)

, (17, CAST(N'2020-04-15T19:37:34.000' AS DateTime(3)), 1, 3, 17, 1)

, (18, CAST(N'2020-09-15T19:15:28.000' AS DateTime(3)), 1, 5, 18, 1)

, (19, CAST(N'2020-01-12T18:10:19.000' AS DateTime(3)), 1, 9, 19, 1)

Note: This continues for 55736 rows. Total Number of rows inserted 55755. Total number of insert statements 190 as I used bulk insert statements to make inserting data faster.

## Creating Patient Vaccinations Sample Data:

To get patient vaccinations sample data I got the list of vaccination types and the percentage they will be given.

Used a temporary table called #TempVaccinationTypes and inputted the name of the vaccination types and percentage of patients which will receive each vaccination type.

I then iterated over each vaccine type and assigned it to patients based on percentage of patients that should receive each vaccination.

Also assigned random date time to DateTime field.

I generated the PatientVaccinations table SQL script.

However, after that I decided to change the SQL query to generate random data for VaccinationCentreId and MedicalPersonId for PatientVaccinations table and took random data in multiple sections to have a wider spread of data.

I then deleted the old data for PatientVaccinations table and created new data for PatientVaccinations table from changed SQL query.

Finally, I generated the PatientVaccinations table SQL script again.

## 07 Stored Procedures

-- [spGetUsernames]

-- This will get the list if medical persons that can login

-- ---------------------------------------------------------------------

DROP procedure IF EXISTS `spGetUsernames`;

DELIMITER $$

CREATE PROCEDURE `spGetUsernames`()

BEGIN

SELECT MedicalPersonId, CONCAT(FirstName , ' ' , LastName , ' (' , Profession , ')') AS Username FROM medicalpersons ORDER BY FirstName, LastName;

END$$

DELIMITER ;

spGetUsernames Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. Selects MedicalPersonnId, concatenates FirstName & LastName & Profession asa Username so it would display as First Name Last Name (Profession), from MedicalPersons table.
2. Order by FirstName and LastName so the usernames display in alphabetical order.

-- [spGetMedicalPerson]

-- This will get the selected medical persons after login

-- ---------------------------------------------------------------------

DROP procedure IF EXISTS `spGetMedicalPerson`;

DELIMITER $$

CREATE PROCEDURE `spGetMedicalPerson`(IN p\_MedicalPersonId INT)

BEGIN

SELECT \* FROM medicalpersons mp WHERE mp.MedicalPersonId = p\_MedicalPersonId;

END$$

DELIMITER ;

spGetMedicalPerson Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. Selects all from MedicalPersons table
2. Condition is where MedicalPersonId matches incoming p\_MedicalPersonId parameter from PHP code.

-- [spGetVaccinationCentres]

-- This will get a list of vaccination centres after login

-- -------------------------------------------------------------------

DROP procedure IF EXISTS `spGetVaccinationCentres`;

DELIMITER $$

CREATE PROCEDURE `spGetVaccinationCentres`()

BEGIN

SELECT VaccinationCentreId, Name FROM vaccinationcentres ORDER BY Name;

END$$

DELIMITER ;

spGetVaccinationCentres Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. TODO

-- [spGetVaccinationTypes]

-- This will get a list of vaccination types after login

-- ----------------------------------------------------------------

DROP procedure IF EXISTS `spGetVaccinationTypes`;

DELIMITER $$

CREATE PROCEDURE `spGetVaccinationTypes`()

BEGIN

SELECT VaccinationTypeId, Name FROM vaccinationtypes ORDER BY Name;

END$$

DELIMITER ;

-- [spGetVaccinationHistory]

DROP procedure IF EXISTS `spGetVaccinationHistory`;

DELIMITER $$

CREATE PROCEDURE `spGetVaccinationHistory`(IN p\_MedicalPersonId INT)

BEGIN

CREATE TEMPORARY TABLE Temp\_VaxHistory

SELECT DATE\_FORMAT(pv.DateTime, "%d %b %Y at %h:%i %p") AS DateTime, vc.Name AS VaccinationCentre, CONCAT(p.FirstName , ' ' , p.LastName) AS PatientName, vt.Name AS VaccinationType

FROM medicalpersons mp

INNER JOIN patientvaccinations pv ON mp.MedicalPersonId = pv.MedicalPersonId

INNER JOIN vaccinationcentres vc ON pv.VaccinationCentreId = vc.VaccinationCentreId

INNER JOIN patients p ON pv.PatientId = p.PatientId

INNER JOIN vaccinationtypes vt ON pv.VaccinationTypeId = vt.VaccinationTypeId

WHERE mp.MedicalPersonId = p\_MedicalPersonId

ORDER BY pv.DateTime DESC;

SET @row\_number = 0;

SELECT \*, (@row\_number:=@row\_number + 1) AS RowNum

FROM Temp\_VaxHistory

-- delete temporary tables as they are no longer required

DROP TEMPORARY TABLE Temp\_VaxHistory;

END$$

DELIMITER ;

spGetVaccinationHistory Stored Procedure Explained

This will get a list of a vaccinators history

1. First it creates a temporary table so that a row number can be assigned to each row.
2. It then selects the date time, name of vaccination centre, first and last name concatenated together and vaccination type ordered all by pv.DateTime.
3. Sets a row number variable
4. Selects all the data from Temp\_VaxHistory temporary table and sets the row number variable to increment by 1.
5. Finally, drops the temporary table as it is no longer required

-- [spSavePatientVaccination]

DROP procedure IF EXISTS `spSavePatientVaccination`;

DELIMITER $$

CREATE PROCEDURE `spSavePatientVaccination`(

IN p\_MedicalPersonId INT,

IN p\_VaccinationCentreId INT,

IN p\_DateTime DATETIME,

IN p\_VaccinationTypeId INT,

IN p\_PatientUniqueId VARCHAR(8),

IN p\_PatientFirstName VARCHAR(256),

IN p\_PatientLastName VARCHAR(256),

IN p\_PatientDOB DATETIME,

IN p\_PatientAddress VARCHAR(256),

IN p\_PatientPostcode VARCHAR(10),

IN p\_PatientTelephone VARCHAR(15)

)

BEGIN

-- get patient from unique id

SELECT PatientId INTO @PatientId FROM patients WHERE PatientUniqueId = p\_PatientUniqueId;

IF @PatientId IS NULL THEN

-- patient does not exist so we'll insert

INSERT INTO patients(PatientUniqueId, FirstName, LastName, DateofBirth, Address, Postcode, Telephone)

VALUES(p\_PatientUniqueId, p\_PatientFirstName, p\_PatientLastName, p\_PatientDOB, p\_PatientAddress, p\_PatientPostcode, p\_PatientTelephone);

SET @PatientId := LAST\_INSERT\_ID();

ELSE

-- patient exists so we'll update

UPDATE patients

SET FirstName = p\_PatientFirstName, LastName = p\_PatientLastName, DateofBirth = p\_PatientDOB, Address = p\_PatientAddress, Postcode = p\_PatientPostcode, Telephone = p\_PatientTelephone

WHERE PatientId = @PatientId;

END IF;

-- insert patient vaccination

INSERT INTO patientvaccinations(DateTime, VaccinationCentreId, MedicalPersonId, PatientId, VaccinationTypeId)

VALUES (p\_DateTime, p\_VaccinationCentreId, p\_MedicalPersonId, @PatientId, p\_VaccinationTypeId);

END$$

DELIMITER ;

spSavePatientVaccination Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. First it selects the PatientId into @PatientId variable where the PatientUniqueId matches.
2. Then it checks if @PatientId is null and if it’s null then the patient does not exist. We have to insert the patient data into the Patients table within the database.
3. Otherwise we just update the patient details.
4. Then insert the patient vaccination data into PatientVaccinations table

-- [spGetReportVaccinationsByCentre]

-- This will get a list of vaccinations carried out by each centre

-- ------------------------------------------------------------------------------

DROP procedure IF EXISTS `spGetReportVaccinationsByCentre`;

DELIMITER $$

CREATE PROCEDURE `spGetReportVaccinationsByCentre`()

BEGIN

SELECT vc.Name, vc.Address, vc.Postcode, vc.Telephone, COUNT(pv.VaccinationCentreId) AS NumberOfVaccinations

FROM patientvaccinations pv

INNER JOIN vaccinationcentres vc ON pv.VaccinationCentreId = vc.VaccinationCentreId

GROUP BY vc.VaccinationCentreId

ORDER BY NumberOfVaccinations DESC;

END$$

DELIMITER ;

spGetReportVaccinationsByCentre Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. Gets the vaccination centre name, address, postcode, telephone and the number of vaccinations carried out by each centre by counting all the patient vaccinations VaccinationCentreId, from PatientVaccinations table inner joined on to VaccinationCentres table based on VaccinationCentreId.
2. Then it is grouped by VaccinationCentres table VaccinnationCentreId.
3. Then ordered by the NumberOfVaccinations descending.

-- [spGetReportPatientsByVaccinationType]

-- This will get a list of vaccinations carried out by each centre

-- ------------------------------------------------------------------------------

DROP procedure IF EXISTS `spGetReportPatientsByVaccinationType`;

DELIMITER $$

CREATE PROCEDURE `spGetReportPatientsByVaccinationType`()

BEGIN

SELECT COUNT(\*) INTO @TotalPatients FROM patients;

SELECT vt.Name, COUNT(pv.VaccinationTypeId) AS NumberOfPatients, COUNT(pv.VaccinationTypeId) / @TotalPatients \* 100 AS PercentOfPatients

FROM patientvaccinations pv

INNER JOIN vaccinationtypes vt ON pv.VaccinationTypeId = vt.VaccinationTypeId

WHERE pv.DateTime BETWEEN DATE("2020-01-01") AND DATE(CURDATE())

GROUP BY vt.VaccinationTypeId

ORDER BY NumberOfPatients DESC;

END$$

DELIMITER ;

spGetReportPatientsByVaccinationType Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. First it selects the total number of patients into @TotalPatients variable.
2. Then it selects the vaccination type name, the total number of vaccinations for the vaccine as the NumberOfPatients and the percentage of patients recieved the vaccination by getting the total number of vaccinations for the vaccine dividing by @TotalPatients multiplied by 100 as PercentOfPatients, from PatientVaccinations table inner joined on VaccinationTypes table based on VaccinationTypeId.
3. The condition is where the patient vaccination date time is between 01/01/20 and whatever the current date is by using the CURDATE() function.
4. Then grouped by VaccinationTypes table VaccinationTypeId
5. Then ordered by NumberOfPatients descending.

-- [spGetReportCovidVaccinationsByArea]

-- This will get the total number of covid vaccinations then will brake down the 4 different types of covid vaccines and number of those given

-- -----------------------------------------------------------------------------------------------------

DROP procedure IF EXISTS `spGetReportCovidVaccinationsByArea`;

DELIMITER $$

CREATE PROCEDURE `spGetReportCovidVaccinationsByArea`()

BEGIN

-- creates a temporary table that gets the area, vaccine type, total for each vaccicne type for each area

-- and adds a column for the total number of covid vaccinations

CREATE TEMPORARY TABLE Temp\_AreaVaxDetails

SELECT

SUBSTRING\_INDEX(p.Address, ', ', -1) AS Area,

vt.name AS Vaccine,

COUNT(\*) AS NumVaxByArea,

0 AS TotalVax

FROM patients p

INNER JOIN patientvaccinations pv ON p.PatientId = pv.PatientId

INNER JOIN vaccinationtypes vt ON pv.VaccinationTypeId = vt.VaccinationTypeId

WHERE vt.name LIKE "COVID-19%"

GROUP BY Area, vt.VaccinationTypeId;

-- created Temp\_AreaVaxQty temporary table to hold the total number of covid vaccinations for each area

-- added an index to this table to make the query run faster from 9 seconds to 0 seconds

CREATE TEMPORARY TABLE Temp\_AreaVaxQty

(INDEX area\_index (Area), Qty INT)

SELECT Area, SUM(NumVaxByArea) AS Qty FROM Temp\_AreaVaxDetails GROUP BY Area;

-- updates values in Temp\_AreaVaxDetails table using the values from Temp\_AreaVaxQty table making it fast

-- as it doesn't need to query to get the values it needs beacuse of the Temp\_AreaVaxQty table having the data already

UPDATE Temp\_AreaVaxDetails t1

INNER JOIN Temp\_AreaVaxQty t2 ON t1.Area = t2.Area

SET t1.TotalVax = t2.Qty;

SELECT \* FROM Temp\_AreaVaxDetails ORDER BY TotalVax DESC, Area, NumVaxByArea DESC, Vaccine;

-- delete temporary tables as they are no longer required

DROP TEMPORARY TABLE Temp\_AreaVaxDetails;

DROP TEMPORARY TABLE Temp\_AreaVaxQty;

END$$

DELIMITER ;

spGetReportCovidVaccinationsByArea Stored Procedure Explained

This will get the total number of covid vaccinations then will break down the 4 different types of covid vaccines and number of those given

1. First it creates a temporary table called Temp\_AreaVaxDetails
2. This temporary table gets the area, vaccine type and adds a new column to store the total for each vaccine type for each area for only covid vaccines
3. Then another temporary table called Temp\_AreaVaxQty is created to hold the data from Temp\_AreaVaxDetails.
4. I also added an index to Temp\_AreaVaxQty temporary table because the query was taking 9 seconds to run and to reduce this to 0 seconds I implemented an index which means all the data is easily accessible which makes it fast and efficient
5. It then updates Temp\_AreaVaxDetails with Temp\_AreaVaxQty using an update join so make it fast as it does not need to query to get the data due to Temp\_AreaVaxQty table already holding the data.
6. Selects all from Temp\_AreaVaxDetails ordering the data by TotalVax desc, Area, NumVaxByArea desc, Vaccine.
7. Deletes both temporary tables as they are no longer required.

-- [spGetTotalPatients]

-- This will get total number of patients

-- --------------------------------------

DROP procedure IF EXISTS `spGetTotalPatients`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetTotalPatients`()

BEGIN

SELECT COUNT(\*) AS TotalPatients FROM Patients;

END$$

DELIMITER ;

spGetTotalPatients Stored Procedure Explained

This will insert or update a patient and record the vaccination

* Counts all as TotalPatients from Patients table to get the total number of patients within Patients table.

-- [spGetTotalVaccinations]

-- This will get total number of vaccinations

-- ------------------------------------------

DROP procedure IF EXISTS `spGetTotalVaccinations`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetTotalVaccinations`()

BEGIN

SELECT COUNT(\*) AS TotalVaccinations FROM PatientVaccinations;

END$$

DELIMITER ;

spGetTotalVaccinations Stored Procedure Explained

This will insert or update a patient and record the vaccination

* Counts all as TotalVaccinations from PatientVaccinations table to get the total number of vaccinations within PatientVaccinations table.

-- [spGetTotalCovidVaccinations]

-- This will get total number of covid vaccinations

-- ------------------------------------------------

DROP procedure IF EXISTS `spGetTotalCovidVaccinations`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetTotalCovidVaccinations`()

BEGIN

SELECT COUNT(\*) AS TotalCovidVaccinations

FROM PatientVaccinations pv

INNER JOIN VaccinationTypes vt on pv.VaccinationTypeId = vt.VaccinationTypeId

WHERE vt.name LIKE "COVID-19%";

END$$

DELIMITER ;

spGetTotalCovidVaccinations Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. Counts all as TotalCovidVaccinations from PatientVaccinations table inner joined on to VaccinationTypes table based on VaccinationTypeId to get the total number of covid vaccinations.
2. Condition is where the vaccination type name is like “COVID-19%”, so any vaccination type with that in it is counted as a covid vaccination type.

-- [spGetTotalMedicalPersonVaccinations]

-- This will get total number of vaccinations for each medical person

-- ------------------------------------------------------------------

DROP procedure IF EXISTS `spGetTotalMedicalPersonVaccinations`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetTotalMedicalPersonVaccinations`(IN p\_MedicalPersonId INT)

BEGIN

SELECT COUNT(\*) AS TotalMedicalPersonVaccinations

FROM medicalpersons mp

INNER JOIN patientvaccinations pv ON mp.MedicalPersonId = pv.MedicalPersonId

WHERE mp.MedicalPersonId = p\_MedicalPersonId;

END$$

DELIMITER ;

spGetTotalMedicalPersonVaccinations Stored Procedure Explained

This will insert or update a patient and record the vaccination

1. Counts all as TotalMedicalPersonVaccinations from MedicalPersons table inner joined on to PatientVaccinationsTypes table based on MedicalPersonId.
2. Condition is where MedialPersonId matches the MedicalPersonId parameter coming in from the PHP code.

## PHP Code

For the web integration I used the include statement. This allows for inserting the content of one php file into another php file. So for each of these files Login.php, AminReports.php, VaccinatePatient.php, VaccinationHistory.php I set a $page\_header variable for the title of the page and $page\_content for the path of the content related file for example Login.php the content file would be Login\_Content.php. At the end of each of these files I included the Master.php file which would display the $page\_header value in a h1 element and underneath include the $page\_content file by using the set path in $page\_content variable.

Index.php

<?php

// When loaded it will always redirect to login page

echo "<script> location.href='/Login/Login.php'; </script>";

exit;

?>

master.php

<?php

session\_start();

// Stops user from going to the previous page while on login page and no login session found

if ($\_SESSION["medicalPersonId"] == null && basename($\_SERVER['PHP\_SELF']) != "Login.php") {

header('Location: ../Login/Login.php');

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset='utf-8'>

<meta http-equiv='X-UA-Compatible' content='IE=edge'>

<title>Vaccination Tracker</title>

<meta name='viewport' content='width=device-width, initial-scale=1'>

<link rel='stylesheet' type='text/css' media='screen'

href='/MasterPages/Master.css'>

<!-- External CSS -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.15.4/css/all.min.css" integrity="sha512- 1ycn6IcaQQ40/MKBW2W4Rhis/DbILU74C1vSrLJxCq57o941Ym01SwNsOMqvEBFlcgUa6xLiPY/NS5 R+E6ztJQ==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<link [href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css](about:blank)" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<!-- PHP Code Files -->

<?php include("DatabaseHelpers.php"); ?>

</head>

<body>

<div class="container">

<!-- Checks to see if the log out button should be shown -->

<?php if (basename($\_SERVER['PHP\_SELF']) != "Login.php") { ?>

<div style="float: right; margin-top: 10px;">

<a href="/Login/Logout.php" class="btn btn-danger"><i class="fas fa-sign-out-alt"></i> Log Out</a>

</div>

<?php } ?>

<h1 style="margin-bottom: 40px;"><?php echo $page\_header ?></h1>

<?php include($page\_content); ?>

<footer class="d-flex flex-wrap justify-content-between align-items- center py-3 my-4 border-top">

<div class="col-md-4 d-flex align-items-center">

<!-- Displays 2022 to the current year in footer -->

<span class="text-muted">© Bilal Asghar 2022 - <?php echo date("Y"); ?></span>

</div>

</footer>

</div>

<!-- External Javascript -->

<script [src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js](about:blank)" integrity="sha384- ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

</body>

</html>

Master.css

.padBottom10{

padding-bottom: 10px;

}

.padBottom30{

padding-bottom: 30px;

}

DatabaseHelpers.php

<?php

// This function creates the connection to database

function OpenConnection()

{

$server = "localhost";

$username = "root";

$password = "";

$database = "Vaccinations";

$conn = new mysqli($server, $username, $password, $database);

if ($conn === false) {

die("ERROR: Could not connect. " . $conn->connect\_error);

}

return $conn;

}

// This function queries to database

function CallDatabase($sql, $isDataReturned)

{

try {

$conn = OpenConnection();

if ($isDataReturned) {

$result = mysqli\_query($conn, $sql);

if ($result == false) {

$\_SESSION["savedPatientVax"] = "failed";

} else {

$\_SESSION["SavedPatientVax"] = "success";

}

// we have data so store in variable then return

while ($row = mysqli\_fetch\_array($result)) {

$rows[] = $row;

}

mysqli\_free\_result($result); // free memory associated with result

$conn->close();

return $rows;

} else {

$result = mysqli\_query($conn, $sql);

if ($result == false) {

$\_SESSION["savedPatientVax"] = "failed";

} else {

$\_SESSION["savedPatientVax"] = "success";

}

$conn->close();

}

} catch (Exception $e) {

$conn->close();

echo "Error!" . $e->getMessage();

}

}

// This function calls spGetUsernames database stored procedure with a value of true for $isDataReturned as data is being returned

function GetUsernames()

{

return CallDatabase("call spGetUsernames()", true);

}

// This function calls spGetMedicalPerson database stored procedure with a value of true for $isDataReturned as data is being returned

function GetMedicalPerson($p\_MedicalPersonId)

{

return CallDatabase("call spGetMedicalPerson('$p\_MedicalPersonId')", true);

}

// This function calls spGetVaccinationCentres database stored procedure with a value of true for $isDataReturned as data is being returned

function GetVaccinationCentres()

{

return CallDatabase("call spGetVaccinationCentres()", true);

}

// This function calls spGetVaccinationTypes database stored procedure with a value of true for $isDataReturned as data is being returned

function GetVaccinationTypes()

{

return CallDatabase("call spGetVaccinationTypes()", true);

}

// This function calls spGetVaccinationHistory database stored procedure with a value of true for $isDataReturned as data is being returned

function GetVaccinationHistory($p\_MedicalPersonId)

{

return CallDatabase("call spGetVaccinationHistory('$p\_MedicalPersonId')", true);

}

// This function calls spSavePatientVaccination database stored procedure with a value of false for $isDataReturned as data is being saved to the database and nothing being returned

function SavePatientVaccination(

$p\_MedicalPersonId,

$p\_VaccinationCentreId,

$p\_DateTime,

$p\_VaccinationTypeId,

$p\_PatientUniqueId,

$p\_PatientFirstName,

$p\_PatientLastName,

$p\_PatientDOB,

$p\_PatientAddress,

$p\_PatientPostcode,

$p\_PatientTelephone

) {

return CallDatabase("call spSavePatientVaccination('$p\_MedicalPersonId', '$p\_VaccinationCentreId', '$p\_DateTime', '$p\_VaccinationTypeId',

'$p\_PatientUniqueId', '$p\_PatientFirstName', '$p\_PatientLastName', '$p\_PatientDOB', '$p\_PatientAddress',

'$p\_PatientPostcode', '$p\_PatientTelephone')", false);

}

// This function calls spGetReportVaccinationsByCentre database stored procedure with a value of true for $isDataReturned as data is being returned

function GetReportVaccinationsByCentre()

