

(NIT5130) DATABASE ANALYSIS AND DESIGN

Lecturer : Dr. Ashkan Yousefi

Contributors

S458459 Alev Akturk

S4598386 Jinxi Han S4580349 Makara Vann S4584593 Nguyen Truong Anh

Table of Contents

[1. Introduction 3](#_Toc11015301)

[1.1. Purpose: 3](#_Toc11015302)

[1.2. Summary of The Business Case: 3](#_Toc11015303)

[1.3. Limitations: 4](#_Toc11015304)

[1.4. Assumptions: 4](#_Toc11015305)

[2. Database Modelling 5](#_Toc11015306)

[2.1. Software and Hardware Requirements 5](#_Toc11015307)

[2.2. Business Rules 6](#_Toc11015308)

[2.3. Entity Relationship Diagram 7](#_Toc11015309)

[2.4. Normalization of ERD 8](#_Toc11015310)

[2.5 Data dictionary 9](#_Toc11015311)

[3. Database Implementation 14](#_Toc11015312)

[3.1. Creation of the ERD in PhPMyAdmin 14](#_Toc11015313)

[3.2. Forms and Reports 20](#_Toc11015314)

[3.3. Codes 23](#_Toc11015315)

[4. Appendix 27](#_Toc11015316)

[5. References 27](#_Toc11015317)

# 1. Introduction

# 1.1. Purpose:

In this report, we will elaborate how we design the retrieval webpage for students in the Victoria University to enjoy International Office service. The report comprises two main components – database modelling and database implementation.

In database modelling, we explain the software and hardware infrastructure for this project. We demonstrate the Entity Relationship Diagram (ERD) and describe our logic. To support our idea, we also describe business rules according to business case and insert in a table for better understanding. Then, we attach our normalization of ERD to the report with data dictionary for users to understand.

On the other hand, database implementation part includes how we implement the database we design in real-life scenario. Screenshots are listed below to prove that our database can run properly.

# 1.2. Summary of the Business Case:

The database is designed for International Office (IO) of Victoria University to assist simple and fast data entry and print reports. IO intends to create a public space for students where they can meet other students, take a rest and have a chat. However, IO staff are concerned with the maintenance of such an area. They plan to set up a roaster for the student members. These rosters are generated from the database of students wishing to volunteer for a particular job or jobs.

In this case, the database should have certain features in order to meet with IO’s requirements. Since key information is collected from the students during their enrolment, the whole process should be fast and easy-to-use. Furthermore, the director of IO requires reporting functions in order to help organising a roaster for the volunteers. Therefore, the data entry should be as capable as possible.

# 1.3. Limitations:

When designing the database and writing SQL codes, we find that first of all, from the point of processing, WordPress can be unstable and slow down the internet connection, which can suspend our database design. Secondly, since localhost is located on a certain computer, other users cannot access to the database system remotely. In this way, the website function is refined in a certain computer, largely undermining its capacity. Thirdly, database storage is limited within our own computer. Much data we store, slower our computer will be.

