|  |
| --- |
| Name: N. S. De Alwis |
| Student Reference Number: 10707160 |



|  |  |  |
| --- | --- | --- |
| Module Code: PUSL3111 | Module Name: API Software Development | |
| Coursework Title: API Coursework | | |
| Deadline Date: 11th May 2022 | | Member of staff responsible for coursework: Dr. Rasika Ranaweera |
| Programme: BSc (Hons) Plymouth Software Engineering | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  J.A. Mujeeb – 10707284  G.M.D.D. Ratnayake – 10707351  S.O. Perera – 10707315  N. S. De Alwis – 10707160  M. D. A. Medhavi – 10707278  P. P. L. Dilhani – 10709402  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: N. S. De Alwis | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed : | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |

* \*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

Table of Contents

[2. Acknowledgement - 5](#_Toc102899312)

[2.1 Abstract 5](#_Toc102899313)

[2.2 Introduction 5](#_Toc102899314)

[2.2.1 Quick Start Guide (figure out proper order) 5](#_Toc102899315)

[3. Project Scope 7](#_Toc102899316)

[3.1 Citizens 7](#_Toc102899317)

[3.2 Bureau Officers (acts as an admin) 7](#_Toc102899318)

[3.3 Foreign Companies Employers 7](#_Toc102899319)

[4. System Architecture 7](#_Toc102899320)

[4.1 Data Flow Diagram Level 0 7](#_Toc102899321)

[4.2 Data Flow Diagram Level 1 8](#_Toc102899322)

[4.3 Use Case Diagram 8](#_Toc102899323)

[4.4 Class Diagram 9](#_Toc102899324)

[4.5 Entity Diagram 9](#_Toc102899325)

[4.6 Flow Chart 10](#_Toc102899326)

[5. API Documentation 11](#_Toc102899327)

[5.1 POST /citizens 11](#_Toc102899328)

[5.2 PUT /citizens/:nid 11](#_Toc102899329)

[5.3 GET /citizens/:nid 11](#_Toc102899330)

[5.4 PUT /citizens/:nid 12](#_Toc102899331)

[5.5 GET /citizens/find(:qualifications) 12](#_Toc102899332)

[5.6 DELETE /citizens/:nid 12](#_Toc102899333)

[5.7 GET /citizens/:nid/contacts 12](#_Toc102899334)

[5.8 JSON Requests and Responses 12](#_Toc102899335)

[5.9 XML requests and responses 12](#_Toc102899336)

[6. Tools and technologies 12](#_Toc102899337)

[6.1 PhpMyAdmin 12](#_Toc102899338)

[6.2 Laravel 13](#_Toc102899339)

[6.3 PHP 13](#_Toc102899340)

[6.3.1 Why did we choose PHP? 13](#_Toc102899341)

[6.4 XML 13](#_Toc102899342)

[6.5 Java 13](#_Toc102899343)

[6.6 JavaScript Object Notation (JSON) 13](#_Toc102899344)

[7. Sri Lanka Bureau of Foreign Employment (SLBFE) 14](#_Toc102899345)

[7.1 Client Mobile Application Registration 14](#_Toc102899346)

[7.2 Client Website features 17](#_Toc102899347)

[7.3 Admin website features 18](#_Toc102899348)

[8. Individual contributions 19](#_Toc102899349)

[8.1 N. S. D. Alwis (10707160) 19](#_Toc102899350)

[8.2 G. M. D. D. Rathnayake (10707351) 19](#_Toc102899351)

[8.3 S. O. Perera (10707315) 19](#_Toc102899352)

[8.4 P. P. L. Dilhani (10709402) 19](#_Toc102899353)

[8.5 J.A. Mujeeb (10707284) 20](#_Toc102899354)

[8.6 M. D. A. Medhavi (10707278) 20](#_Toc102899355)

[9. Team Contribution 21](#_Toc102899356)