{

return CallDatabase("call spGetReportVaccinationsByCentre()", true);

}

// This function calls spGetReportPatientsByVaccinationType database stored procedure with a value of true for $isDataReturned as data is being returned

function GetReportPatientsByVaccinationType()

{

return CallDatabase("call spGetReportPatientsByVaccinationType()", true);

}

// This function calls spGetReportCovidVaccinationByArea database stored procedure with a value of true for $isDataReturned as data is being returned

function GetReportCovidVaccinationsByArea()

{

return CallDatabase("call spGetReportCovidVaccinationsByArea()", true);

}

// This function calls spGetTotalPatients database stored procedure with a value of true for $isDataReturned as data is being returned

function GetTotalPatients()

{

$total = CallDatabase("call spGetTotalPatients()", true);

return $total[0]['TotalPatients'];

}

// This function calls spGetTotalVaccinations database stored procedure with a value of true for $isDataReturned as data is being returned

function GetTotalVaccinations()

{

$total = CallDatabase("call spGetTotalVaccinations()", true);

return $total[0]['TotalVaccinations'];

}

// This function calls spGetTotalCovidVaccinations database stored procedure with a value of true for $isDataReturned as data is being returned

function GetTotalCovidVaccinations()

{

$total = CallDatabase("call spGetTotalCovidVaccinations()", true);

return $total[0]['TotalCovidVaccinations'];

}

// This function calls spGetTotalMedicalPersonVaccinations database stored procedure with a value of true for $isDataReturned as data is being returned

function GetTotalMedicalPersonVaccinations($p\_MedicalPersonId)

{

$total = CallDatabase("call spGetTotalMedicalPersonVaccinations('$p\_MedicalPersonId')", true);

return $total[0]['TotalMedicalPersonVaccinations'];

}

Login.php

<?php

$page\_header = '';

$page\_content = 'Login\_Content.php';

include('../MasterPages/Master.php');

?>

Login\_Content.php

<?php

// Gets all usernames by calling database function which runs stored procedure

$usernames = GetUsernames();

?>

<div class="row">

<div class="col-md-6 mx-auto">

<div class="card">

<div class="card-header">

<h3><i class="fas fa-sign-in-alt"></i> Login</h3>

</div>

<div class="card-body">

<form action='Login\_Authenticate.php' method='POST'>

<div class="form-group">

<label for="username">Username</label>

<select name="txtUsername" class="form-select" aria-label="select user">

<option selected disabled="disabled">-- Select Administrator --</option>

<option value="DemoAdministrator">Demo Administrator</option>

<option disabled="disabled"></option>

<option disabled="disabled">-- Select Vaccinator --</option>

<!-- Loops through all usernames and displays in drop down field -->

<?php foreach ($usernames as $username) { ?>

<option value="<?= $username['MedicalPersonId'] ?>"><?= $username['Username'] ?></option>

<?php } ?>

</select>

</div>

<div class="form-group" style="margin-top: 10px;">

<label for="username">Password</label>

<input type="password" name="txtPassword" class="form-control" disabled value="\*\*\*\*\*\*\*">

</div>

<div class="form-group" style="margin-top: 20px;">

<button type="submit" class="btn btn-primary btn-block"><i class="fas fa-sign-in-alt"></i> Login</button>

</div>

</form>

</div>

</div>

</div>

</div>

Login\_Authenticate.php

<?php

session\_start();

$username = $\_POST["txtUsername"];

$\_SESSION["medicalPersonId"] = $username;

// Direct user to Administrator Reports Page

if ($username == "DemoAdministrator"){

header("Location: ../AdminReports/AdminReports.php");

}

// Direct user to Vaccinate Patient Page

elseif($username != null) {

header("Location: ../PatientVaccinations/VaccinatePatient.php");

}

// Direct user to back to login page as no valid login

else {

header("Location: ../Login/Login.php");

}

die();

?>

Logout.php

<!-- Gets rid of all session variables and directs user to login page -->

<?php

session\_start();

session\_unset();

session\_destroy();

header('Location: Login.php');

exit;

?>

AdminReports.php

<?php

$page\_header = 'Administrator Reports';

$page\_content = 'AdminReports\_Content.php';

include('../MasterPages/Master.php');

?>

AdminReports\_Content.php

<div class="row padBottom30">

<div class="col-sm">

<div class="card">

<div class="card-body">

<h5 class="card-title">1. Covid Vaccinations by Area</h5>

<p class="card-text">Displays the total number of covid vaccinations carried out by area.</p>

<a href="?ReportType=1" class="btn btn-primary"><i class="fas fa-play"></i> Run Report</a>

</div>

</div>

</div>

<div class="col-sm">

<div class="card">

<div class="card-body">

<h5 class="card-title">2. Vaccination Type Report</h5>

<p class="card-text">A report on the number of patients for each type of vaccination.</p>

<div class="text-muted">

<!-- Date function gets day/month/year -->

Uses data from 01/01/2020 - <?= date("d/m/Y") ?>

</div>

<a href="?ReportType=2" class="btn btn-primary"><i class="fas fa-play"></i> Run Report</a>

</div>

</div>

</div>

<div class="col-sm">

<div class="card">

<div class="card-body">

<h5 class="card-title">3. Vaccinations by Centre</h5>

<p class="card-text">Displays the number of vaccinations carried out by each centre.</p>

<a href="?ReportType=3" class="btn btn-primary"><i class="fas fa-play"></i> Run Report</a>

</div>

</div>

</div>

</div>

<?php

$reportType = null;

// Checks if ReportType query string is set and assigns it to $reportType variable

if(isset($\_GET['ReportType'])){

$reportType = $\_GET['ReportType'];

}

// Switch statement checks query string and what report it should display

switch ($reportType) {

case "1":

include("ReportType\_1.php");

break;

case "2":

include("ReportType\_2.php");

break;

case "3":

include("ReportType\_3.php");

break;

}

?>

ReportType\_1.php

<?php

// Gets total covid vaccinations by calling database function which runs stored procedure to get data from database

$totalCovidVaccinations = GetTotalCovidVaccinations();

// Gets report data by calling database function which runs stored procedure to get data from database

$reportData = GetReportCovidVaccinationsByArea();

?>

<div class="alert alert-primary" role="alert" style="font-weight: bold;">

<!-- Uses $totalCovidVaccinations variable to display total number of covid vaccinations -->

<h5>1. Covid Vaccinations by Area</h5>

Total Number of Covid Vaccinations: <?= $totalCovidVaccinations ?>

</div>

<table class="table table-striped table-hover">

<colgroup>

<col style="width: 30%;" />

<col style="width: 20%;" />

<col style="width: 50%;" />

</colgroup>

<thead>

<tr>

<th scope="col">Area</th>

<th scope="col">Total Administered</th>

<th scope="col">Vaccine Breakdown</th>

</tr>

</thead>

<tbody>

<?php

$previous = "";

$currentArea = "";

$totalItems = count($reportData);

<!-- Loops through all items within report data -->

for ($i = 0; $i < $totalItems; $i++) {

$item = $reportData[$i];

$currentArea = $item['Area']; ?>

<!-- Checks if current area is not the same as previous area -->

<?php if ($currentArea != $previous) { ?>

// Creates new row in table for next area

<tr>

<?php $previous = $currentArea; ?>

<td>

<span style="font-size: larger;">

<?= $item['Area'] ?>

</span>

</td>

<td>

<span style="font-size: larger;">

<?= $item['TotalVax'] ?>

</span>

</td>

<td>

<?php $p = $i;

// Gets all the vaccines and the number carried out by the area and displays in <td> element

while ($currentArea ==

$reportData[$p]['Area']) { ?>

<span style="font-size:

larger;">

<?= $reportData[$p]['Vaccine'] ?>: <?= $reportData[$p]['NumVaxByArea'] ?>

</span>

<?php $p++; ?>

<br />

<?php } ?>

</td>

</tr>

<?php }

} ?>

</tbody>

</table>

ReportType\_2.php

<?php

// Gets total number of patients by calling database function which runs stored procedure to get total number from database

$totalPatients = GetTotalPatients();

// Gets report data by calling database function which runs stored procedure to get data from database

$reportData = GetReportPatientsByVaccinationType();

?>

<div class="alert alert-primary" role="alert" style="font-weight: bold;">

<!-- Uses $totalPatients variable to display total number of patients -->

<h5>2. Vaccination Type Report</h5>

Total Number of Patients: <?= $totalPatients ?>

</div>

<table class="table table-striped table-hover">

<colgroup></colgroup>

<thead>

<tr>

<th scope="col">Vaccination Type</th>

<th scope="col">Number of Patients</th>

<th scope="col">% of Total Patients</th>

</tr>

</thead>

<tbody>

<!-- Loops through all items within report data and displays the Vaccine Name, Number of patients received the vaccine type and Percentage of total patients -->

<?php foreach ($reportData as $i) { ?>

<tr>

<td>

<span style="font-size: larger;">

<?= $i['Name'] ?>

</span>

</td>

<td>

<span style="font-size: larger;">

<?= $i['NumberOfPatients'] ?>

</span>

</td>

<td>

<span style="font-size: larger;">

// Formats the PercentOfPatients as a percentage and rounds to whole number

<?= number\_format((float)$i['PercentOfPatients'], 0, '.', '') ?>%</span>

</td>

</tr>

<?php } ?>

</tbody>

</table>

ReportType\_3.php

<?php

// Gets total number of vaccinations carried out by each centre by calling database function which runs stored procedure to get total number from database

$totalVaccinations = GetTotalVaccinations();

// Gets report data by calling database function which runs stored procedure to get data from database

$reportData = GetReportVaccinationsByCentre();

?>

<div class="alert alert-primary" role="alert" style="font-weight: bold;">

<!-- Uses $totalVaccinations variable to display total number of vaccinations carried out -->

<h5>3. Vaccinations by Centre</h5>

Total Number of Vaccinations: <?= $totalVaccinations ?>

</div>

<table class="table table-striped table-hover">

<colgroup></colgroup>

<thead>

<tr>

<th scope="col">Vaccination Centre</th>

<th scope="col">Number of Vaccinations</th>

</tr>

</thead>

<tbody>

<!-- Loops through all items within report data and displays the Centre Name, Address, Postcode, Telephone and Number of vaccinations carried out by the centre -->

<?php foreach ($reportData as $i) { ?>

<tr>

<td>

<span style="font-size: larger;"><b>

<i class="fas fa-clinic-medical fa-fw"></i> <?= $i['Name'] ?></b>

</span>

<br />

<span class="text-muted"> <i class="fas fa-map-marker-alt fa-fw"></i> <?= $i['Address'] ?>

</span>

<br />

<span class="text-muted">

<i class="fas fa-map-pin fa-fw"></i> <?= $i['Postcode'] ?>

</span>

<br />

<span class="text-muted">

<i class="fas fa-phone-alt fa-fw"></i> <?= $i['Telephone'] ?>

</span>

</td>

<td>

<span style="font-size: larger;">

<b>

<?= $i['NumberOfVaccinations'] ?> </b>

</span>

</td>

</tr>

<?php } ?>

</tbody>

</table>

VaccinatePatient.php

<?php

$page\_header = 'Vaccinate Patient';

$page\_content = 'VaccinatePatient\_Content.php';

include('../MasterPages/Master.php');

?>

VaccinatePatient\_Content.php

<?php

// Gets medical person name and profession by calling database function which runs stored procedure

$medicalPerson = GetMedicalPerson($\_SESSION['medicalPersonId'])[0];

// Gets all vaccination centres by calling database function which runs stored procedure

$vaccinationCentres = GetVaccinationCentres();

// Gets all vaccination types by calling database function which runs stored procedure

$vaccinationTypes = GetVaccinationTypes();

// Sets the current date and time

$currentDateTime = date("Y-m-d\TH:i");

?>

// Checks if query string is success then display success message and if it is failed then display error message

<?php if ($\_GET['SavedPatient'] == "success") { ?>

<div class="alert alert-success" role="alert">

<i class="fas fa-check fa-lg"></i>

Successfully saved patient vaccination.

</div>

<?php } elseif ($\_GET['SavedPatient'] == "failed") { ?>

<div class="alert alert-danger" role="alert">

<i class="fas fa-times fa-lg"></i>

Failed to save patient vaccination, pleast try again.

</div>

<?php } ?>

<div class="row">

<div class="col-sm-12">

// Displays medical person first name, last name, and profession using $medicalPerson variable

<h3><i class="fas fa-user"></i>

<?= $medicalPerson['FirstName'] . ' ' . $medicalPerson['LastName'] . ' - ' .

$medicalPerson['Profession']; ?>

</h3>

// Displays medical person address, postcode, and telephone using $medicalPerson variable

<div class="text-muted"><?=

$medicalPerson['Address'] . ', ' .

$medicalPerson['Postcode'] . ', Tel:' .

$medicalPerson['Telephone']; ?>

</div>

</div>

</div>

<div class="row padBottom30">

<div class="col-sm-12" style="text-align: right;">

<a href="/PatientVaccinations/VaccinationHistory.php" class="btn btn-success"><i class="fas fa-book-medical"></i> Vaccination History</a>

</div>

</div>

<form action="VaccinatePatient\_Save.php" method="post">

<!-- Choose Vaccination Centre -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="ddlVaccinationCentre"><i class="fas fa-clinic-medical fa-fw"></i> Vaccination Centre:</label>

</div>

<div class="col-md-9">

<select name="ddlVaccinationCentre" class="form-select" aria-label="select vaccination centre" required>

<option value="">-- Select Vaccination Centre --</option>

<!-- loops through vaccination centres and displays in drop down field -->

<?php foreach ($vaccinationCentres as $vaccinationCentre) { ?>

<option value="<?= $vaccinationCentre['VaccinationCentreId'] ?>"><?= $vaccinationCentre['Name'] ?></option>

<?php } ?>

</select>

</div>

</div>

<!-- Date -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtDateTime"><i class="fas fa-calendar-day fa-fw"></i> Date:</label>

</div>

<div class="col-md-9">

<!-- Sets input to use current date & time values using $currentDateTime -->

<input type="datetime-local" class="form-control" name="txtDateTime" value="<?= $currentDateTime; ?>" required>

</div>

</div>

<!-- Vaccination Type -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="ddlVaccinationType"><i class="fas fa-syringe"></i> Vaccination Type:</label>

</div>

<div class="col-md-9">

<select name="ddlVaccinationType" class="form-select" aria-label="select user" required>

<option value="">-- Select Vaccination Type --</option>

<!-- loops through vaccination types and displays in drop down field -->

<?php foreach ($vaccinationTypes as $vaccinationType) { ?>

<option value="<?= $vaccinationType['VaccinationTypeId'] ?>"><?= $vaccinationType['Name'] ?></option>

<?php } ?>

</select>

</div>

</div>

<hr />

<!-- Patient Identifier -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientUniqueId"><i class="fas fa-id-card-alt fa-fw"></i> Patient Identifier:</label>

</div>

<div class="col-md-9">

<input type="text" class="form-control" name="txtPatientUniqueId" placeholder="Enter Unique Patient Identifier" maxlength="256" required>

</div>

</div>

<!-- Patient First Name -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientFirstName"><i class="fas fa-user fa-fw"></i> Patient First Name:</label>

</div>

<div class="col-md-9">

<input type="text" class="form-control" name="txtPatientFirstName" placeholder="Enter Patient First Name" maxlength="256" required>

</div>

</div>

<!-- Patient Last Name -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientLastName"><i class="fas fa-user fa-fw"></i> Patient Last Name:</label>

</div>

<div class="col-md-9">

<input type="text" class="form-control" name="txtPatientLastName" placeholder="Enter Patient Last Name" maxlength="256" required>

</div>

</div>

<!-- Date of Birth -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientDOB"><i class="fas fa-calendar-day fa-fw"></i>Patient Date of Birth:</label>

</div>

<div class="col-md-9">

<input type="date" class="form-control" name="txtPatientDOB" required>

</div>

</div>

<!-- Patient Address -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientAddress"><i class="fas fa-map-marker-alt fa-fw"></i> Patient Address</label>

</div>

<div class="col-md-9">

<input type="text" class="form-control" name="txtPatientAddress" placeholder="Enter Patient Address" maxlength="256" required>

</div>

</div>

<!-- Patient Postcode -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientPostcode"><i class="fas fa-map-pin fa-fw"></i> Patient Postcode</label>

</div>

<div class="col-md-9">

<input type="text" class="form-control" name="txtPatientPostcode" placeholder="Enter Patient Postcode" maxlength="10" required>

</div>

</div>

<!-- Patient Phone Number -->

<div class="row padBottom10">

<div class="col-md-3">

<label for="txtPatientTelephone"><i class="fas fa-phone-alt fa-fw"></i> Patient Phone Number</label>

</div>

<div class="col-md-9">

<input type="tel" class="form-control" name="txtPatientTelephone" placeholder="Enter Patient Phone Number" maxlength="15" required>

</div>

</div>

<!-- Add Patient Button -->

<div class="row padBottom10" style="margin-top: 20px;">

<div class="col-md-3">

<button type="submit" class="btn btn-primary mb-3"><i class="fas fa-save"></i> Save Patient Vaccination</button>

</div>

</div>

</form>

VaccinatePatient\_Save.php

<?php

session\_start();

$p\_MedicalPersonId = $\_SESSION["medicalPersonId"];

// Gets data from form and assign to variables

$p\_VaccinationCentreId = $\_POST["ddlVaccinationCentre"];

$p\_DateTime = $\_POST["txtDateTime"];

$p\_VaccinationTypeId = $\_POST["ddlVaccinationType"];

$p\_PatientUniqueId = $\_POST["txtPatientUniqueId"];

$p\_PatientFirstName = $\_POST["txtPatientFirstName"];

$p\_PatientLastName = $\_POST["txtPatientLastName"];

$p\_PatientDOB = $\_POST["txtPatientDOB"];

$p\_PatientAddress = $\_POST["txtPatientAddress"];

$p\_PatientPostcode = $\_POST["txtPatientPostcode"];

$p\_PatientTelephone = $\_POST["txtPatientTelephone"];

// Need to include Database Helpers to access its functions

include("../MasterPages/DatabaseHelpers.php");

// Calls function to save patient vaccination into database

SavePatientVaccination($p\_MedicalPersonId, $p\_VaccinationCentreId, $p\_DateTime, $p\_VaccinationTypeId, $p\_PatientUniqueId, $p\_PatientFirstName, $p\_PatientLastName, $p\_PatientDOB, $p\_PatientAddress, $p\_PatientPostcode, $p\_PatientTelephone);

// Direct user back to Vaccinate Patient Page with query string to see if saving data was a success or failed

header("Location: VaccinatePatient.php?SavedPatient=" . $\_SESSION["savedPatientVax"]);

?>

VaccinationHistory.php

<?php

$page\_header = 'Vaccination History';

$page\_content = 'VaccinationHistory\_Content.php';

include('../MasterPages/Master.php');

?>

VaccinationHistory.php

<?php

// Gets medical person name and profession by calling database function which runs stored procedure

$medicalPerson = GetMedicalPerson($\_SESSION['medicalPersonId'])[0];

// Gets total number of medical person vaccinations by calling database function which runs stored procedure

$totalMedicalPersonVaccinations = GetTotalMedicalPersonVaccinations($\_SESSION['medicalPersonId']);

// Gets vaccination history by calling database function which runs stored procedure

$vaccinationHistory = GetVaccinationHistory($\_SESSION['medicalPersonId']);

?>

<div class="row">

<div class="col-sm-12">

<!-- Displays medical person first name, last name, and profession using $medicalPerson variable -->

<h3><i class="fas fa-user"></i> <?= $medicalPerson['FirstName'] . ' ' . $medicalPerson['LastName'] . ' - ' . $medicalPerson['Profession']; ?></h3>

<!-- Displays medical person address, postcode, and telephone using $medicalPerson variable -->

<div class="text-muted"><?= $medicalPerson['Address'] . ', ' . $medicalPerson['Postcode'] . ', Tel:' . $medicalPerson['Telephone']; ?></div>

</div>

</div>

<div class="row padBottom30">

<div class="col-sm-12" style="text-align: right;">

<a href="VaccinatePatient.php" class="btn btn-secondary"><i class="fas fa-arrow-left"></i> Back</a>

</div>

</div>

<div class="alert alert-primary" role="alert" style="font-weight: bold;">

<!-- Uses $totalMedicalPersonVaccinations variable to display total number of medical person vaccinations carried out -->

<h5>Vaccination History</h5>

Total Number of Vaccinations: <?= $totalMedicalPersonVaccinations ?>

</div>

<table class="table table-striped table-hover">

<colgroup></colgroup>

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Date & Time</th>

<th scope="col">Vaccination Centre</th>

<th scope="col">Patient Name</th>

<th scope="col">Vaccination Type</th>

</tr>

</thead>

<tbody>

<!-- Loops through all items within report data and displays the Row Number, DateTime, VaccinationCentre, PatientName and VaccinationType for currently logged in medical person -->