## 1.4. Assumptions:

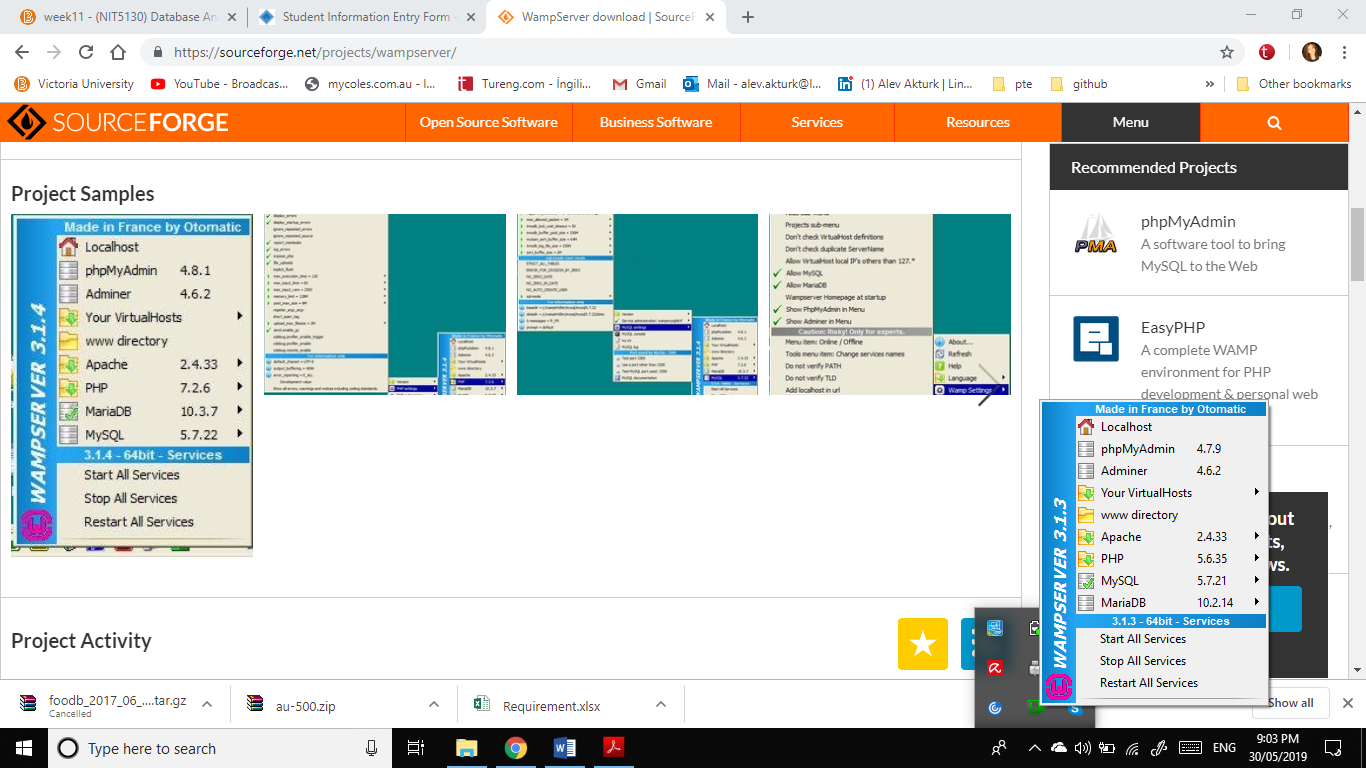
We assume that the database is designed for the following scenario. Victoria University has am International Office that requires a website to record the student information and arrange a roster of volunteers for maintenance work. We have to make a database have the capacity of 2000 students and 100 teachers. According to this, we need to create 10 tables covering all the relevant information, which are STUDENT, COURSE, FACULTY, STAFF, IO\_OFFICE, FACILITY, ROSTER\_TABLE, NUTRITION, POSTAL\_CODE,and LECTURE.Depending on the data, we will design ‘Student information entry form’, ‘Student information update form’,and ‘Job Allocation Form’ for users to retrieve the information they want to know.

Definition of all abbreviations in a table

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| Database (DB) | An organized collection of data, generally stored and accessed electronically from a computer system. |
| DBMS | A **database management system** (**DBMS**) is system software for creating and managing databases. |
| ERD | An **entity relationship diagram** (ERD) shows the **relationships** of **entity** sets stored in a database. |
| SQL | Structured Query Language (**SQL)** is a language used to communicate with a database. |
| MySQL | MySQL is an open-source relational database management system. |
| PHPMyAdmin | phpMyAdmin is a free and open source administration tool for MySQL and MariaDB. |
| HTML | Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. |
| WordPress | WordPress is a free and open-source content management system based on PHP & MySQL. |

# 2. Database Modelling

# 2.1. Software and Hardware Requirements

Wamp Server has been used to create web application and to connect phpMyAdmin to manage our database easily (WampServer, 2019). It provides web development platform and allows us to use local host. Hardware requirements for Wamp Server (WampServer, 2019):

Operating System: Windows XP (32 or 64 bits)

HDD: 200 MB free space

Memory: 512 MB RAM

Full administrator Access PhpMyAdmin database system is useful software tool. It not only supports relational database model but also it provides MySQL features. In addition, WordPress can be directly connected phpMyAdmin and store all posts, forms, reports database. Apart from storing database, it also manages the databases by deleting, editing, updating multiple tables. Another database software requirement is creation of primary key, index of multiple columns, which are supplied by phpMyAdmin software tool (phpMyAdmin, 2019).

WordPress is an amazing free tool which provides us to build own website beyond imaginations (WordPress, 2019). WordPress has been used for visualizing forms and reports for the assignment purposes. Furthermore, it allows us to run php script codes to create forms. Every information from WordPress has been stored phpMyAdmin.