[Figure 1: Type "php artisan migrate" 6](#_Toc102899357)

[Figure 2: Type "php artisan migrate" 6](#_Toc102899358)

[Figure 3: Type "php artisan serve –port=8001" 6](#_Toc102899359)

[Figure 4: database configuration 6](#_Toc102899360)

[Figure 5: Database Table 7](#_Toc102899361)

[Figure 6: Context Diagram 8](#_Toc102899362)

[Figure 7: Data flow diagram - Level 1 8](#_Toc102899363)

[Figure 8: Use case Diagram 9](#_Toc102899364)

[Figure 9: Class Diagram 9](#_Toc102899365)

[Figure 10: Entity Diagram 10](#_Toc102899366)

[Figure 11: Flow chart 10](#_Toc102899367)

[Figure 12: Login page in website 14](#_Toc102899368)

[Figure 13: Mobile App Registration 14](#_Toc102899369)

[Figure 14: Toast for confirmation 15](#_Toc102899370)

[Figure 15: Mobile App Login page 15](#_Toc102899371)

[Figure 16: Mobile app Vacancy screen 16](#_Toc102899372)

[Figure 17: Citizen's profile updating page 17](#_Toc102899373)

[Figure 18: Complaints page 17](#_Toc102899374)

[Figure 19: Job seekers 17](#_Toc102899375)

[Figure 20: Admin side - create citizen profile 18](#_Toc102899376)

[Figure 21: Replying complaints 18](#_Toc102899377)

[Figure 22: View Job seeker page 19](#_Toc102899378)

[Figure 23: New Vacancy 19](#_Toc102899379)

[Figure 24: Manage Users 19](#_Toc102899380)

**No table of figures entries found.**

# Acknowledgement -

First and foremost, we’d like to extend our sincere gratitude towards Dr. Rasika Ranaweera, our module lecturer. We are extremely humbled and grateful to have been able to receive his mentorship, guidance, and support.

The overall accomplishment of this project demanded a significant amount of guidance from many individuals. As a team, we are extremely fortunate to have had this from start to finish.

Finally, we wouldn’t have been able to successfully complete this assignment without the hard work and assistance of all the team colleagues itself. We all enjoyed working with each other.

## Abstract

This assignment comprises of a documented report on an API Web Application and Mobile Application, pertinent to the Sri Lanka Bureau of Foreign Employment (SLBFE).

Coursework Supervisor: Dr. Rasika Ranaweera

Group: Group No. 22

Deadline: 11th of May 2022

## Introduction

The Sri Lanka Bureau of Foreign Employment (SLBFE) is a government owned foreign employment agency that assists its citizens in the procurement of global employment opportunities.

It is evident that labour is a fundamental asset that would greatly contribute towards substantial job creation, playing a key role in securing a stable source of foreign currency inflow. It also recognizes predicaments related unemployment, helping more Sri Lankans escape the risks of poverty, bestowing the potentiality to further enhance their prosperity and well-being.

Through the SLBFE, any citizen can hold the potentiality to acquire membership via free online registration. Those that are on the lookout for jobs can utilize the system to update and upload their qualifications, CVs, birth certificates, etc. Bureau Officers is authorized to view and validate the uploaded information. Furthermore, foreign companies and bureau officers can look for workers based on their respective qualifications.

Those who have left for foreign employment must update their real-time location once they go to the foreign company.

Citizens are also permitted to be able to post complaints, which bureau officers can parallelly view and respond to. We have developed a RESTful API web application to address the above scenario.