<?php foreach ($vaccinationHistory as $vh) { ?>

<tr>

<td><?= $vh['RowNum'] ?></td>

<td><?= $vh['DateTime'] ?></td>

<td><?= $vh['VaccinationCentre'] ?></td>

<td><?= $vh['PatientName'] ?></td>

<td><?= $vh['VaccinationType'] ?></td>

</tr>

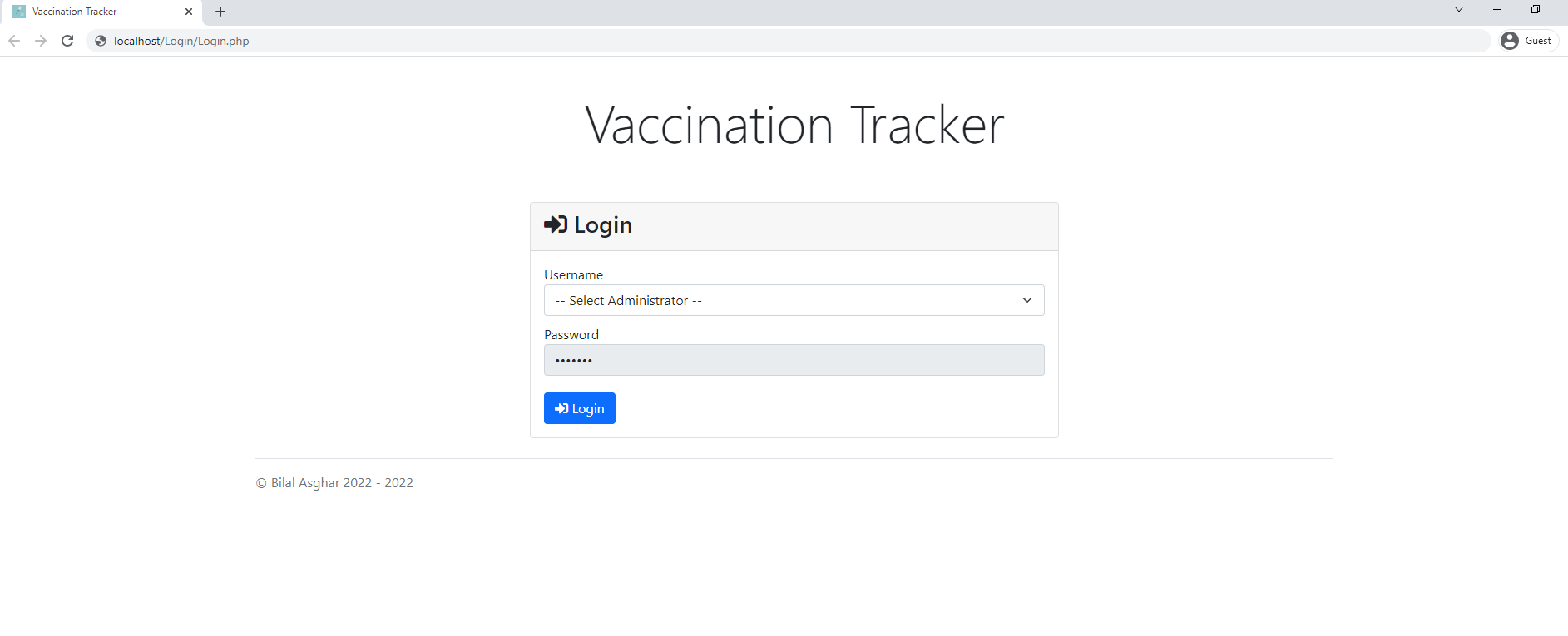
<?php } ?>

</tbody>

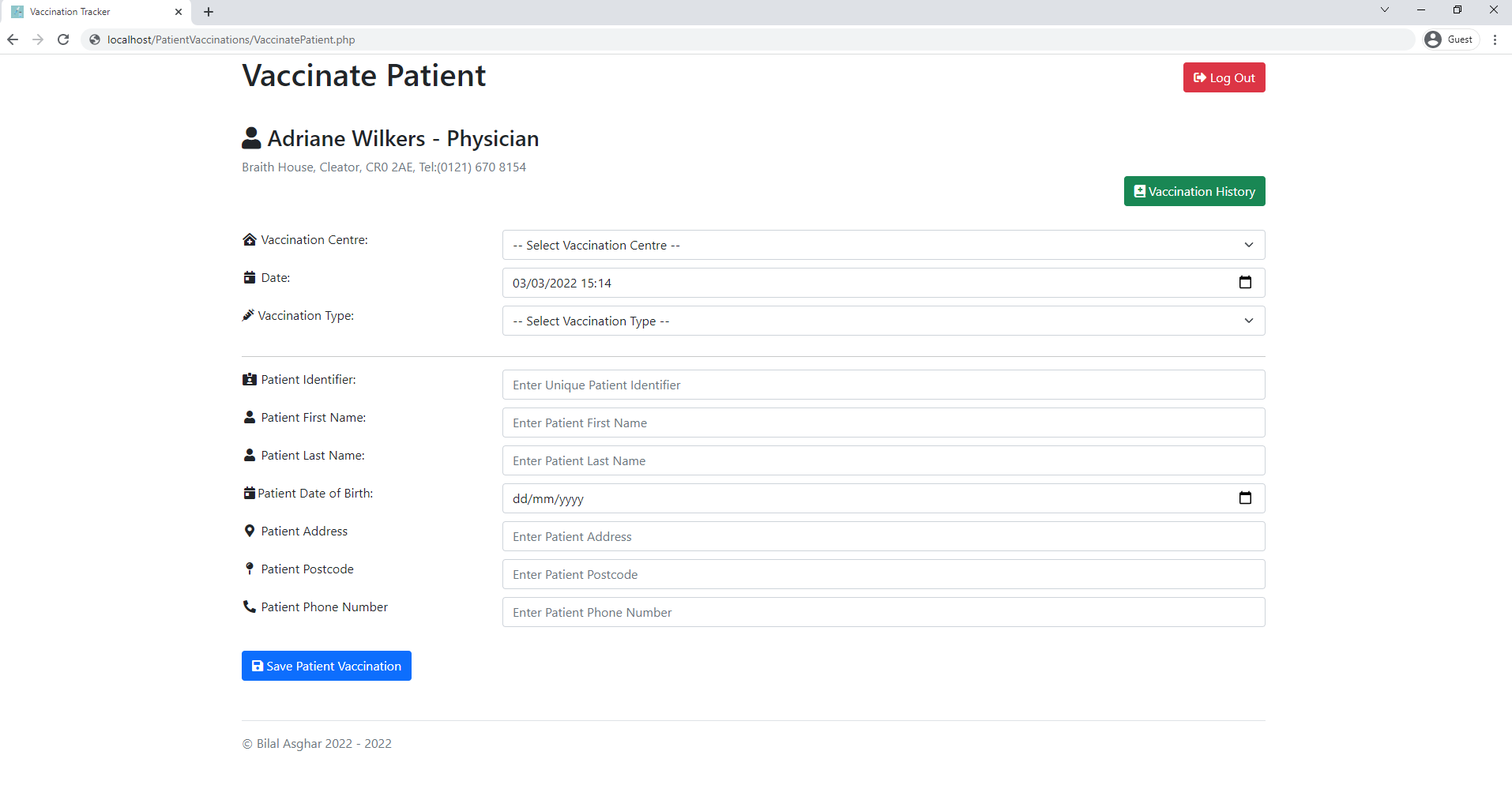
</table>

Web Design and Development Screenshots

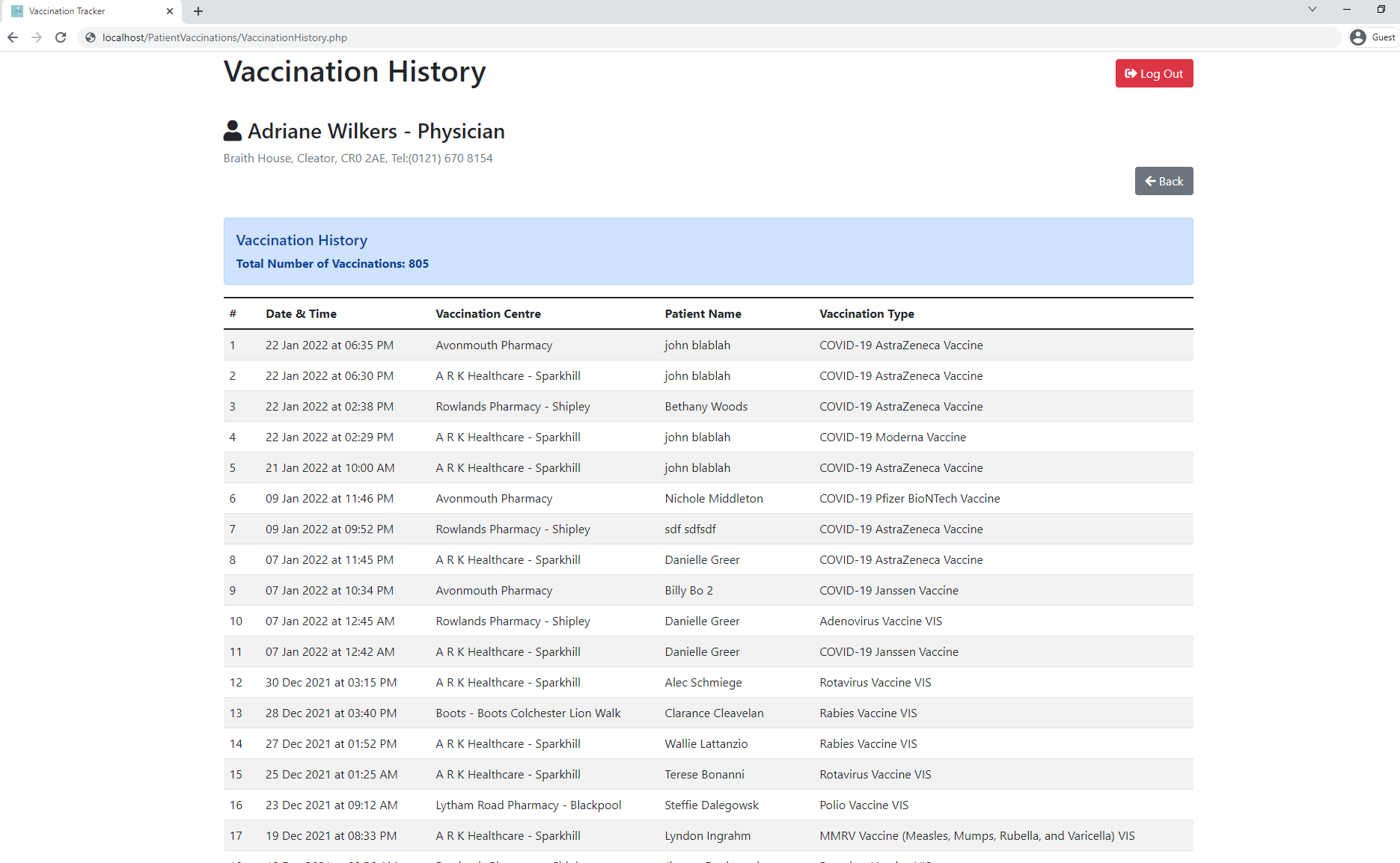
Login Page

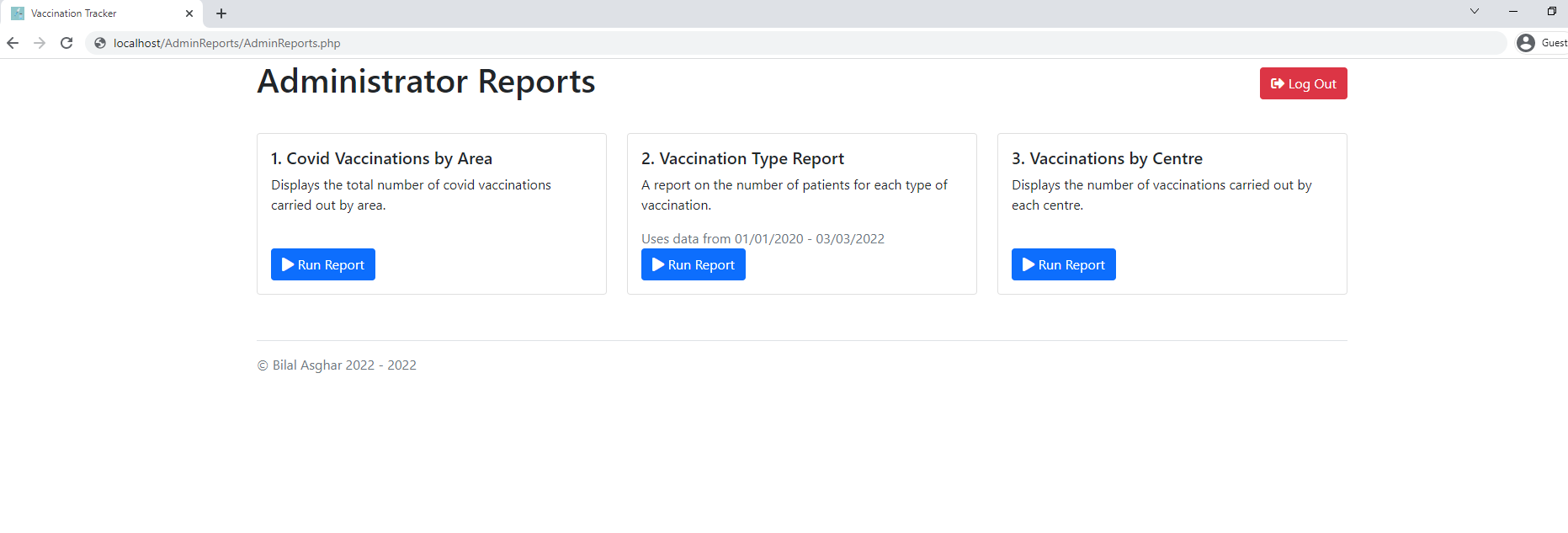


Vaccinate Patient Page

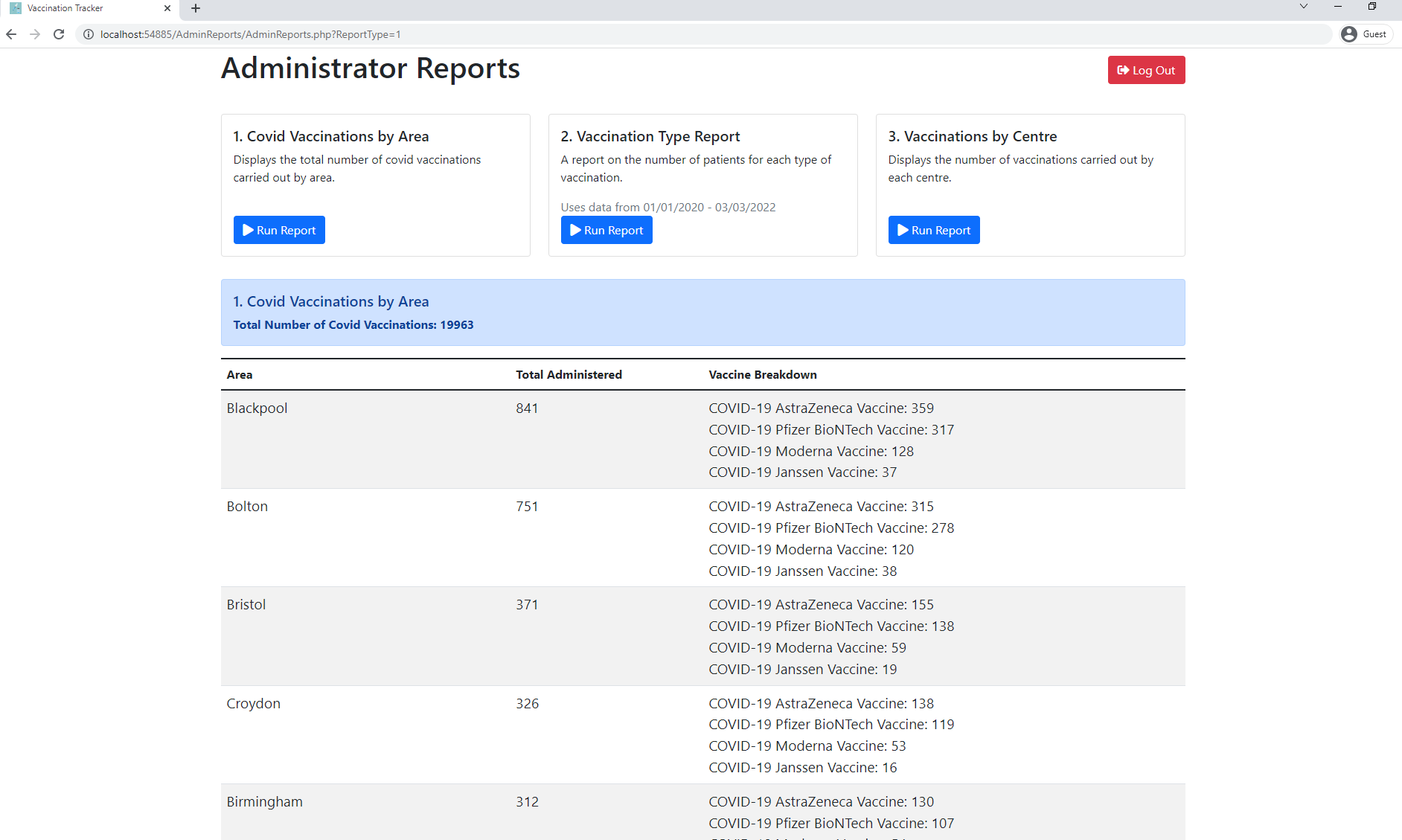


Vaccination History Page

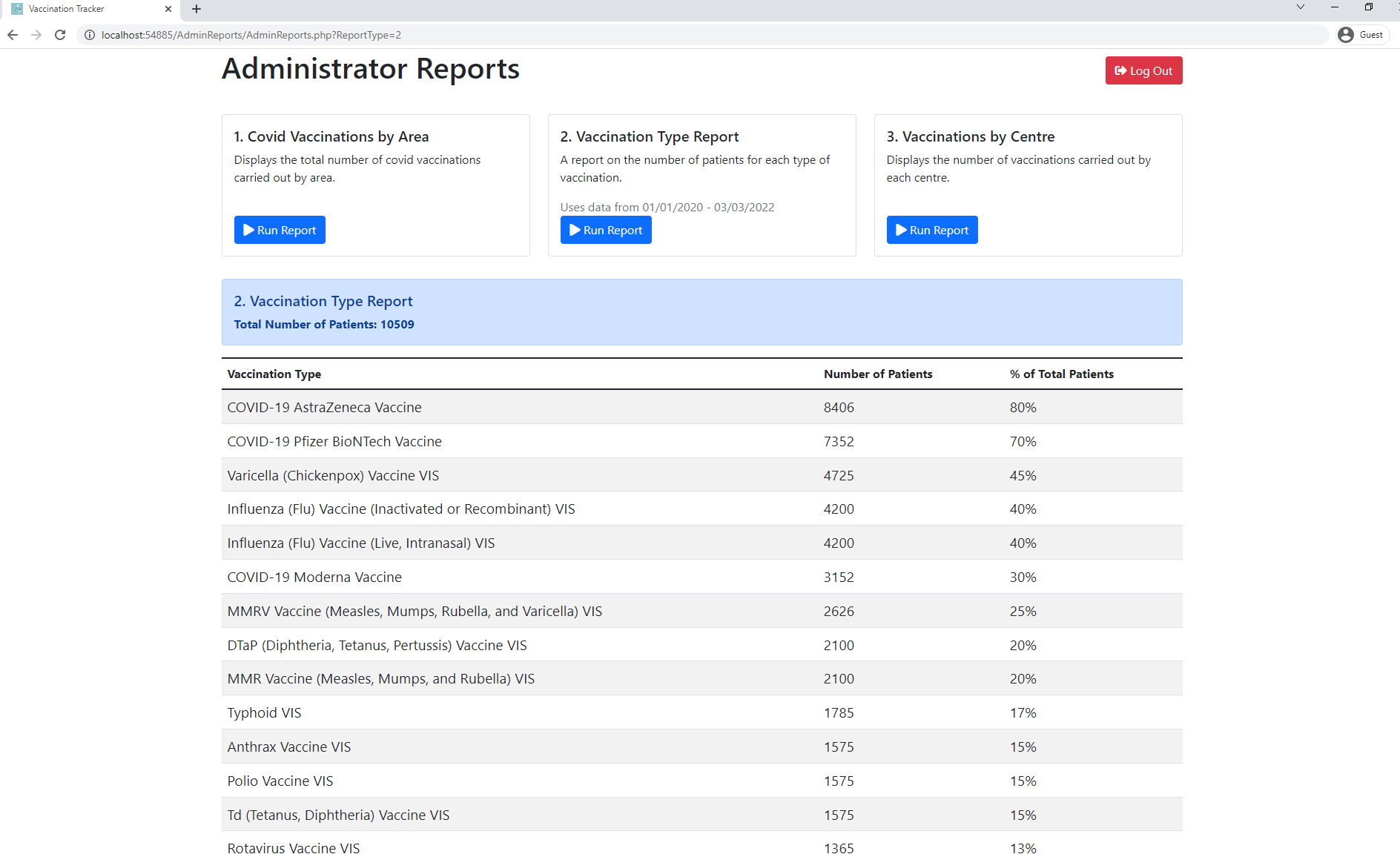


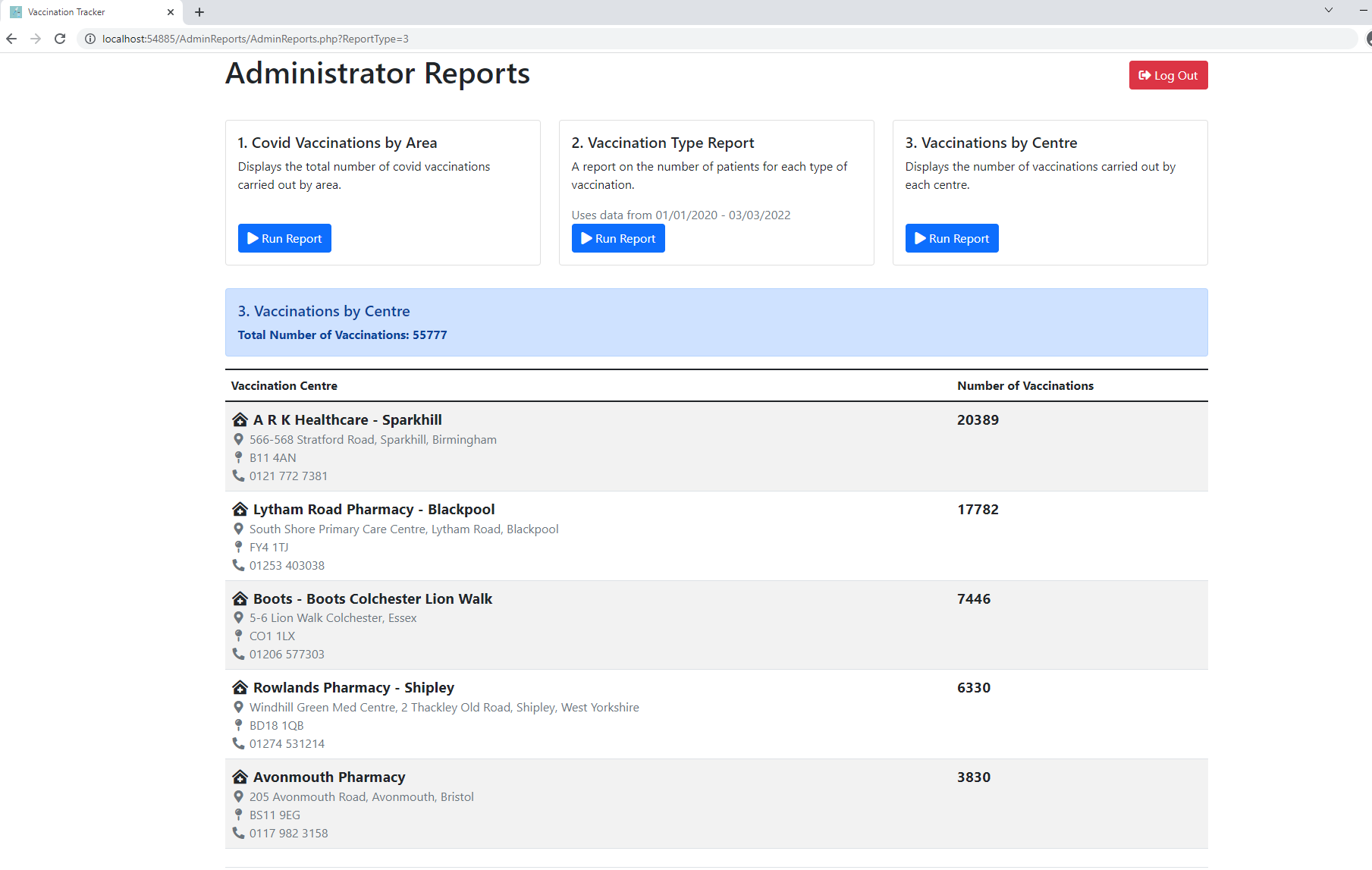
Administrator Reports Page

Administrator Reports Page - Report 1



Administrator Reports Page - Report 2



Administrator Reports Page - Report 3

Research and Development of New Skill

Throughout creating this project, I had to learn new skills that were not included as part of the Advanced Higher course:

Stored Procedures:

Throughout developing this project, I felt writing all the SQL queries within the PHP files became messy as I added more. I looked online for an alternative solution because the SQL queries were hard to look through and change. I came across an article which mentioned stored procedures where essentially you have a function in the database with the SQL code and then you call the function inside the PHP file making it easier to understand exactly what each query should be doing.