# 2.2. Business Rules

|  |  |
| --- | --- |
| **BUSINESS RULE** | **RELATIONSHIP WITH ERD** |
| A student can only register for one volunteer job. | 1->1 |
| A student can have more than one choice of nutrition requirements. | 1->N |
| A student enrolls in more than one course and each course can be enrolled by more than one students. | N->M |
| A student can use more than one facility and each facility can host many students. | N->M |
| All volunteers belong to the same IO Office. | N->1 |
| More than one course are in one faculty. | N->1 |
| Many courses can be taught by one lecturer. | N->1 |
| Many lecturers are in one faculty. | N->1 |
| More than one staff can work in different faculties. | N->M |
| More than one staff work in the same IO office. | N->1 |
| IO office has many facilities. | 1->N |

# 2.3. Entity Relationship Diagram

There is a complex relationship among various entities within this application. The design of all DBMS tables has the relationship as below:

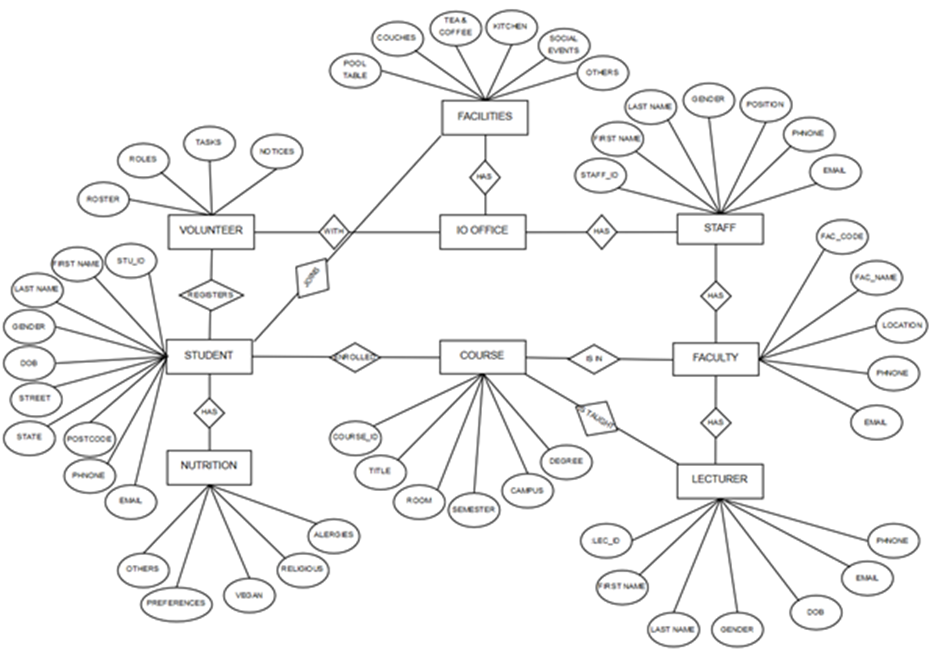
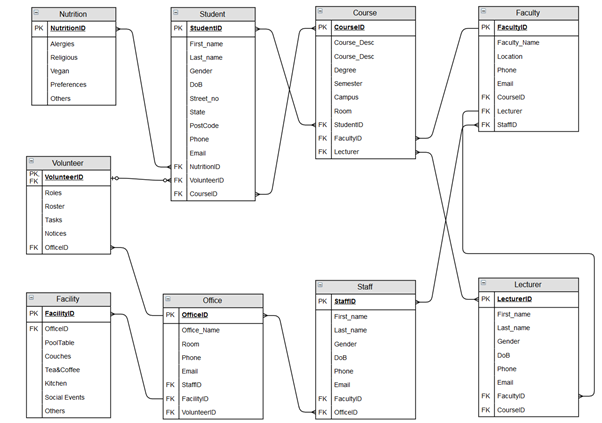
This model is using Crow’s foot ERD model as presented below.

Figure 2.2.1 ERD Diagram

  
Figure 2.2.2 Crow’s Foot ERD

# 2.4. Normalization of ERD

In this part we will try to restructure the relational database table in accordance with the series of rules which we can normal forms with the purpose to reduce data redundancy while improving the data integrity. The process begins by the first normal form, which we will ensure that all the values in each single column of a table are atomic.