### Quick Start Guide (figure out proper order)

* Let’s start off by unzipping the folders containing the code [folder name]
* Choose a code ide of your choice (VS code recommended) and open up the SLBFE API and the SLBFE client-side application same goes for the mobile application.
* Once everything is set up let us import the relevant data into our computer {need to correct}
* By typing following commands
* To import the databases, we type the following**.**

**Graphical user interface, text

Description automatically generated**

Figure 1: Type "php artisan migrate"

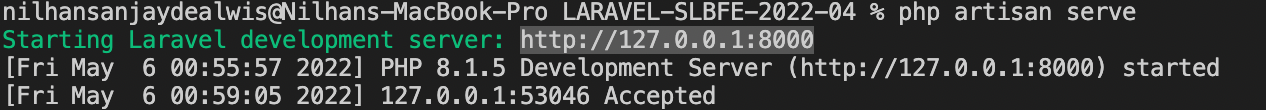
****

Figure 2: Type "php artisan migrate"

Next when running on the client application it needs to be served in a different port , since the API is running in port 8000 , the client application should run on port 8001 for that we type the following code

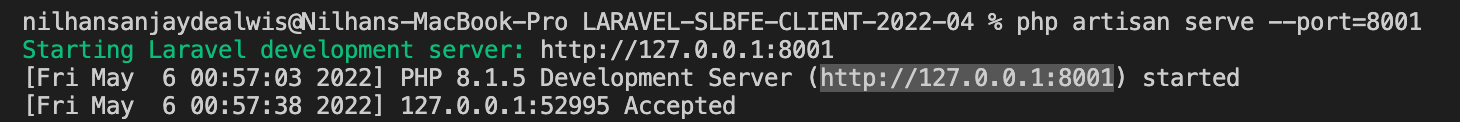


Figure 3: Type "php artisan serve –port=8001"

Now let us configure the database.



Figure 4: database configuration

After click create the following rows will be visible (check vid tutorial again)

Graphical user interface

Description automatically generated with low confidence

Figure 5: Database Table

# Project Scope

The project scope mainly caters towards bestowing the Sri Lankan citizens with productive employment opportunities.

## Citizens

Citizens possess the volition to Sign Up or Sign In through the Website or Mobile Application. They must fill in their login credentials to access their user account. Those who aren’t registered as a member must undergo the registration process. After logging in, they must update their respective qualifications and uploading their CV’s, birth certificate, etc. so that they can seek jobs offers. Citizens also could file a complaint, and bureau officers should be able to see it and respond appropriately. Once a citizen has left for foreign employment, they are required to update their current location as soon as they visit the company.

## Bureau Officers (acts as an admin)

Bureau Officers act as an admin, holding the power to be able to validate and review the profiles of the already created user accounts. They can also admin can block any citizen. Bureau Officers act like an intermediate between the Citizens and Foreign Companies. This is because they parallelly help the Citizens find suitable work outside of the country, while also helping Foreign Companies find appropriate workers.

Bureau Officers can also respond to any complaints made by the respective Citizens. The Bureau Officers also checks into the foreign employee’s overall safety and wellbeing.

## Foreign Companies Employers

Foreign Companies can select employees based on their qualifications.

# System Architecture

## Data Flow Diagram Level 0

A picture containing text

Description automatically generated

Figure 6: Context Diagram

## Data Flow Diagram Level 1

Diagram

Description automatically generated

Figure 7: Data flow diagram - Level 1

## Use Case Diagram

Diagram

Description automatically generated

Figure 8: Use case Diagram

## Class Diagram

Graphical user interface

Description automatically generated

Figure 9: Class Diagram

## Entity Diagram

Diagram

Description automatically generated

Figure 10: Entity Diagram

## Flow Chart

Diagram

Description automatically generated with medium confidence

Figure 11: Flow chart

# API Documentation

With the utilization of a RESTful API, we have successfully developed both the Web and Mobile Application. Furthermore, we have authorized a capability for Citizen’s to get themselves registered into the system’s Database through either the Web and Mobile Application.

Presented right below are the API’s constructed to suit the functional requirements of both the Web and Mobile Applications.