Stored Procedures Sources:

* <https://phoenixnap.com/kb/mysql-stored-procedure>
* https://www.sqlshack.com/learn-mysql-the-basics-of-mysql-stored-procedures/

Temporary Tables:

While making the medical persons vaccination history I noticed when I displayed the data in a table, the row numbers wouldn’t be correct. I later figured out the issue was in the query it added the row numbers to the rows before the order by happened. So, after the order by all the numbers would be messed up. The only solution I found to fixing this was using temporary tables. A temporary table is a table that is not stored in the database but only exist while the database session that created it is active.

Temporary Tables Sources:

* <https://www.mysqltutorial.org/mysql-temporary-table/>
* <https://stackoverflow.com/questions/5859391/create-a-temporary-table-in-a-select-statement-without-a-separate-create-table>

Update Join:

Is used to perform cross table updates that we can update one table using another table based on a specified join condition. I used this because I needed to update a temporary table with another temporary tables data for spGetReportCovidVaccinationsByArea stored procedure.

Update Join Sources:

* https://www.mysqltutorial.org/mysql-update-join/

MySQL Substring:

When creating the spGetReportCovidVaccinationsByArea I needed a way to get the area from the address. I looked online and found the way to do this was to use the SUBSTRING\_INDEX function. This function returns the specified number of characters from a particular possition of a given string.

MySQL Substring Sources:

* https://stackoverflow.com/questions/12775352/last-index-of-a-given-substring-in-mysql

Indexing Temporary Table:

When creating the spGetReportCovidVaccinationsByArea I ran into some performance issuess as it took 9 seconds to run this procedure. To fix this I found adding an index to the Temp\_AreaVaxQty temporary table significantly improved the performance and reduced the query run time from 9 to 0 seconds. Indexes are used to quickly locate data without having to search every row n the table every time the table is accessed. Indexes can be created using columns of a table providing efficient access to data.

Indexing Temporary Table Sources:

* <https://stackoverflow.com/questions/14397785/create-a-temporary-table-in-mysql-with-an-index-from-a-select>

PHP Includes:

I wanted to essentially have a template for all pages to use and didn’t want to repeat the code to do so. I found that using php includes made it possible to include code into another file with all the references to Boostrap, Font Awesome and other files without having to repeat it. This made making more pages easier as I didn’t have to keep writing all the head and body tags but only the php and html to make the page.

PHP Includes Sources:

* <https://stackoverflow.com/questions/20589482/how-to-add-content-to-php-include-files/20589533>
* https://stackoverflow.com/questions/49631265/using-the-include-function-in-php

Query String:

A query string is usesd for adding / passing data in the url. I used this as I needed a way to tell the user if the vaccinate patient page form data being saved was successful or unsuccessful.

Query String Sources:

* <https://stackoverflow.com/questions/8469767/get-url-query-string-parameters>
* https://stackoverflow.com/questions/31628948/how-to-show-a-success-message-after-php-form-submission

Use last inserted value for an Auto Increment column:

The LAST\_INSERT\_ID() function returns the lastt Auto Incremented value. I needed this to insert a new patient into the Patients table with the PatientId carrying on from the previously inserted one.

Auto Increment last inserted value Sources:

* https://stackoverflow.com/questions/3837990/last-insert-id-mysql

|  | | | |
| --- | --- | --- | --- |
| What was being tested | Issues highlighted throughout testing | How I fixed these issues | References Used |
| Logout Button | When the logout button is pressed it took the user back to the login page. Users on the login page should not have been able to go to the previous page but while I was testing this I was able to go to the previous page. | I found a solution to this issue by checking if the medicalPersonId session variable was null and the current page was Login.php if this came back as true then the user would be redirected to the login page (Login.php). This logic was placed in the master.php as each page include master.php all pages would check for those conditions | * https://stackoverflow.com/questions/13032930/how-to-get-current-php-page-name |
| Date Field on Vaccinate Patient Form | When logged in as a medical person it goes to the Vaccinate Patient page where the date field should automatically fill in the current date and time of the day. When I was testing to see If this was working it had auto filled the datetime field with the correct date but the time was an hour ahead. | I went on w3schools to check if my code was correct and it was. After some searching the problem seemed to be that the code was printing out the time for a different time zone as I never specified one. I went on the php.net website to see what the code was to specify and also looked at what time zones were valid to put in. This fixed the issue and the correct time and date was being displayed in the datetime field on the Vaccinate Patient form. | * <https://www.w3schools.com/php/php_date.asp> * <https://www.php.net/date_default_timezone_set> * <https://www.php.net/manual/en/timezones.europe.php> |
| Vaccination History | When testing to see if all the data for the vaccination history was being displayed I noticed that the row numbers were incorrect as it was supposed to start from 1 to how many rows that medical person had. But instead all the data was showing but the row number was in the incorrect order. | The problem was that when I was ordering by the datetime it would change the row number as the row number was assigned before the order by was executed. To solve this I created a temporary table called Temp\_VaxHistory and selected the vaccination history into it. Then I would assign the row number to VaxHistory as the data would already be sorted before assigning a row number. Then delete temporary table as it is no longer required. | For defining MySQL variables   * <https://www.mysqltutorial.org/mysql-variables/>   For temporary tables:   * + <https://www.mysqltutorial.org/mysql-temporary-table/>   + <https://stackoverflow.com/questions/5859391/create-a-temporary-table-in-a-select-statement-without-a-separate-create-table> |
| Report 1 Covid Vaccinations By Area | When testing to see if all the correct data would show up in the table on the report page I noticed that it was taking a very long time time to load up. So I went into phpMyAdmin and ran the query that returns the data for the table. It took 9 seconds for it to run. This was too slow as it kept the user waiting for far too long. | After some research I decided to change my query to make use of temporary tables and indexes. Indexes are used to find rows with specific column values very quickly. A temporary table is a table that is not stored in the database but only exist while the database session that created it is active. I created a temporary table called Temp\_AreaVaxDetails that gets the area, vaccine type, total for each vaccine type for each area and adds a column for the total number of covid vaccinations. I then created Temp\_AreaVaxQty temporary table to hold the total number of covid vaccinations for each area adding an index to this table to make the query run faster from 9 seconds to 0 seconds. After that I updated values in Temp\_AreaVaxDetails table using the values from Temp\_AreaVaxQty table making it much more efficient  as it doesn't need to query to get the values it needs because of the Temp\_AreaVaxQty table having the data already. Finally, it selects all from Temp\_AreaVaxDetails ordering by TotalVax DESC, Area, NumVaxByArea DESC, Vaccine. After this it deletes the temporary tables as they are no longer required. | * For temporary tables:   + <https://www.mysqltutorial.org/mysql-temporary-table/>   + <https://stackoverflow.com/questions/5859391/create-a-temporary-table-in-a-select-statement-without-a-separate-create-table> * For indexing a temporary table:   + <https://stackoverflow.com/questions/14397785/create-a-temporary-table-in-mysql-with-an-index-from-a-select> |
| Problem with DatabaseHelpers.php calling SavePatientVaccination function | The original CallDatabaes function I made only delt with queries returning data. So when I called the SavePatientVaccination I it did not return any data as the function only inserts data, so the php returned an error and was not inserting the data into the database. | To fix this I added a Boolean parameter $isDataReturned that checks if any data is being returned. Inside the CallDatabase function it check if any data has been returned then store the data in a variable to then return. If there is no data being returned then just run the query. So in the SavePatientVaccination function I put false for the $isDataReturned parameter then inserting the patient vaccination into the database started to work. For all the other functions that returned data I had to put true for the $isDataReturned parameter. | * Did not use any sources to solve this issue |
| Query String | Fot the vaccinate patient page I needed a way to tell the user that the data being saved was successful or unsuccessful as the form looked at another PHP file to save the data and redirects back to the vaccinate patient page. | The way I got around this was by setting a session variable called savedPatientVax and in the CallDatabase function. If the result is equal to false then the session variable is set to failed otherwise it’s set to success. Then in VaccinatePatient\_Save.php I set in the header statement to redirect back to the vaccinate patient page after the file name put ?SavedPatient = plus the session variable. Then back in the saved patient page I checked if the Saved Patient query string was success then display a success message and if it was failed then display an error message. | https://stackoverflow.com/questions/8469767/get-url-query-string-parameters  https://stackoverflow.com/questions/31628948/how-to-show-a-success-message-after-php-form-submission |
| Change Patients table Age column to DateOfBirth | As I was making the Vaccinate Patient form I realised that the patient age field was incorrect and instead should be the date of birth. | To fix this mistake in the Patients table I created a new column called DateOfBirth and set it as a date field. Then created a query that took the current date minus the patient age giving the patient date of birth. I then updated the rows in the Patients table. After this, removed the age column from the table. | Did not use any sources to solve this issue |
| UPDATE JOIN | While making spGetReportCovidVaccinationsByArea stored procedure I needed a way to update data from one temporary table to another. This is because updating the data from another temporary table makes it fast to run as there are no extra queries taken place to get the data. | To do this I found online that you can have an update join between two tables based on a specific join condition. | https://www.mysqltutorial.org/mysql-update-join/ |
| MySQL Substring function for gettting area | I needed to get the area from the address in spGetCovidVaccinationsBByArea as I was going to group all covid vaccinations carried out by area. | I used the MySQL SUBSTRING\_INDEX() function AS Area and passed through the Patients Address to only get the area from the address so that I can do a GROUP BY on the Area | https://stackoverflow.com/questions/12775352/last-index-of-a-given-substring-in-mysql |
| Adding new patient to Patients table | When I was checking spSavePatientVaccination stored procedure was working correctly and inserting the data correctly into the database. I realised that the PatientId auto\_increment field was not inserting the previously inserted one. | I found by setting the @PatientId variable to LAST\_INSERTED\_ID() function in the stored procedure that it would correctly use the next auto\_increment value | https://stackoverflow.com/questions/3837990/last-insert-id-mysql |

# Testing the Final Solution

## Final Test Plan

Persona

Medical Person:

This Persona is made for a medical person with their medical profession being a Physician. Their name is Adriana Wilkers who works at a vaccination centre and is registered as a vaccinator. The vaccinator will be able to login by selecting their name and profession from the drop down and pressing the login button. They will then be able to input their patient vaccination details and save a patient vaccination to the system. They will also be able to view their patient vaccination history with all their vaccinations carried out. This person has no accessibility issues.

They have a patient who had a vaccination and is needed to be stored in the system. The patient is in for COVID-19 Pfizer BioNTech Vaccine at the A R K Healthcare – Sparkhill vaccination centre. Use the current date & time auto filled in field. The patient name is John Hector, patient identifer is 12345678, date of birth is 03/07/1945, address is 23 Test Grove, Blackpool, postcode is B48 3ST, phone number is 07756945537.

Should validate whether a user has been selected to log in and store the patient vaccination details into a database. The form should validate all the data and the database should also. All previous and current vaccinations added to the system should be available to view in the vaccination history.

This Persona will be doing integrative testing and will test all parts of the site for a vaccinator works correctly with all the components.

Administrator:

This Persona is for someone who is an administrator for a vaccination centre and is registered as an administrator. The administrator will be able to log in by selecting the Demo Administrator username from the drop down and pressing the login button. They will then be directed to the page where the can view three reports. Report 1 covid vaccinations by area, report 2 vaccination type report and report 3 vaccinations by centre. This person has no accessibility issues.

The site should run report and get data from database and display in a table for each of the reports.

This Persona will be doing integrative testing and will test all parts of the site for the administrator and check if all the components work correctly.

## Test Table

Medical Person:

| Test Requirements | Description | Test Case | Type of Testing | Expected Output |
| --- | --- | --- | --- | --- |
| Login | Select vaccinator from Username drop down field. | Click on Username drop down and select from the list of Vaccinators and login. | Normal:  Select a Vaccinator  Extreme:  N/A  Exceptional:  Don’t select any username | The user should be taken to the Vaccinate Patient page. |
| Vaccinate Patient form | Enter patient vaccination details and save patient vaccination | Fill out form to save patient vaccination.  A vaccination centre must be selected from drop down.  Enter a date – should use current pre-filled date time.  A vaccination type must be selected from drop down.  Enter Patient Identifier, characters must be greater than 0.  Enter Patients First Name, characters must be greater than 0.  Enter Patients Last Name, characters must be greater than 0.  Enter Patients Date of Birth.  Enter Patients Address, characters must be greater than 0.  Enter Patients Postcode, characters must be greater than 0.  Enter Patients Phone Number, characters must be greater than 0. | Normal:  Enter Patient Identifier, First Name, Last Name Address, Postcode, Phone Number with the characters length greater than 0  Extreme:  N/A  Exceptional:  Leave fields blank. Don’t select a drop down item. | Normal:  Successfully insert data into database and display success message to user.  Exceptional:  Error message to be displayed to user and no data inserted into database. |
| Vaccination History | Check new patient vaccination has been added to table with other previous patient vaccinations. | Check if number of vaccinations is the same number as the number of rows in the table. | Normal:  Click the Vaccination History button to direct to Vaccination History Page.  Extreme:  N/A  Exceptional:  N/A | Total number of vaccinations should update because of new patient vaccination added to the system. Also displaying the details of the newest patient vaccination with all the previous ones. |
| Update existing patient details | Add new patient vaccination but for the patient identifier use the one you used previously | Fill in the same details into input fields used to test vaccinate patient form previously.  For patient identifier put: 1234578  For patient last name put Bruno | Normal:  Fill in details used to test vaccinate patient form.  For patient identifier: 12345678  For patient last name put Bruno  Extreme:  N/A  Exceptional:  N/A | In Vaccination History there should be a new vaccination in the table with the previous one and the patient last name should be Bruno instead of Hector. The database should also update these details. |
| Log out |  | At the top right of the page click the log out button to log out. | Normal:  Click the button to log out.  Extreme:  N/A  Exceptional:  N/A | The log out will be successful and will redirect you back to the login page. You will not be able to go back to the previous page after logging out as you are not logged in to proceed further than the login page. |

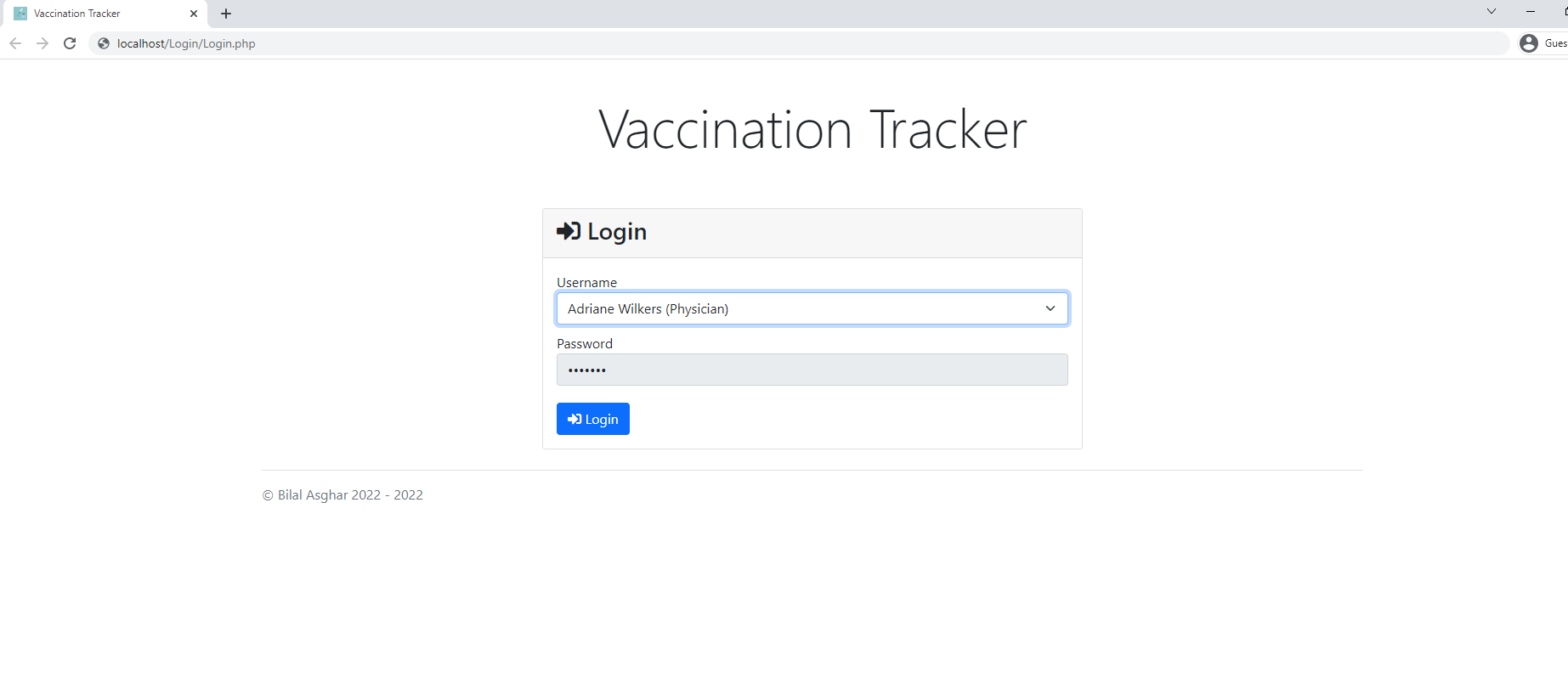
Administrator:

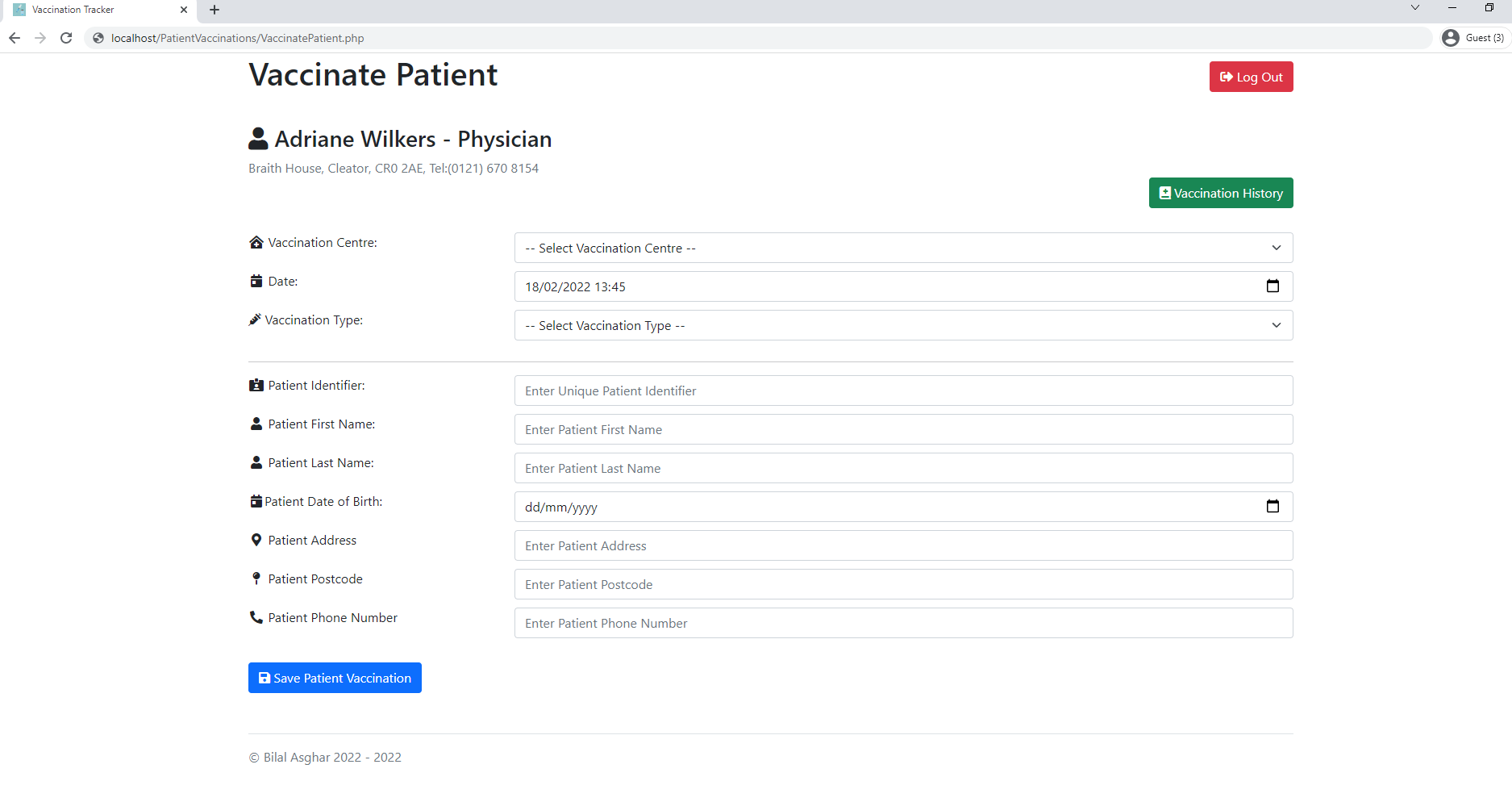
| Test Requirements | Description | Test Case | Type of Testing | Expected Output |
| --- | --- | --- | --- | --- |
| Login | Select Demo Administrator from Username drop down field. | Click on Username drop down field and select from the Demo Administrator and login. | Normal:  Select Demo Administrator  Extreme:  N/A  Exceptional:  Don’t select any username | The user should be taken to the Administrator Reports page. |
| Run Report 1 | Report  1. Covid Vaccinations By Area | Under 1. Covid Vaccinations by Area Click Run Report | Normal:  Click run report button for report 1 in Administrators Reports Page.  Extreme:  N/A  Exceptional:  N/A | Report 1 will display the total number of covid vaccinations. A table with the Area, Total Administered and Vaccine Breakdown data displayed. |
| Run Report 2 | Report  2. Vaccination Type Report | Under 2. Vaccination Type Report Click Run Report | Normal:  Click run report button for report 2 in Administrators Reports Page.  Extreme:  N/A  Exceptional:  N/A | Report 2 will display the total number of patients vaccinated. A table with the Vaccination Type, Number of Patients and percentage of Total Patients data displayed. |
| Run Report 3 | Report  3. Vaccinations by Centre Report | Under 3. Vaccinations by Centre Click Run Report | Normal:  Click run report button for report 3 in Administrators Reports Page.  Extreme:  N/A  Exceptional:  N/A | Report 3 will display the total number of vaccinations carried out. A table with the Vaccination Centre and Number of Vaccinations for each centre data displayed. |
| Log out |  | At the top right of the page click the log out button to log out. | Normal:  Click the button to log out.  Extreme:  N/A  Exceptional:  N/A | The log out will be successful and will redirect you back to the login page. You will not be able to go back to the previous page after logging out as you are not logged in to proceed further than the login page. |

## Requirements Testing

Medical Person:

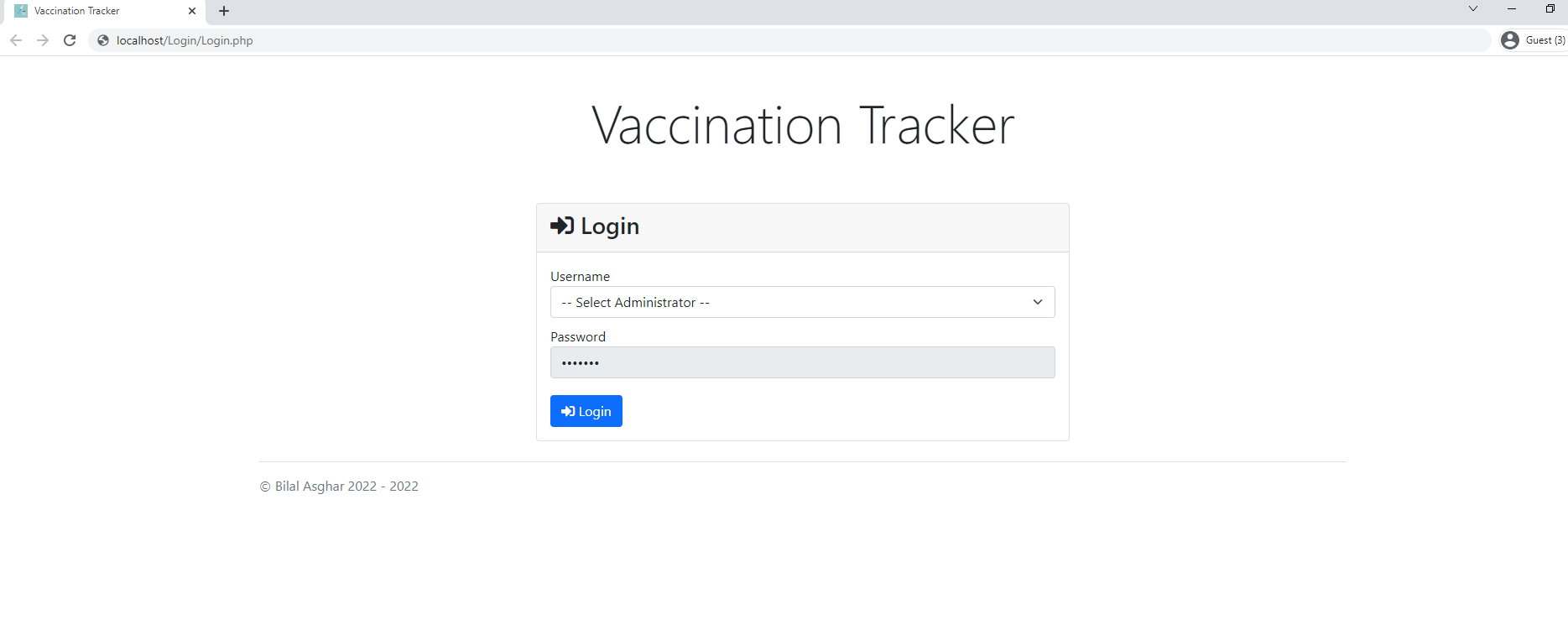
Login (Normal)

Input:

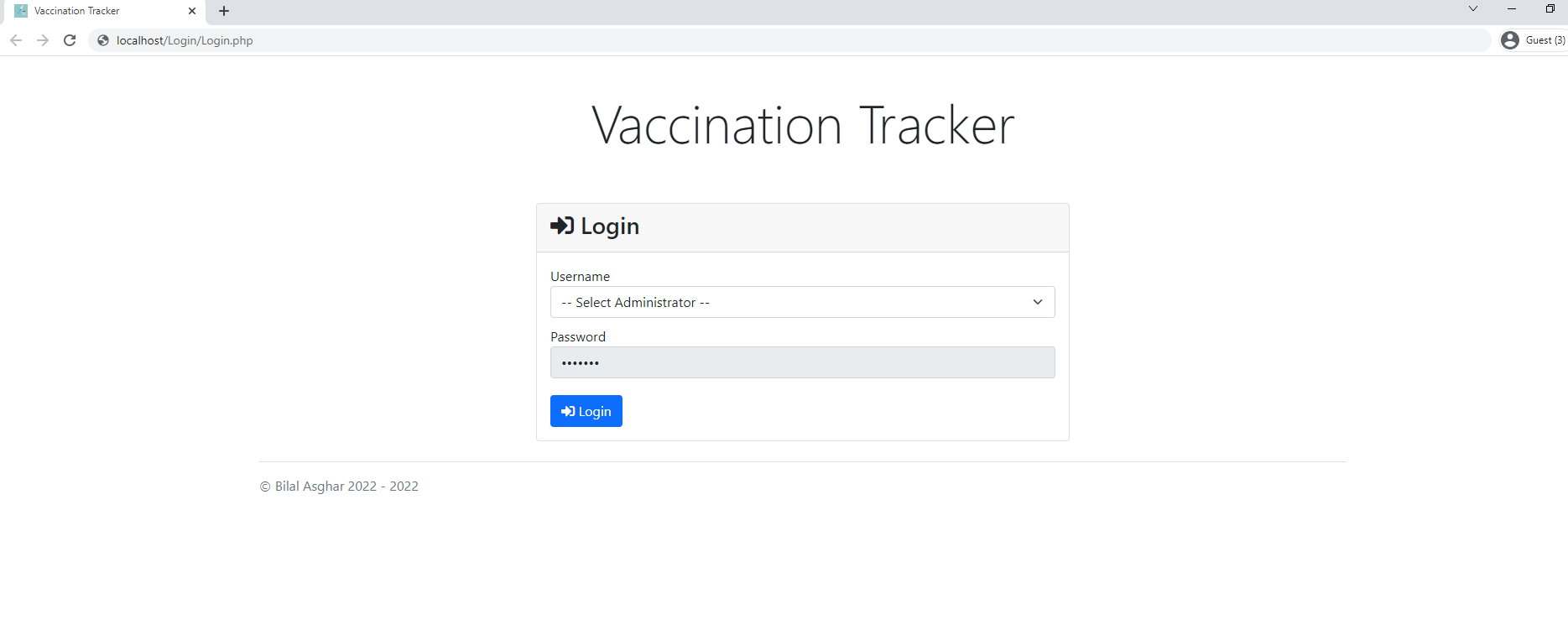
Output:

Login (Exceptional)

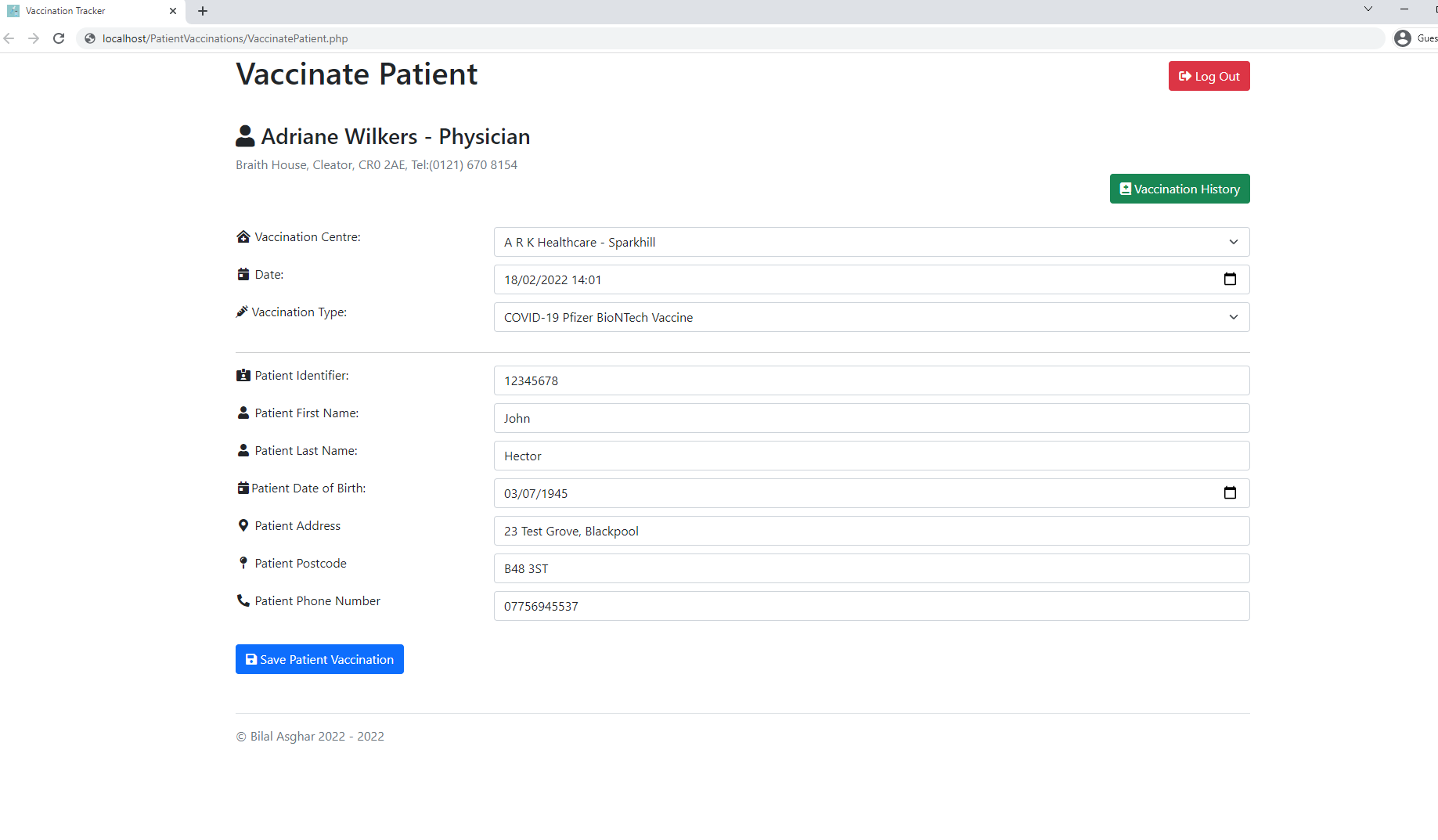
Input:

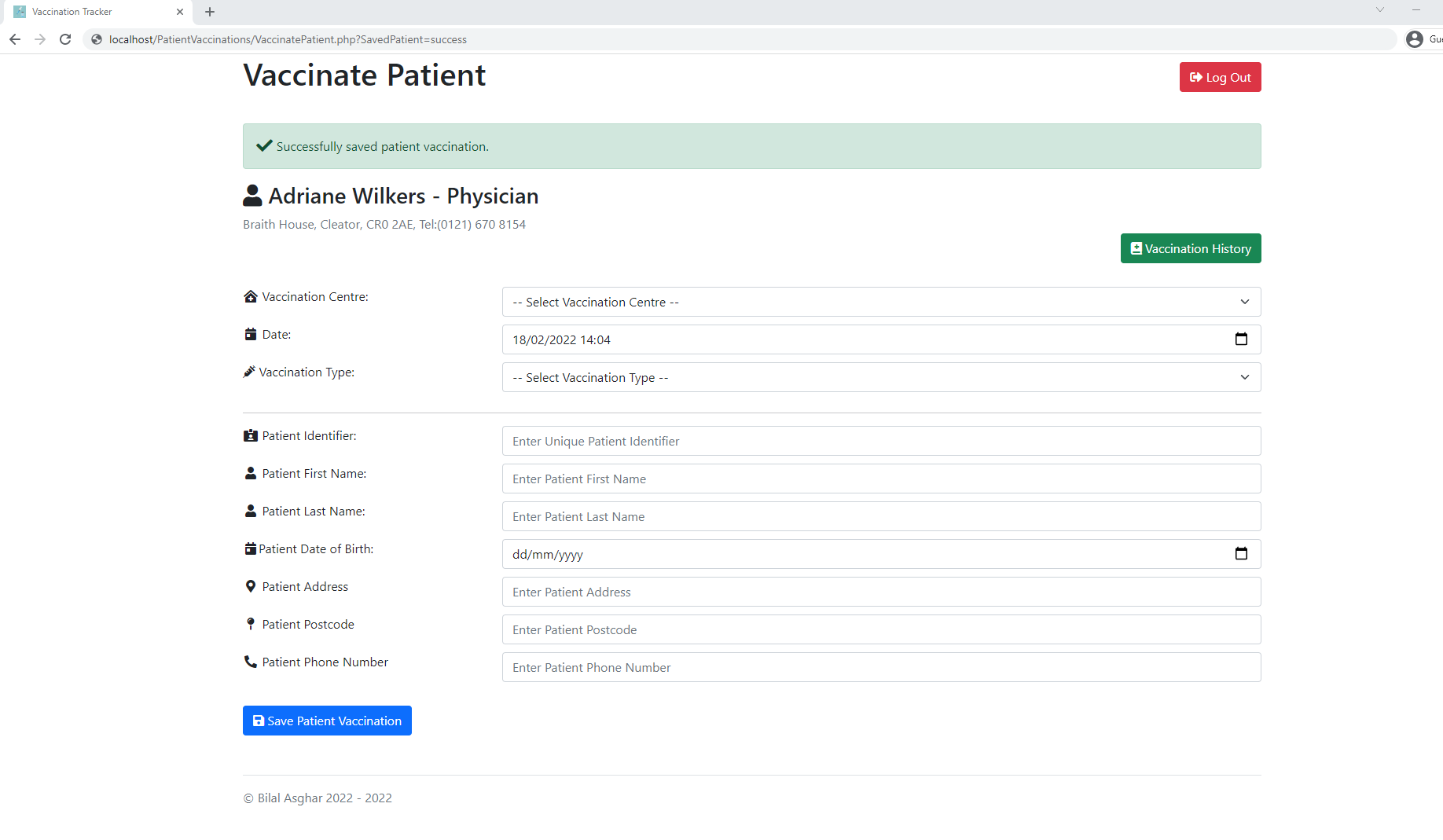
No selected log in

Output:

Redirected back to login page when no valid login is selected

Vaccinate Patient Form (Normal)

Input:

Output:

New Patient added to Patients table

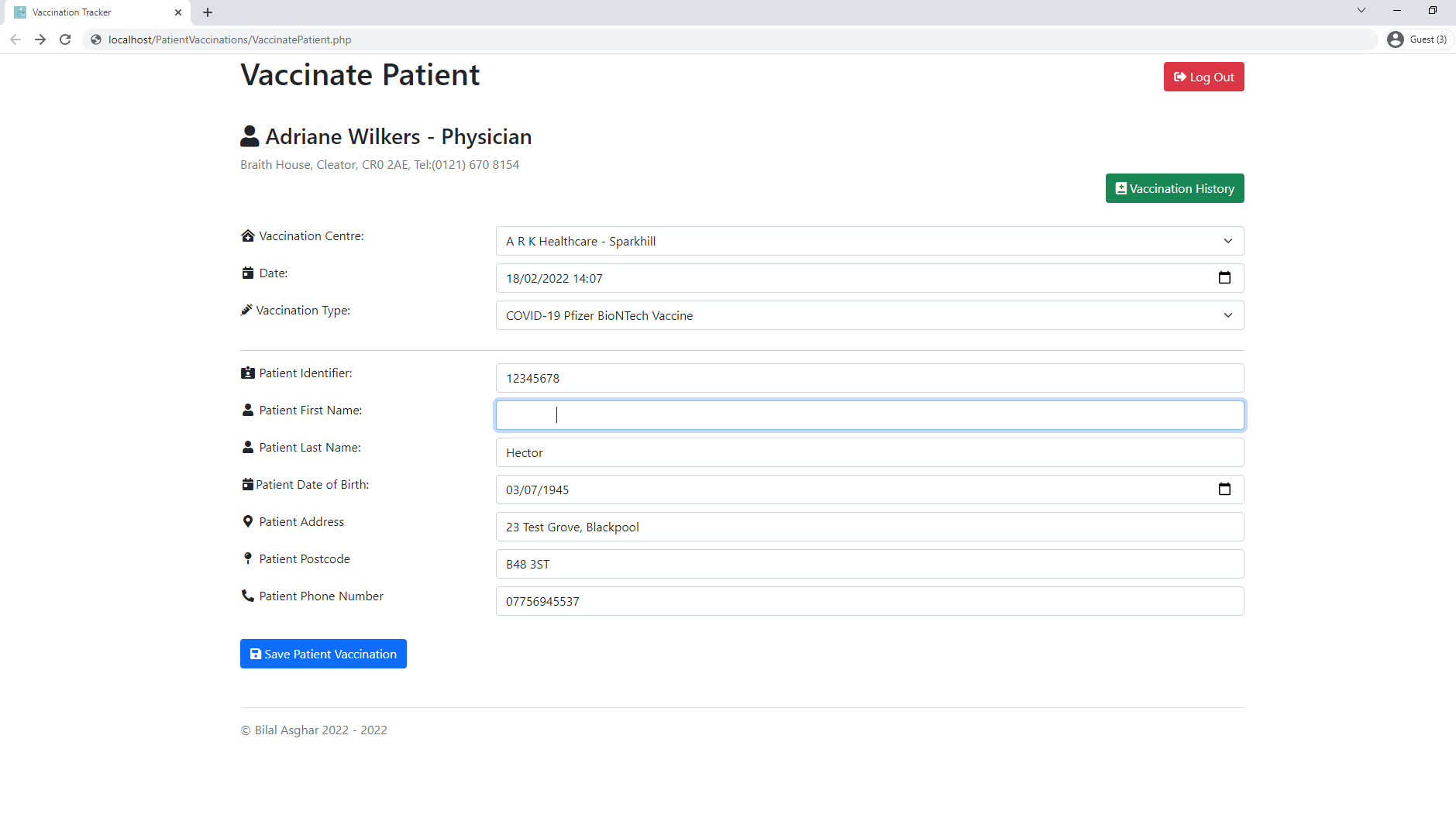


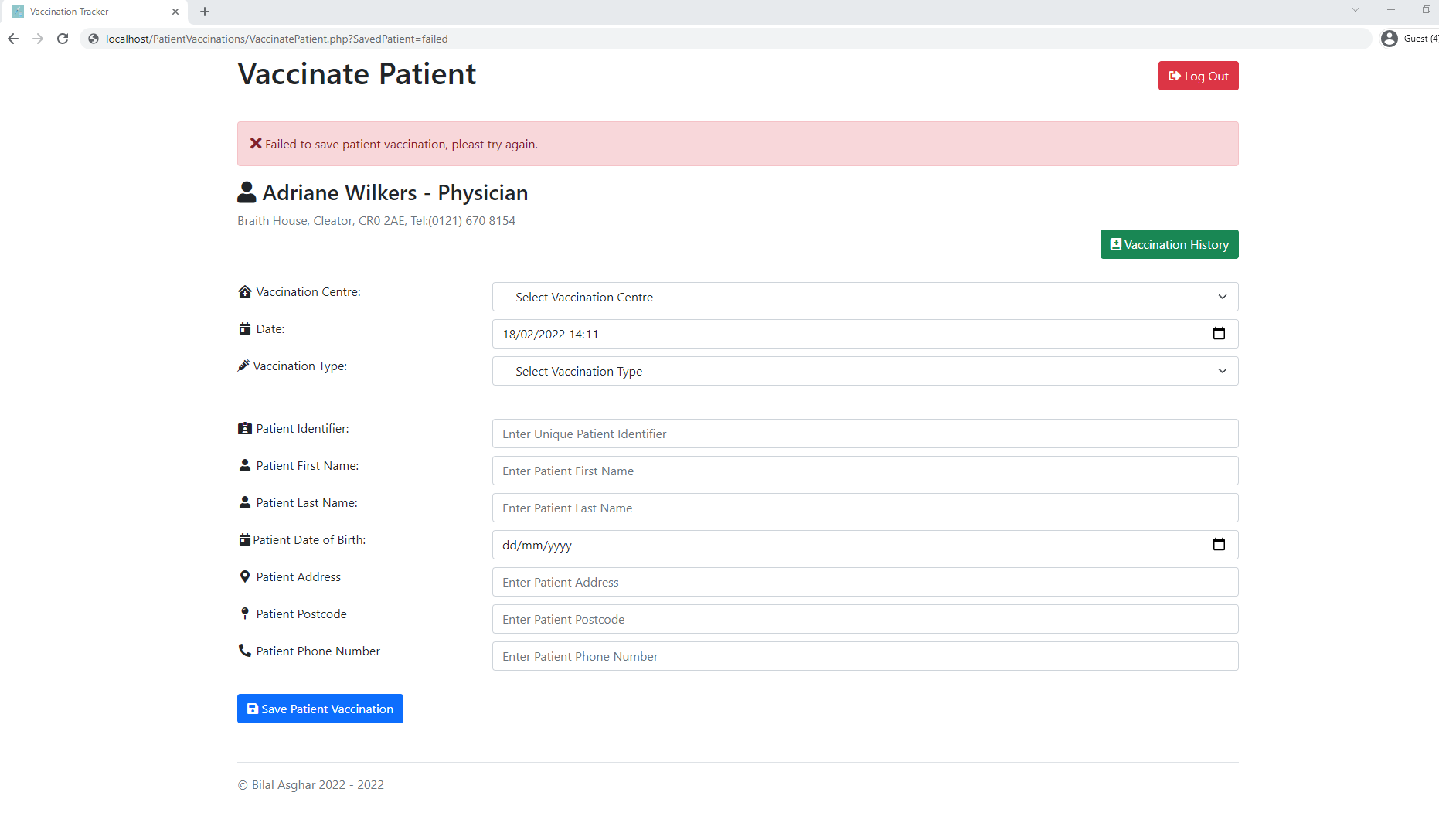
New Patient Vaccination added to PatientVaccinations table