## POST /citizens

Both Citizens and Bureau Officers hold the capability to register themselves with following details:

* National ID (NIC)
* Name
* Age
* Address
* Current Location (latitude and longitude)
* Profession
* Email
* Affiliation
* Password, retype Password

Route::post('register', [AuthController::class, 'register']);

Route::post('login', [AuthController::class, 'login']);

## PUT /citizens/:nid

Job seekers should be able to update their qualifications and upload certificates.

Route::put('vacancy/add', [JobController::class, 'addVacancy']);

Route::put('location/add', [JobController::class, 'location']);

Route::put('complaints/add', [ComplaintsController::class, 'add']);

Route::put('complaints/reply', [ComplaintsController::class, 'reply']);

## GET /citizens/:nid

Bureau Officers is able to access any citizen’s information by their National ID (NIC).

Route::get('vacancy/list', [JobController::class, 'listVacancy']);

Route::get('qualification/get', [JobController::class, 'getQualifications']);

Route::get('qualification/search', [JobController::class, 'searchQualifications']);

Route::get('complaints/get', [ComplaintsController::class, 'get']);

Route::get('users/get', [UserController::class, 'get']);

## PUT /citizens/:nid

Bureau Officers can verify, validate and review the profiles of the already created user accounts.

## GET /citizens/find(:qualifications)

Foreign Company Officers can find candidates based on the Citizens respective qualifications.

## DELETE /citizens/:nid

SLBFE staff (Bureau Officers) are able to deactivate an individual’s account if the Citizen is deceased (or for any other suitable reason).

Route::delete('users/delete', [UserController::class, 'delete']);

## GET /citizens/:nid/contacts

SLBFE staff can collect information regarding contacts of any Citizen.

## JSON Requests and Responses

## XML requests and responses

# Tools and technologies

In pursuance of securing a successful project outcome, we have contended in carefully choosing the most suitable tools and technologies.

When it comes to developing the API, we have utilized PHP based Laravel framework. The Web Application also uses the same framework.

We have used Android Studios to create our mobile application. Android Studios uses SDK (specified for android technologies). Furthermore, we have also selected Java as the coding language for the Mobile Application. With the use of Android Studios, developers like us can successfully create high-performance products. Developers can also build emulators in accordance to their preference. These emulators are nimble and rich in features. Various testing mechanisms can also be deployed to test and maintain projects.

## PhpMyAdmin

As PHP is utilized to code the Web Application, it was more coherent to utilize PhpMyAdmin for MySQL database management. PhpMyAdmin is an open-source third-party software tool, written in PHP. The primary objective of utilizing phpMyAdmin is to superintend and operate the administration of MySQL over the web. It is also feasible to run CRUD operations like databases, copy, tables, rename, databases, tables, columns, etc.

phpMyAdmin can run on any server or any OS as it has a web browser.

With the utilization of phpMyAdmin, it is possible to edit, create or delete the database without much difficulty. In collation to the MySQL command-line editor, it is easier to manage elements with the utilization of the phpMyAdmin graphical interface. Several servers can also be operated simultaneously. Data can also be exported into various formats like Word, PDF, SQL, XML, Spreadsheet, etc.

## Laravel

Laravel is a PHP framework that adheres to an MVC design pattern. It is possible to increase the overall web application scalability with the Laravel framework. The utilization of Laravel framework has been deemed time-saving, as the existing framework components that aid in the creation of the web application is reused.

## PHP

The general purpose of PHP is that it’s an open-source scripting language that is well suited for the web development whether they are web application, dynamic websites or static websites, and the code can be embedded to HTML in the server-side.

### Why did we choose PHP?

One of main reason why our team choose PHP is that it is platform independent, meaning it can be used on windows, Linux, macOS and also supports almost all web browser. It also works with all major web servers, making it simple to deploy on a variety of systems and platforms at a low cost.

PHP makes it easier to connect to almost any databases. This provided us more flexibility in deciding which database would be ideal for the application we were developing.

One of the other main reasons why we chose PHP is that it is very simple to get started with. Even without extensive knowledge or experience in the web development, our team could create a web page using PHP in short period of time. The syntax seems to be simple and learning the function is not hard, which means chances of errors are lower in PHP than other languages.

## XML

XML is a markup language that defines a set of rules for data encoding in a document that can be understood by both humans and machines.

Simplicity, generality, and usability and Its design goals

It simplifies data transfer, shared storage, data compatibility etc. It allows extending or switching to new software, operating systems (OS) without missing a single detail. XML can be defined as knowledge enclosed within identifiers. (force, 2014)

## Java

Java was been utilized in the Mobile Application of our project due to its robustness, ease of use, and cross-platform development competencies. It is secure, straightforward and easy to comprehend, making code implementation relatively easier.

As it is an Object-Oriented Programming Language, OOPs concepts like abstraction, encapsulation and inheritance help in the enhancement of security. It is also compatible and platform-independent, meaning that code written once can be run anywhere else (on other systems that also possess Java). As it is a High-Level Programming Language, it has fewer and simpler syntaxes.

## JavaScript Object Notation (JSON)

Standard text-format that is used to act for the purpose of transmitting data within web applications. It is a language-independent format that is easy for humans to understand (read and write) and it is also uncomplicated when parsing data.

# Sri Lanka Bureau of Foreign Employment (SLBFE)

Shown below is the Login Page. This Login Page is common to both the Admin and Citizens.

**Graphical user interface, application

Description automatically generated**

Figure 12: Login page in website

## Client Mobile Application Registration

The Citizen/User can get registered (free of charge).

**A screenshot of a computer

Description automatically generated with low confidence**

Figure 13: Mobile App Registration

Once the registration is completed and the process is successful, the “Registration Successful! Please Login” will be temporarily displayed.

**Graphical user interface, application

Description automatically generated**

Figure 14: Toast for confirmation

Here, in the Mobile Application, any successfully registered Citizen can enter their credentials to Log In.

**Graphical user interface, application

Description automatically generated**

Figure 15: Mobile App Login page

Shown below is a Vacancy Screen that any registered Citizen/User will first see when logged in. They can seek and view available vacancies.

**Graphical user interface, text, application

Description automatically generated**

Figure 16: Mobile app Vacancy screen

## Client Website features

Here, on the Website, the Citizen/User will first see their profile when logged in. They can update their credentials by using their profile page. They also have the ability to file complaints. Furthermore, there is a seekers page and a logout option.

**Graphical user interface, website

Description automatically generated**

Figure 17: Citizen's profile updating page

**Graphical user interface, text

Description automatically generated**

Figure 18: Complaints page

**Graphical user interface, text, application

Description automatically generated**

Figure 19: Job seekers

## Admin website features

**Graphical user interface, text, application, email

Description automatically generated**

Figure 20: Admin side - create citizen profile

**Graphical user interface, application

Description automatically generated**

Figure 21: Replying complaints

**Graphical user interface, text, application, email

Description automatically generated**

Figure 22: View Job seeker page

**Graphical user interface

Description automatically generated**

Figure 23: New Vacancy

**Graphical user interface, text, application, email

Description automatically generated**

Figure 24: Manage Users

# Individual contributions

## N. S. D. Alwis (10707160)

## G. M. D. D. Rathnayake (10707351)

## S. O. Perera (10707315)

## P. P. L. Dilhani (10709402)

## J.A. Mujeeb (10707284)

Text

Description automatically generated

## M. D. A. Medhavi (10707278)

# Team Contribution

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Contribution |
| J.A. Mujeeb | 10707284 | 16.7% |
| G.M.D.D. Ratnayake | 10707351 | 16.7% |
| S.O. Perera | 10707315 | 16.7% |
| N. S. De Alwis | 10707160 | 16.7% |
| M. D. A. Medhavi | 10707278 | 16.7% |
| P. P. L. Dilhani | 10709402 | 16.7% |

Table 1: Team contribution table

# Bibliography

force, i. e. (2014). *XML .*