Vaccinate Patient Form (Exceptional)

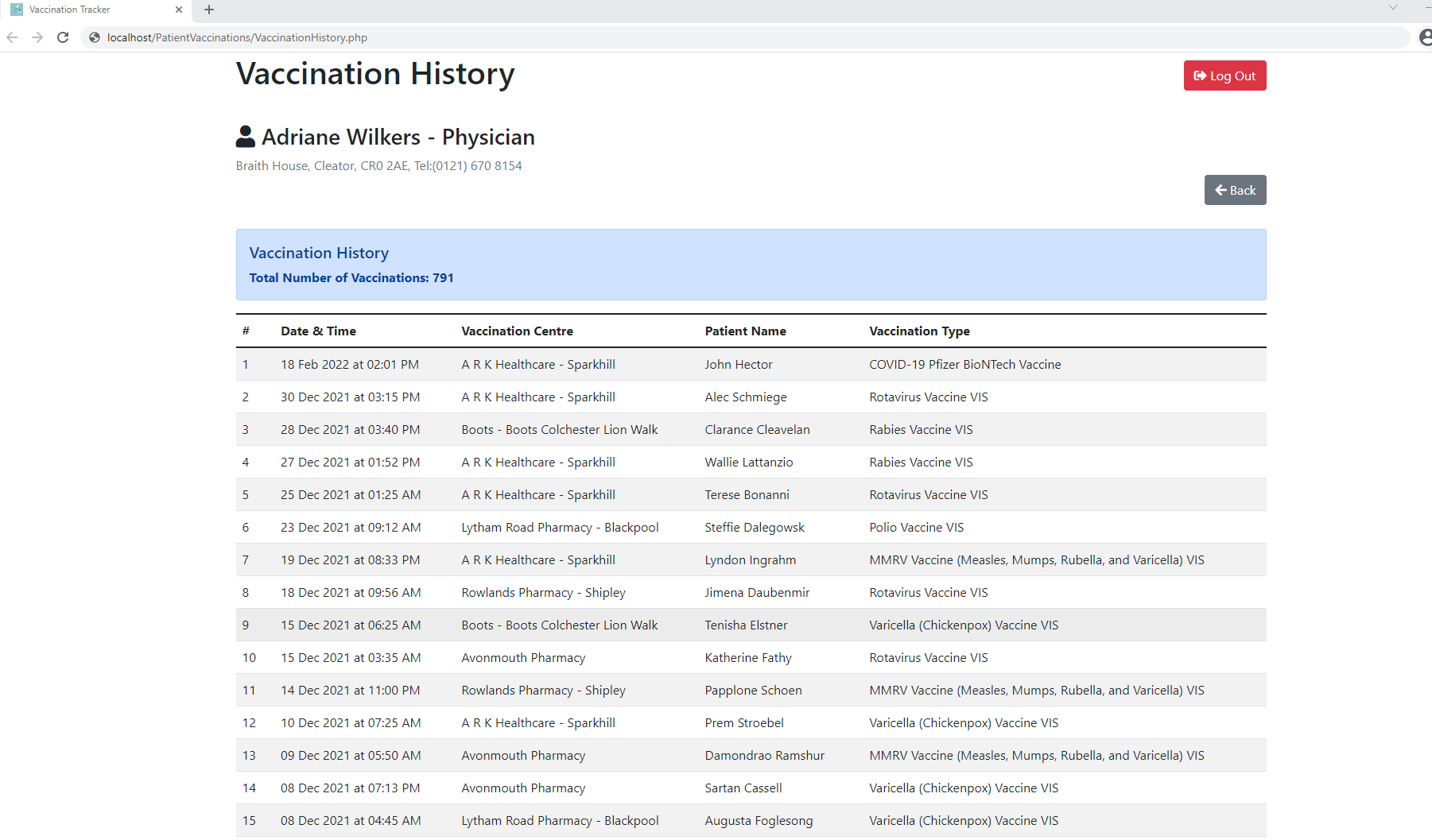
Input:

Blank spaces in Patient First Name input field

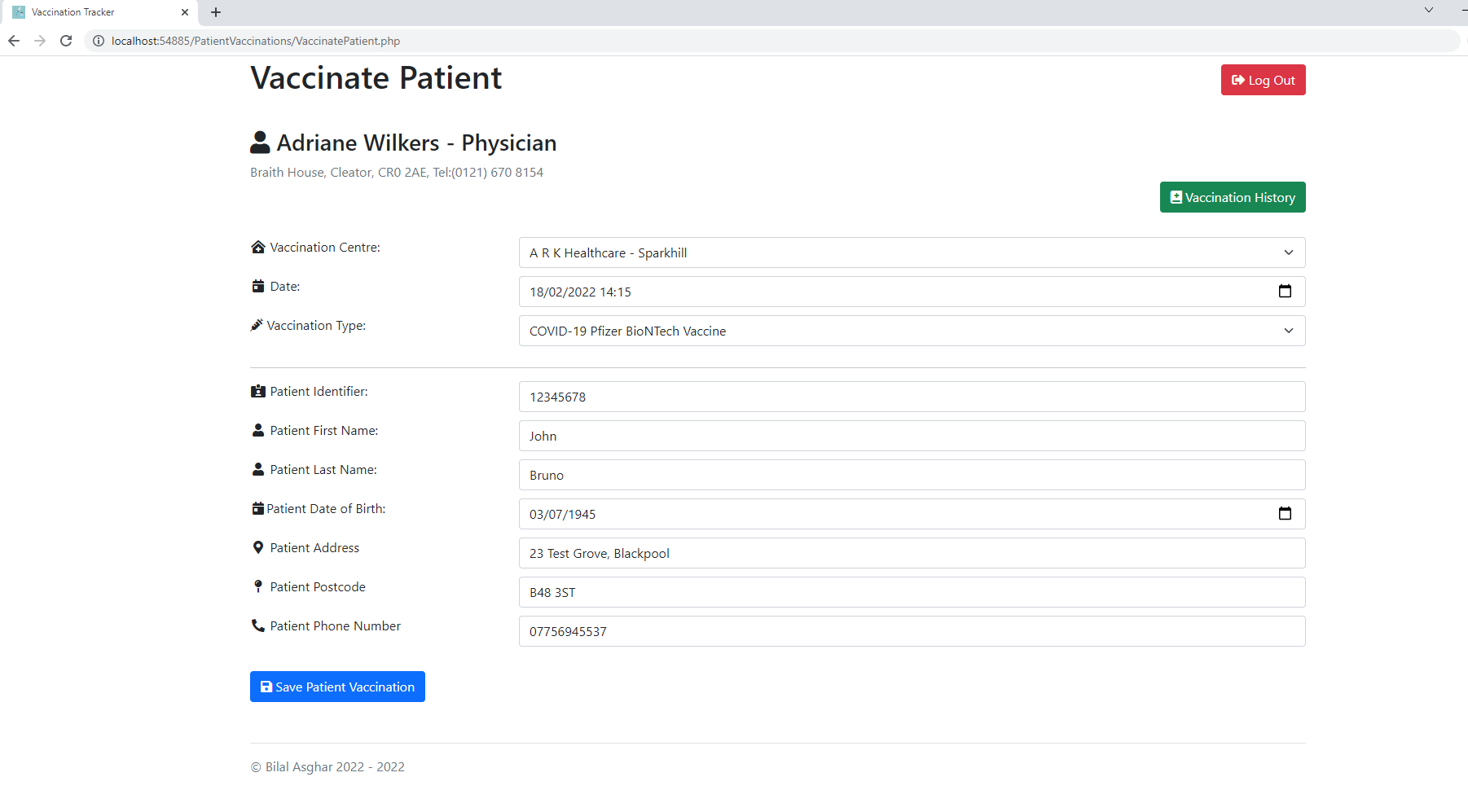
Output:

No data inserted into database and error message displayed to user

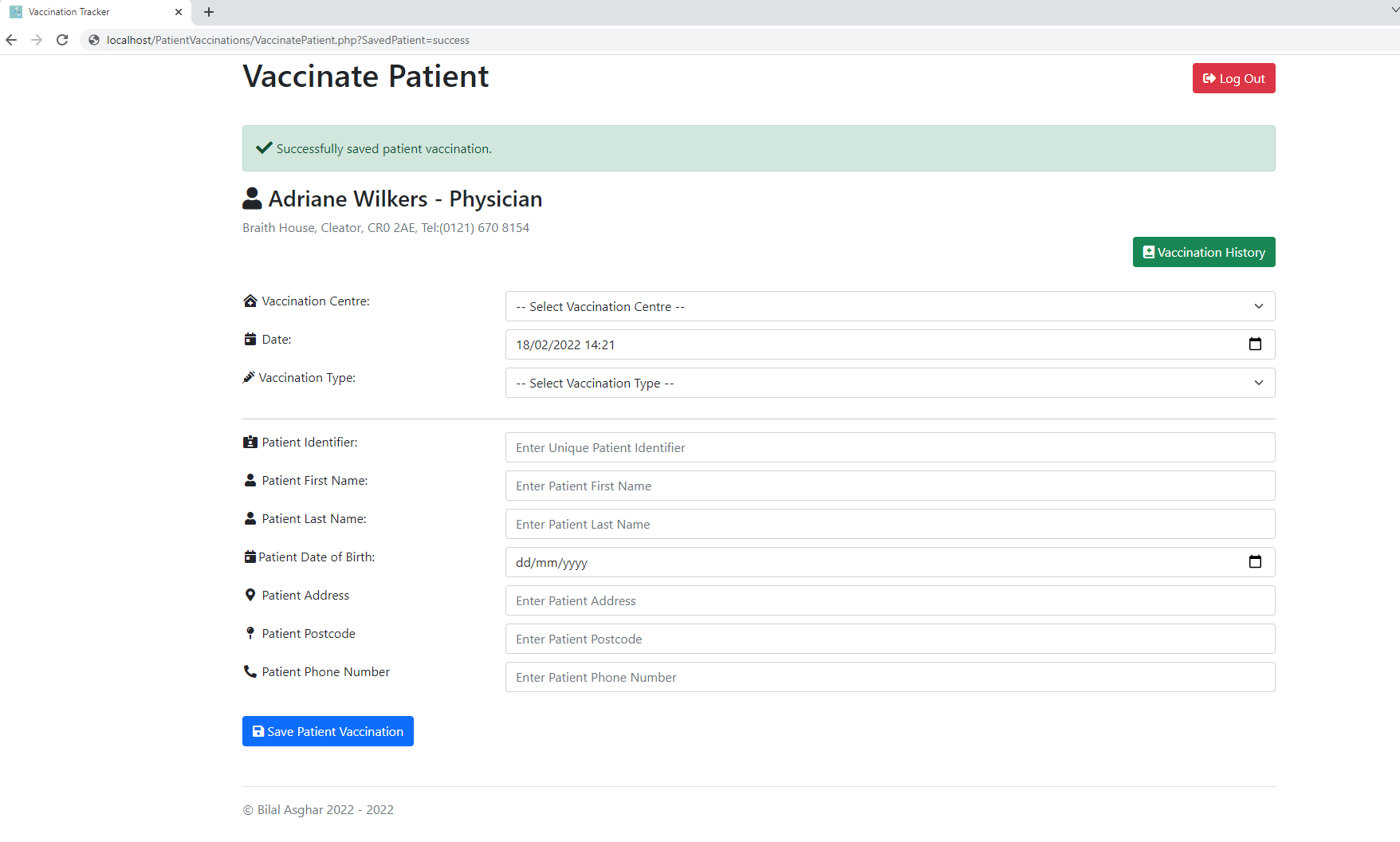
Vaccination History

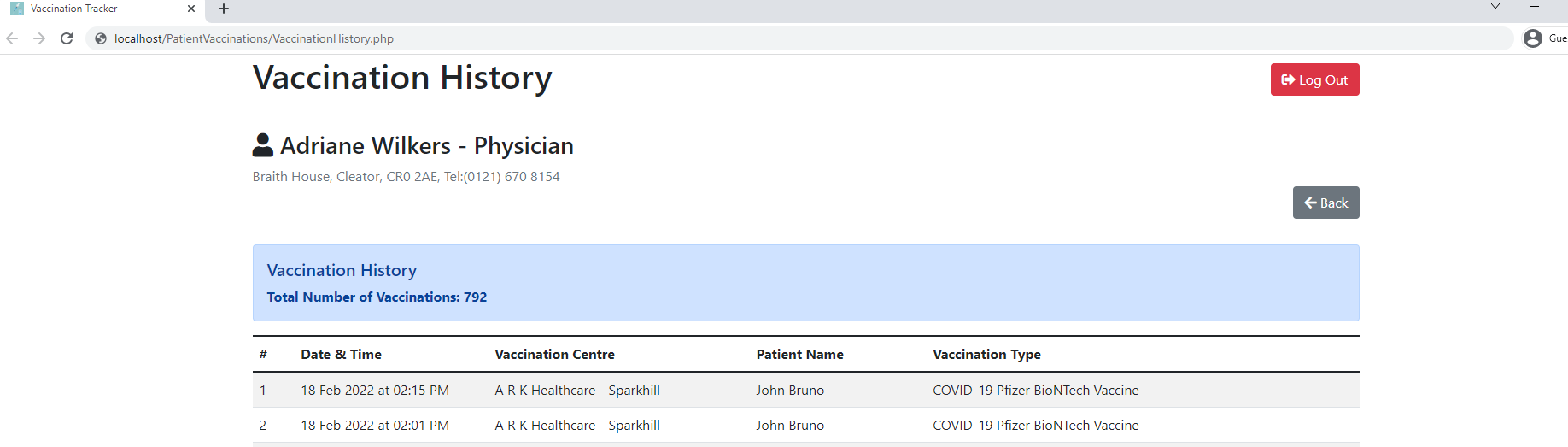
Output:

Update Existing Patient Details (Normal)

Input:

Output:





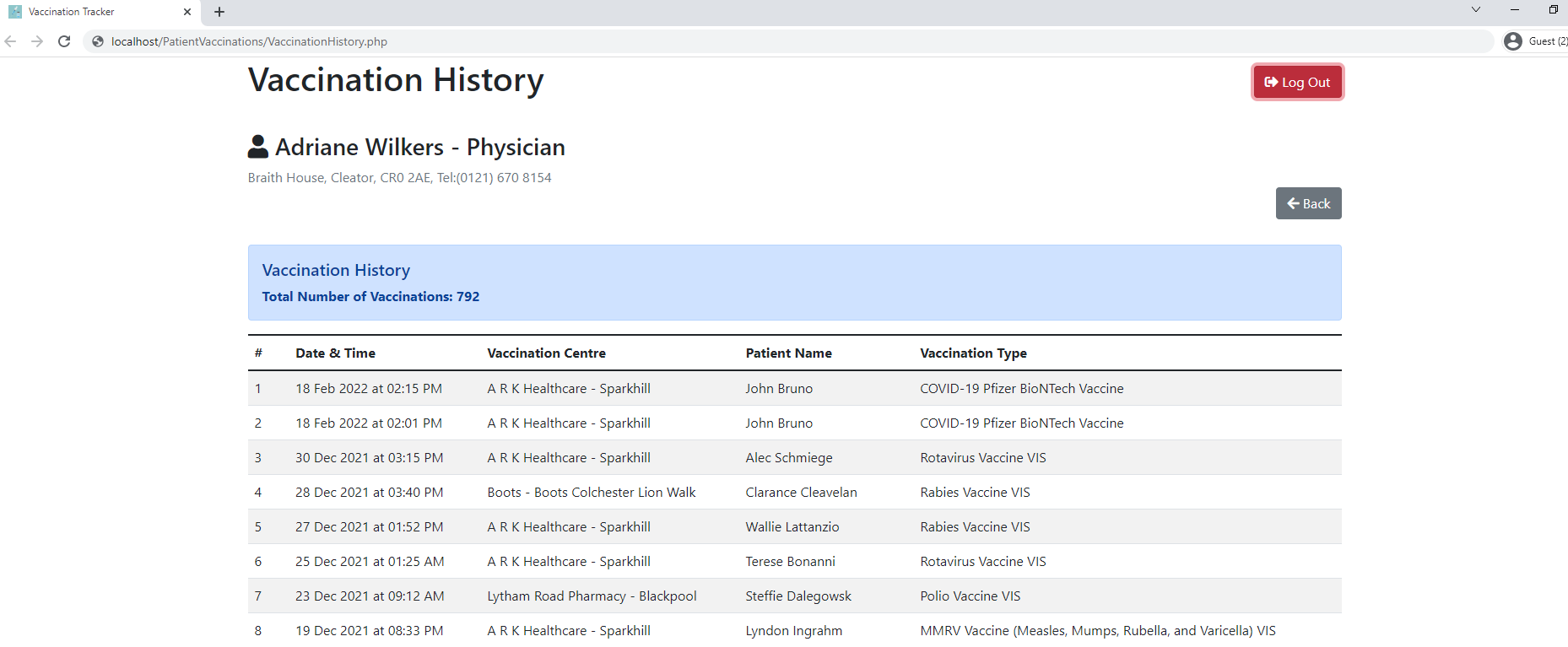
Updated patient in Patients table:

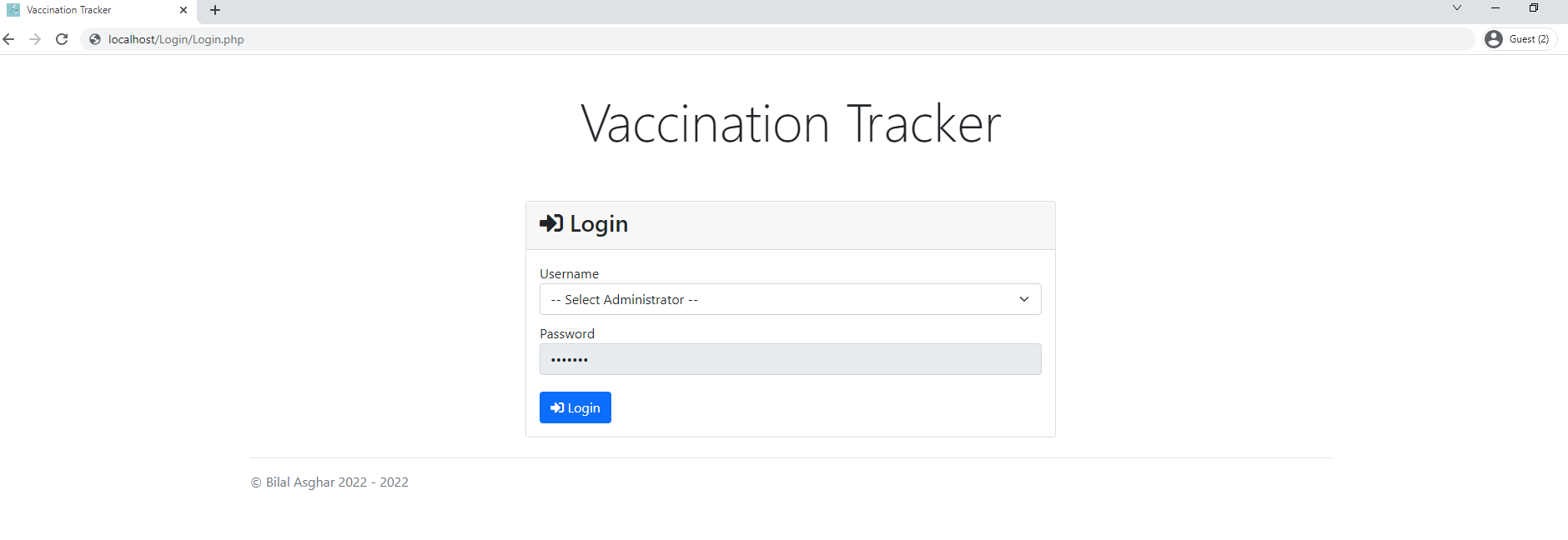


New patient vaccination added to PatientVaccinations table:



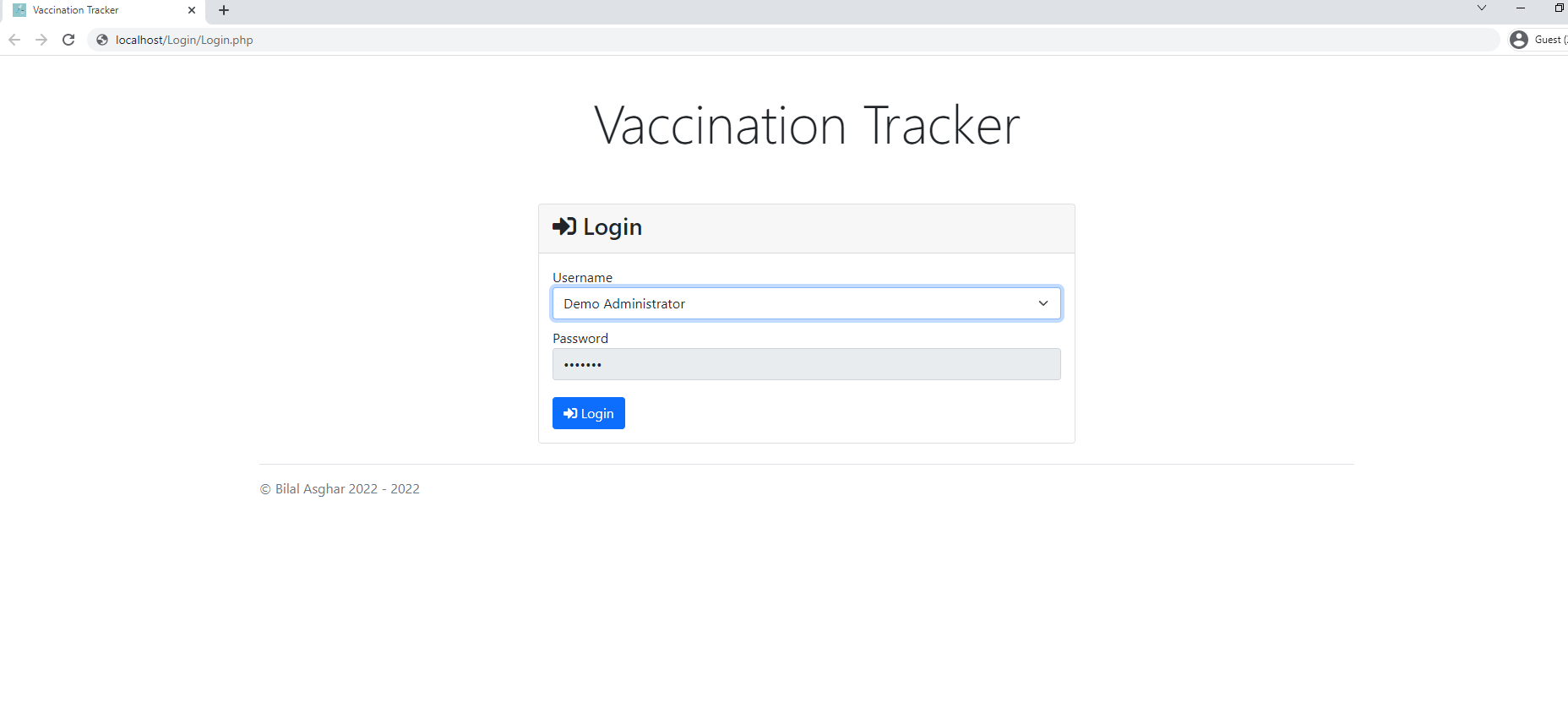
Log Out (Normal)

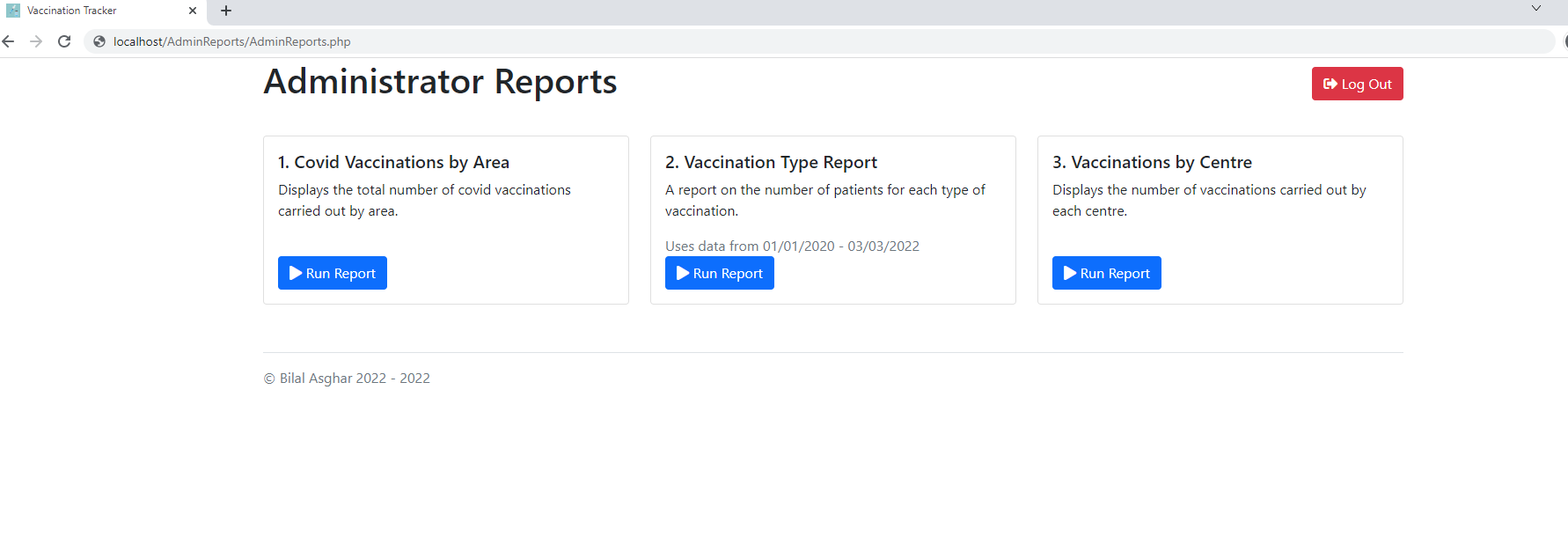
Input:

Output:

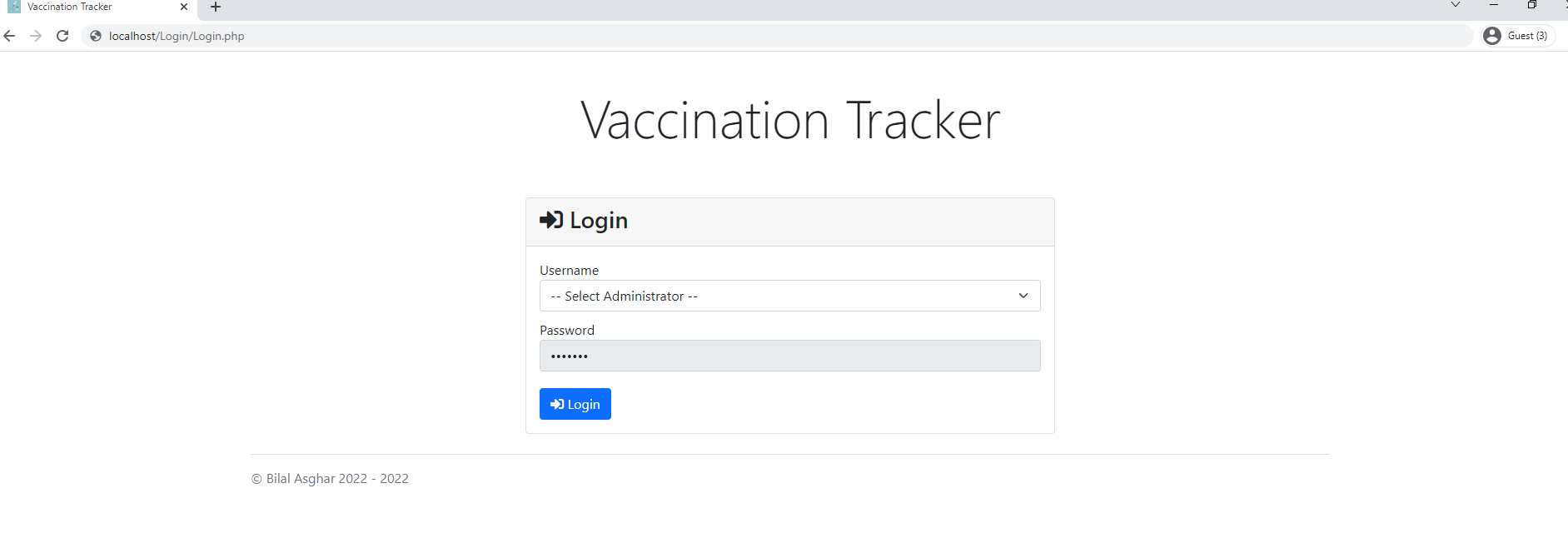
Administrator:

Login (Normal)

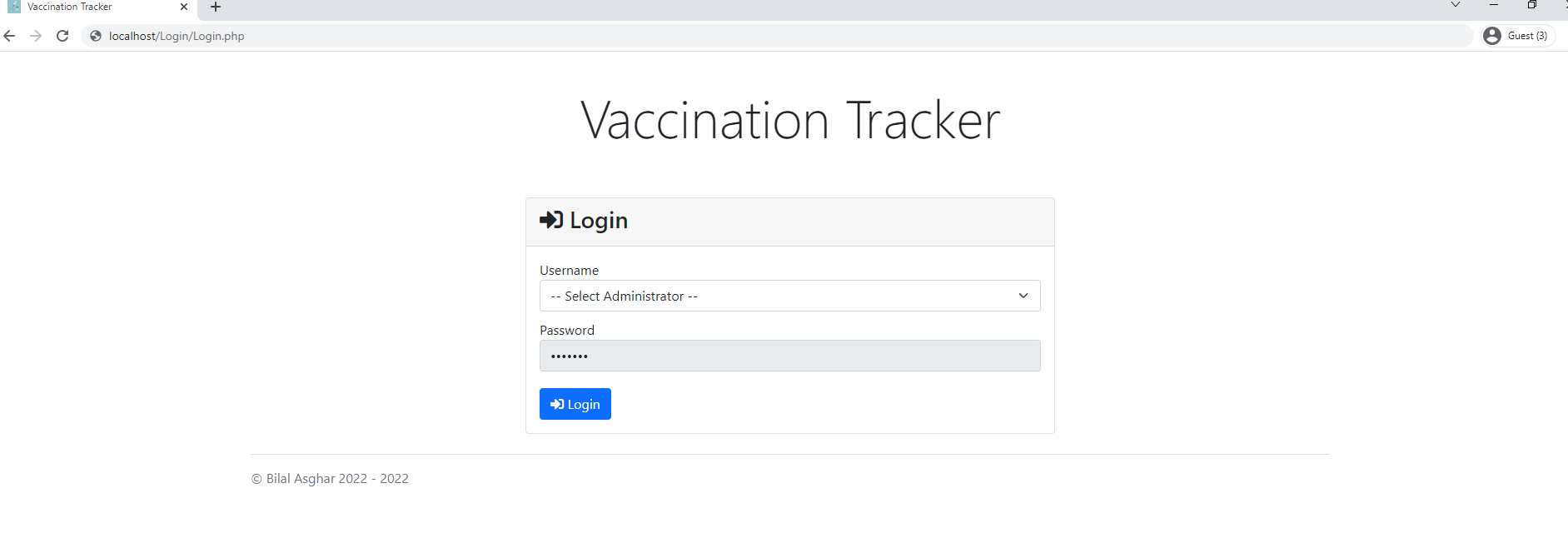
Input:

Output:

Login (Exceptional)

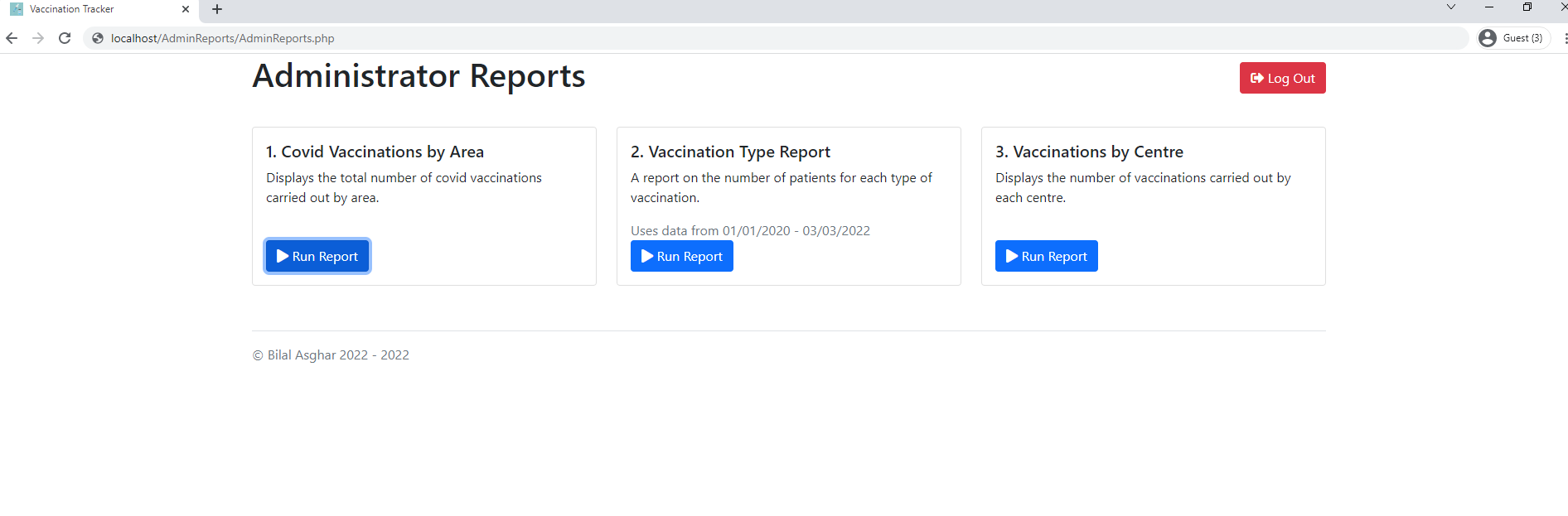
Input:

Output:

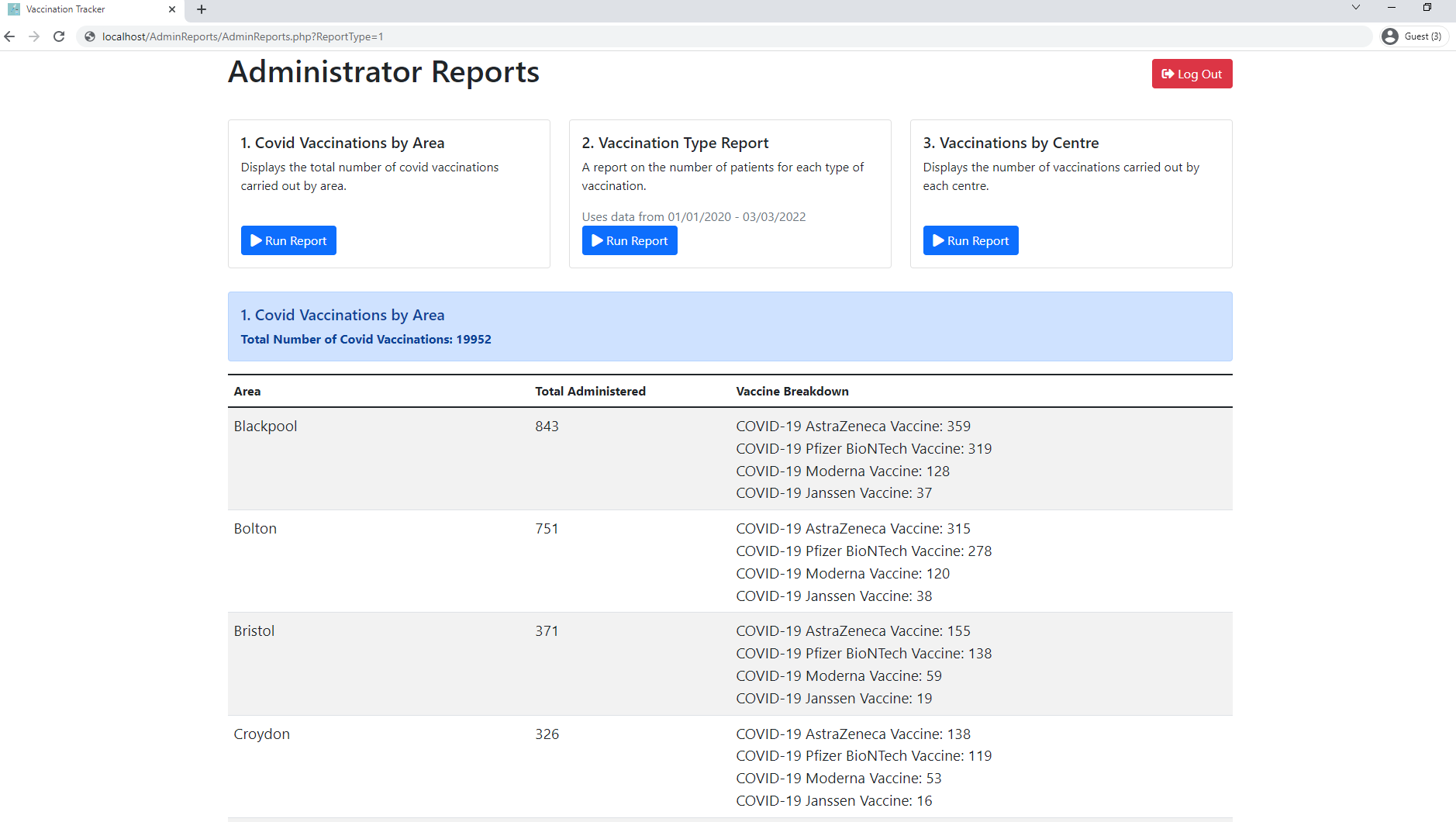
Redirected back to login page when no valid login is selected

Run Report 1 (Normal)

Input:

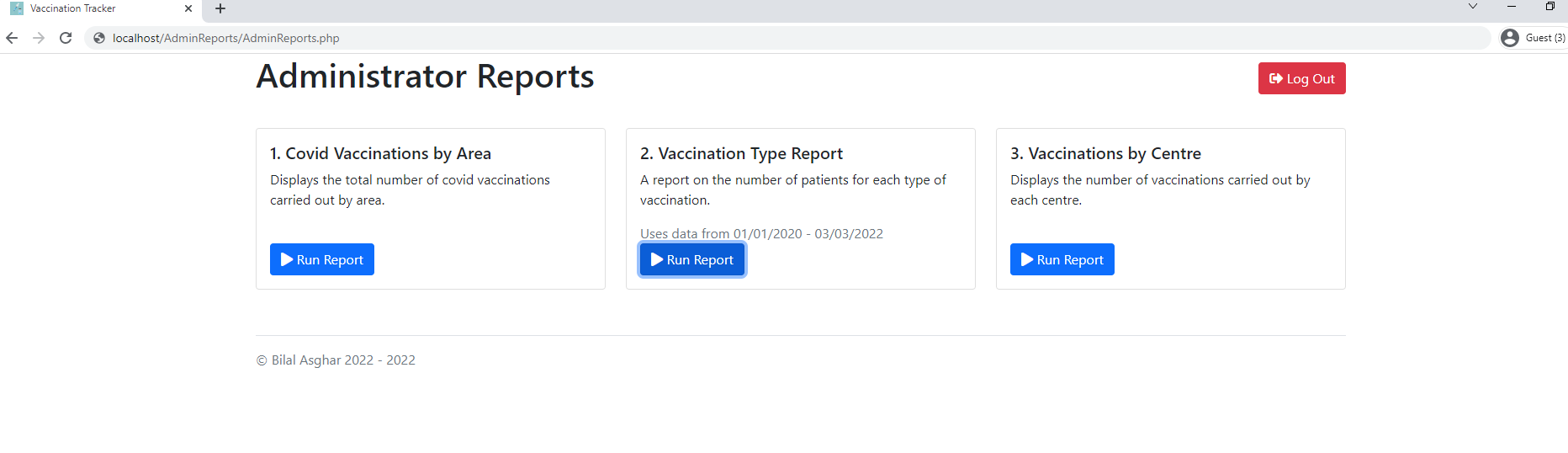
Report 1 button being clicked

Output:

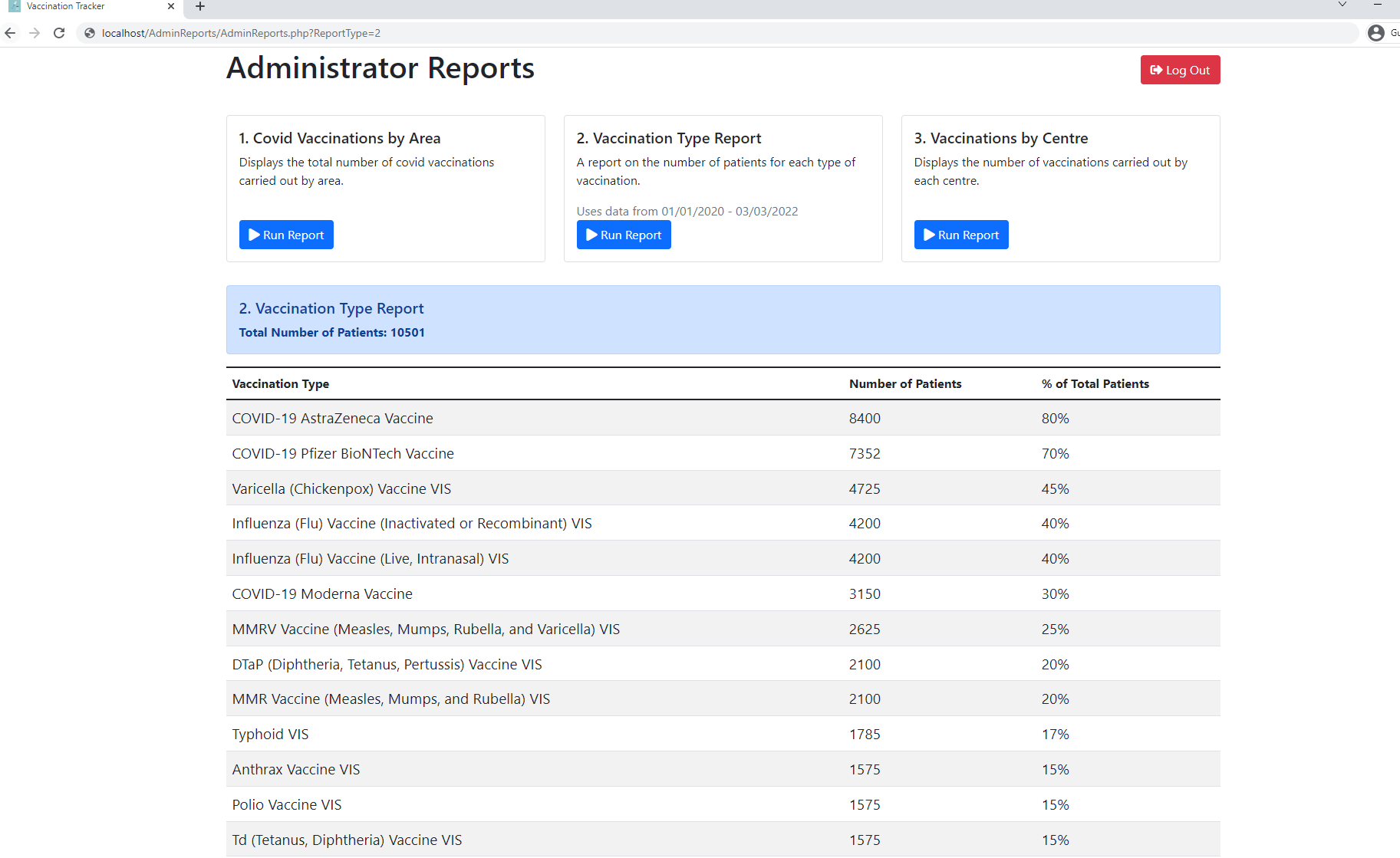
Report 1 being displayed

Run Report 2 (Normal)

Input:

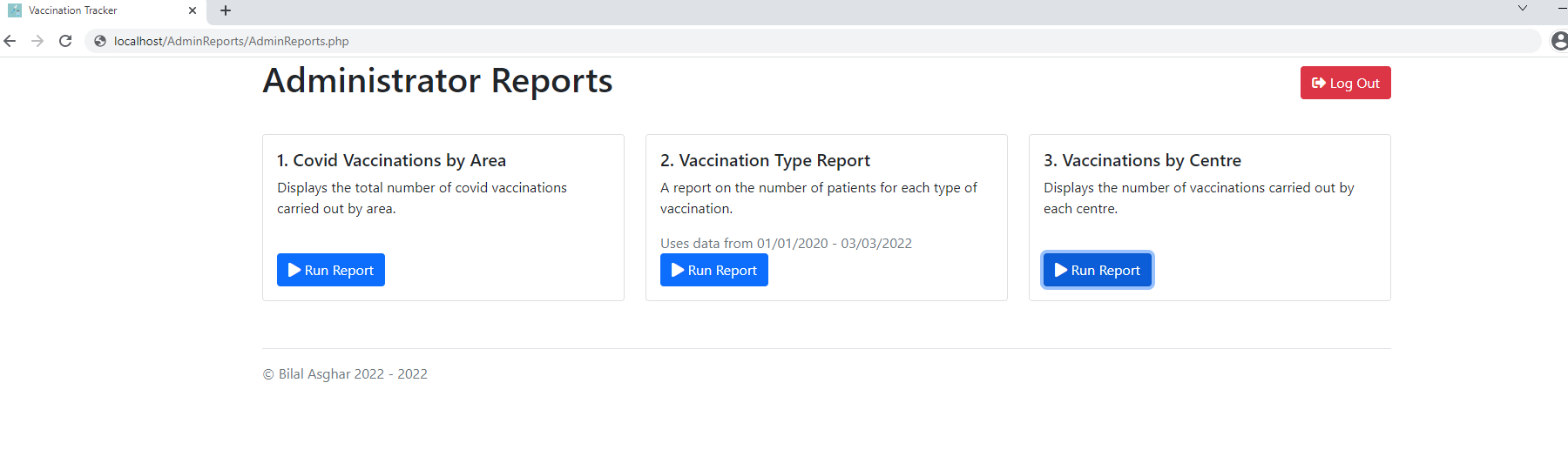
Report 2 button being clicked

Output:

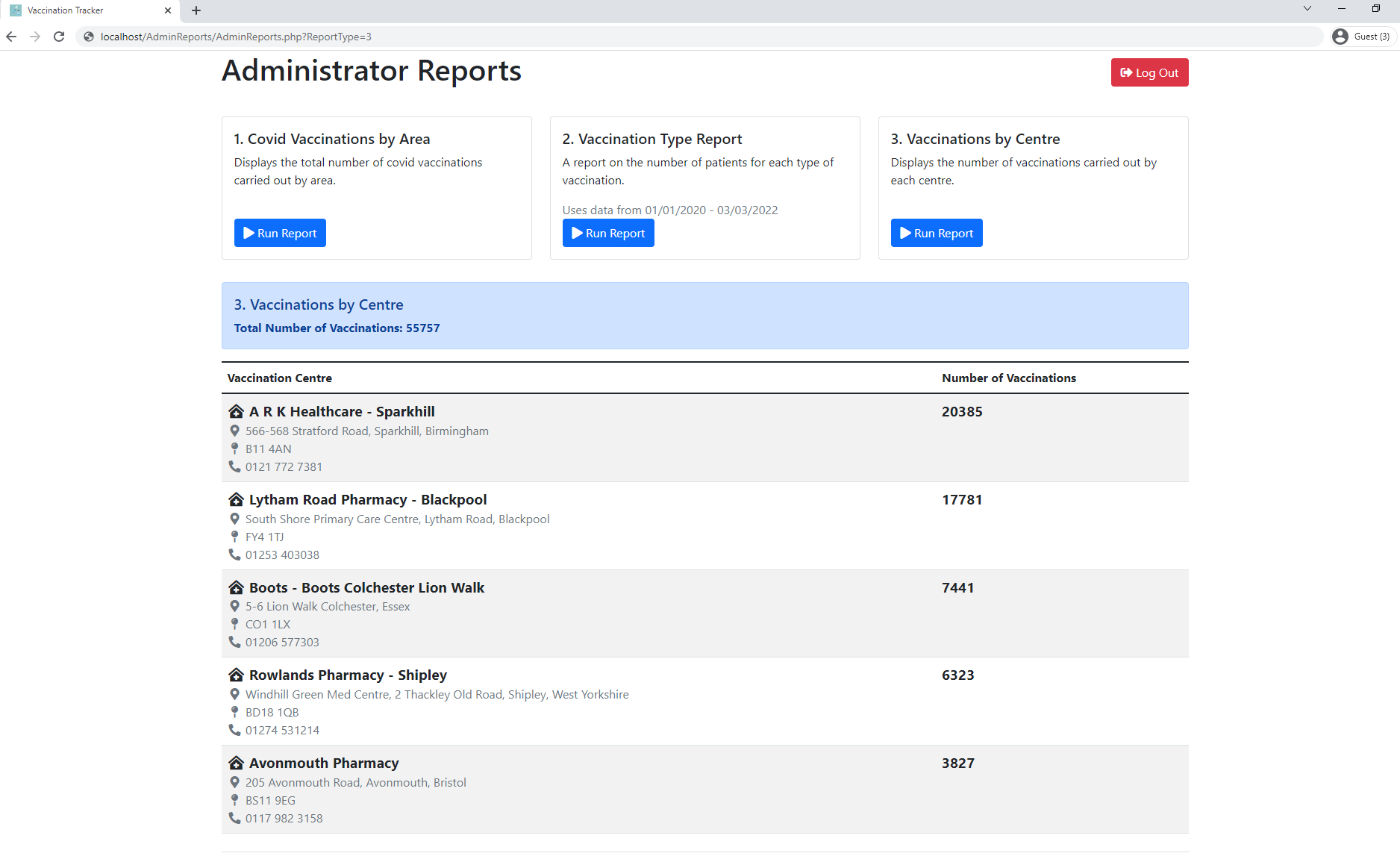
Report 2 being displayed

Run Report 3 (Normal)

Input:

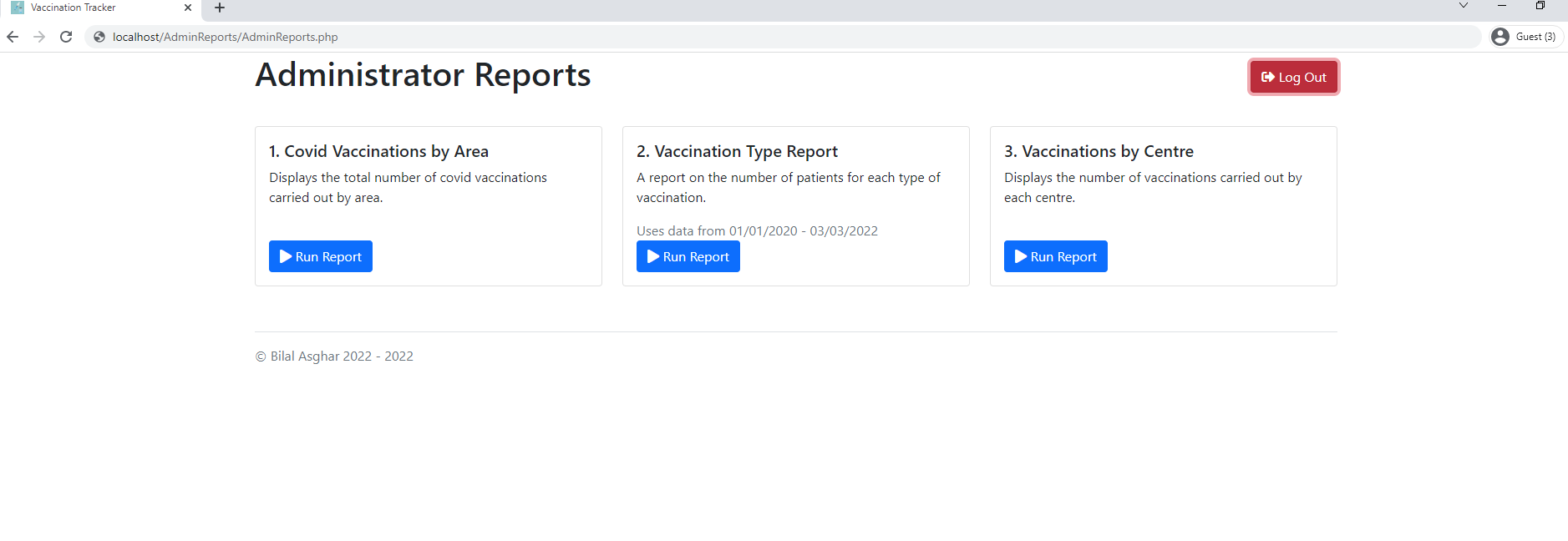
Report 3 button being clicked

Output:

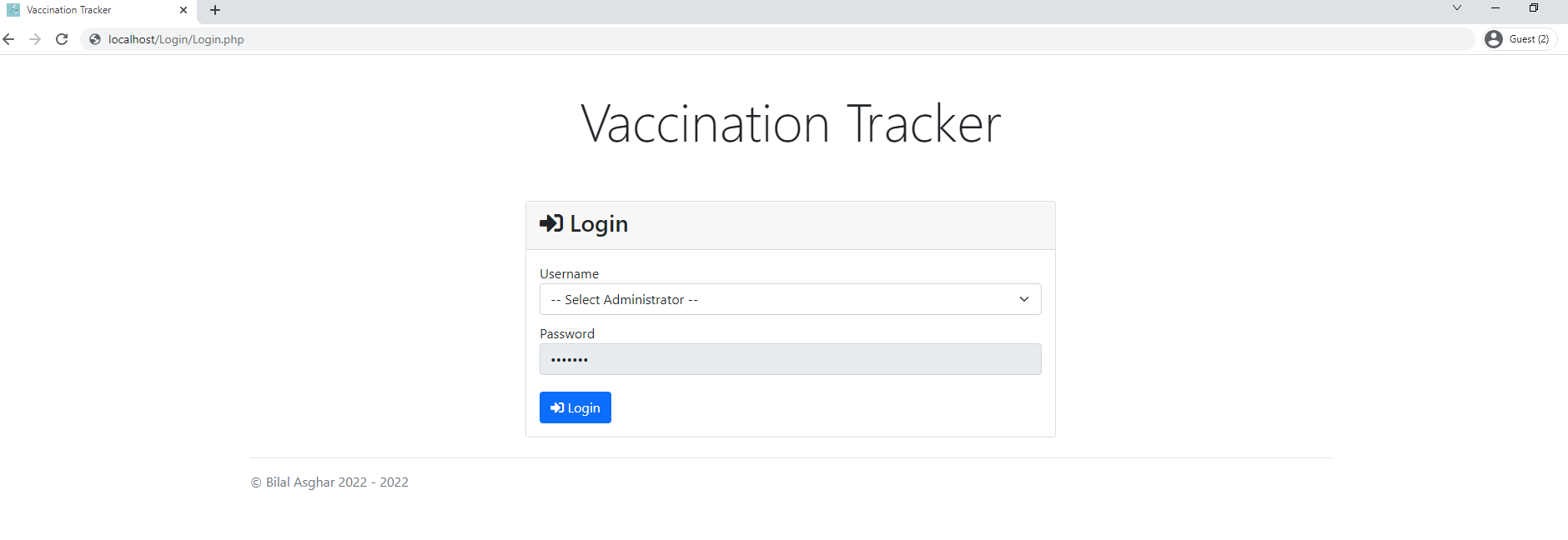
Report 3 being displayed

Log Out (Normal)

Input:

Log out button being clicked

Output:

Directed back to Login Page

## Test Restults

Medical Person:

| Test Case | Actual Result |
| --- | --- |
|  |  |
| Login | |
| Login (Normal) | Select vaccinator from Username drop down field. |
| Login (Exceptional) | User is redirected back to login page. |
| Vaccinate Patient Form | |
| Vaccinate Patient Form (Normal) | Successfully inserted data into database and displayed success message to user. |
| Vaccinate Patient Form (Exceptional) | Error message is displayed and no data inserted into database. |
| Vaccination History | |
| Vaccination History (Normal) | Displayed total number of vaccinations carried out and all patient vaccinations within table. |
| Update existing patient details | |
| Update existing patient details (Normal) | The database updated the last name of the patient with the patient identifier of 12345678 to Bruno. The new patient vaccination is also added to the PatientVaccination table within the database. In the Vaccination History page there is a new patient vaccination added to the table and the last name of the previous patient vaccination is changed to Bruno. |
| Log out | |
| Log out (Normal) | Log out is successful and user is redirected back to login page and are unable to go back to the previous page after logging out. |

Administrator:

| Test Case | Actual Result |
| --- | --- |
|  |  |
| Login | |
| Login (Normal) | Select vaccinator from Username drop down field. |
| Login (Exceptional) | User is redirected back to login page. |
| Run Report 1 | |
| Vaccinate Patient Form (Normal) | Displayed total number of covid vaccinations by area the report used. A table with the Area, Total Administered and Vaccine Breakdown data is displayed. |
| Run Report 2 | |
| Vaccination History (Normal) | Displayed total number of patients vaccinated the report used. A table with the Vaccination Type, Number of Patients and Percentage of Total Patients data is displayed. |
| Run Report 3 | |
| Update existing patient details (Normal) | Displayed total number of vaccinations carried out the report used. A table with the Vaccination Centre and Number of Vaccinations for each centre data is displayed. |
| Log out | |
| Log out (Normal) | Log out is successful and user is redirected back to login page and are unable to go back to the previous page after logging out. |

# Evaluation of the solution

## Fitness for Purpose

After testing my project, I know that it meets all of the end user and functional requirements. The table fields were validated in the patients table where the character length had to be greater than 0 so no empty values could be inserted. Also, my project made use of a database with 5 related tables: MedicalPersons, Patients, VaccinationCentres, VaccinationTypes and PatientVaccinations. I incorporated two advanced higher concepts a query across 5 tables for saving the patient vaccination data and the between operator for Vaccination Type Report to display the data from 01/01/2020 to the current date. I integrated my database with a web user interface that took in inputs to insert data into the database and execute queries to display the data from the database. My web interface executes the queries effectively as I used stored procedures which means that in the php code I only needed to call the stored procedure name and not write out the entire SQL query in the php file. After testing the solution with Test Cases, it showed that my Personas were able to successfully complete the list of tasks given with no problems or confusion. With all of this I am certain that my project meets all of the requirements stated in the functional and end user requirements.

## Maintainability

I feel that my solution is very maintainable as I have used meaningful field names for each of the five database tables. I have used stored procedures so that in the php code I only need to call the name of the stored procedure as a function instead of writing out each query inside the php file. Making the code very readable and easy to follow.

Inside the php code I used white space to separate out the code and used comments to explain and separate sections.

For example, within VaccinatePatient\_Content.php I put a comment saying what each form input represented. This makes it easier to differentiate with the rest of the form inputs and easier to navigate the code.



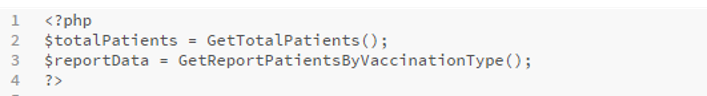
I created a separate php file to handle the database side of things called DatabaseHelpers.php. This set the connection to the database and ran the queries. I used functions to call each stored procedure making my code modular and separated.





I also used meaningful variable names within my php code to make it clear what each line of code was doing.

For example, in ReportType\_2 I used $totalPatients variable to hold the total number of patients and $reportData to hold the report 2 data. This clearly shows what each variable is doing and contributes to the clarity of the code



## Robustness

All inputs into database and web user interface are validated so that the user does not run into any unexpected errors.

When trying to go back to the previous page after logging out, the php code checks if the medicalPersonId session variable is null and the current page is the login page (Login.php) then it will keep the user on the login page.



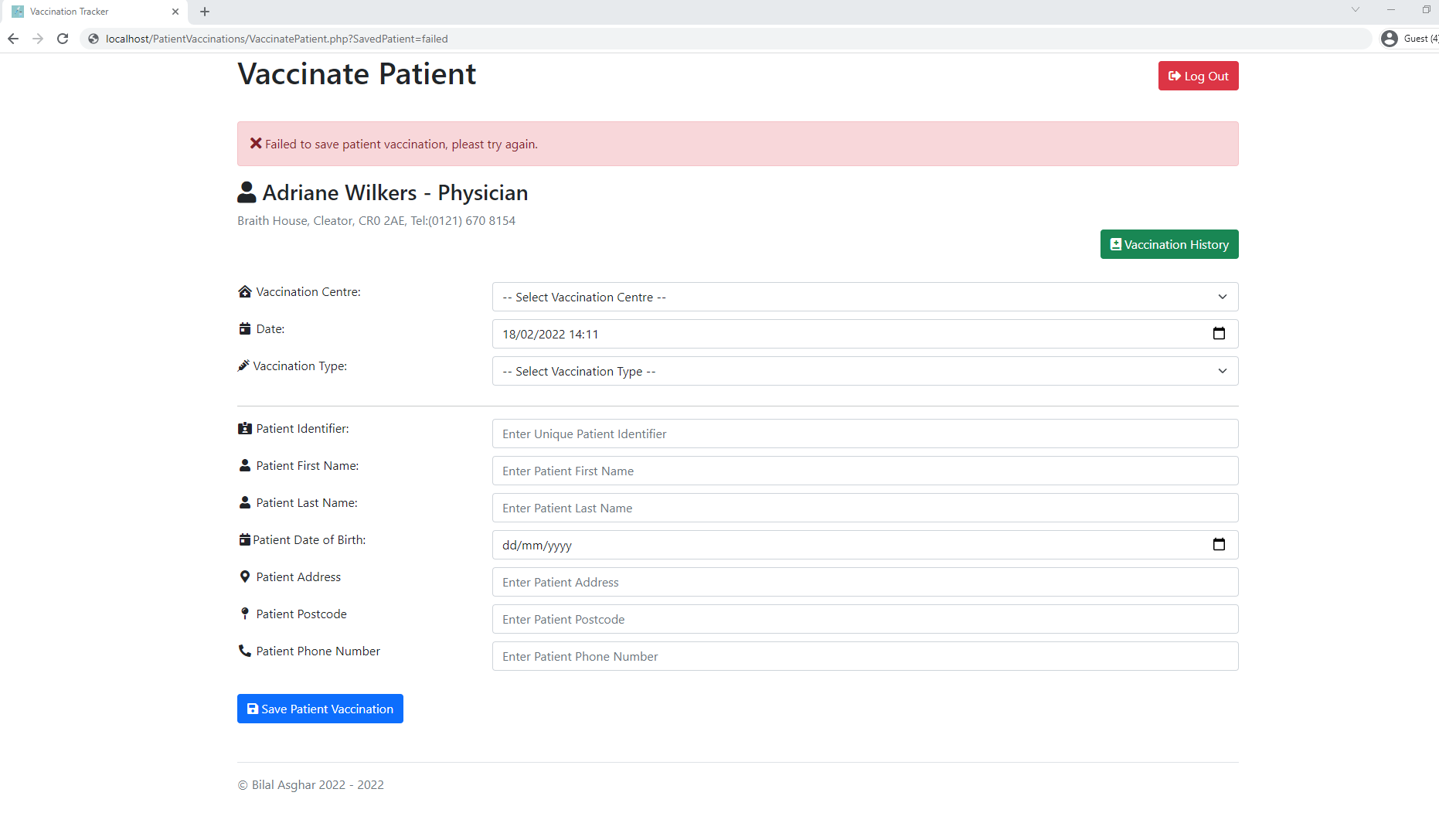
Also, when logging in the Login\_Athenticate.php checks if a valid username has been selected and if not the user will be redirected back to the login page (Login.php).



When entering data into the vaccinate patient form, all the fields have a required attribute on it, so the user has to enter data. Also, when the Save Patient Vaccination button is clicked the SavePatientVaccination DatabaseHelper function is called which runs the spSavePatientVaccination stored procedure. The data that is passed through will also be validated by the database making sure that there are no empty data being inserted as the character length has to be greater than 0. To inform the user if the data was saved or not into the database I wrote some php code that would display a success message or error message depending on if there were any errors or not. This helped keep my project robust as I had validation in place and error messages to inform the user if any were to occur.



Here is an example of an error message being displayed when blank spaces are entered and does not enter any data into the database:



## Overall Evaluation

Finally, I am confident that my implementation matches my analysis and design as seen from my testing and evaluation. I have made sure this project was completed by the deadline the 4th of March 2022. I have complied with the Copyright, Design and Patents Act 1988 as the name of my project is “Vaccination Tracker” and the data used was sample data. Also, I used Font Awesome which is **FREE** and open-source icon library. Bootstrap a **FREE** open-source CSS framework. There were no costs accumulated while making of this project as I only used free resources. Therefore, with all this taken into consideration I feel that my project matches the analysis and design sections.