In the second normal form, we will make sure that all the columns depend on the table’s primary key.

Moving to the last 3rd normal form, we will make sure there is no transitive dependencies between columns. For example, in the ERD table we can see the relationship between postal code and state because if we change the postal code city would change as well so we have to make another table where postcode is the primary key.

# 2.5. Data dictionary

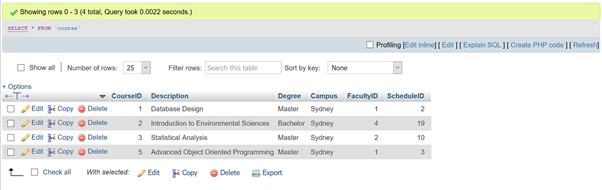
The table below is the data dictionary table which containing all the records about the object in the database and data descriptions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | **Attribute Name** | **Content** | **Data Type** | **PK or FK** |
| **Student** | StudentID | Student ID number | int | PK |
|  | First\_Name | First Name | Varchar(100) |  |
|  | Last\_Name | Last Name | Varchar(100) |  |
|  | Gender |  | Char(1) |  |
|  | Dob | Date of Birth | Date |  |
|  | Street\_no | Street number | Varchar(100) |  |
|  | PostCode | Postal Code | Num |  |
|  | Phone |  | int |  |
|  | Email |  | Varchar(100) |  |
|  | NutritionID | Nutrition ID number | int | FK |
|  | VolunteerID | Volunteer ID number | int | FK |
|  | CourseID | Course Id number | int | FK |
| **Course** | CourseID | Course Id number | int | PK |
|  | Degree |  | text |  |
|  | Semester |  | Number |  |
|  | Campus |  | Varchar(100) |  |
|  | Room |  | Char(20) |  |
|  | StudentID | Student ID number | int | FK |
|  | FacultyID | Faculty ID number | int | FK |
|  | LecturerID | Lecture ID number | int | FK |
| **Faculty** | FacultyID | Faculty ID number | int | PK |
|  | Faculty\_Name | Faculty Name | Varchar(100) |  |
|  | Location |  | text |  |
|  | Fal\_Phone | Faculty Phone | int |  |
|  | Fal\_Email | Faculty Email | Varchar(100) |  |
|  | CourseID | Course ID number | int | FK |
|  | LecturerID | Lecturer ID number | int | FK |
|  | StaffID | Staff identification | int | FK |
| **Staff** | StaffID | Staff identification | int | PK |
|  | Staff\_Email | Staff Email | text |  |
|  | Staff\_Phone | Staff Phone Number | Number |  |
|  | FacultyID | Faculty Identification number | int |  |
|  | OfficeID | Office ID number | int |  |
| **IO\_Office** | OfficeID | Office ID number | int | PK |
|  | Office\_Name | Office Name | Varchar(100) |  |
|  | Office\_Room | Office Room Number | Char(20) |  |
|  | StaffID | Staff Identification | int | FK |
|  | FacultyID | Faculty ID number | int | FK |
|  | VolunteerID | Volunteer ID number | int | FK |
| **Facility** | FacilityID | Facility ID number | int | PK |
|  | OfficeID | Office ID number | int |  |
|  | PoolTable |  | Boolean |  |
|  | Couches |  | Boolean |  |
|  | Tea&Coffee |  | Boolean |  |
|  | Kitchen |  | Boolean |  |
|  | SocialEvents |  | text |  |
|  | Others |  | text |  |
| **RosterTable** | RosterID | Roster Number | int | PK |
|  | Task\_Name | Task Name | Text |  |
|  | Notices |  | Text |  |
|  | Hours |  | int |  |
| **Nutrition** | NutritionID | Nutrition ID number | int | PK |
|  | Allergies |  | Text |  |
|  | Religious |  | text |  |
|  | Vegan |  | Boolean |  |
|  | Preferences |  | Text |  |
|  | Others |  | text |  |
| **PostalCode** | PostCode | Postal Code | Char(50) | PK |
|  | City |  | text |  |
|  | State |  | text |  |
| **Lecturer** | LecturerID | Lecturer ID number | int | PK |
|  | First\_Name | First Name | Varchar(100) |  |
|  | Last\_Name | Last Name | Varchar(100) |  |
|  | Dob | Date of birth | Date |  |
|  | FacultyID | Faculty ID number | Int | FK |
|  | CourseID | Course ID number | int | FK |

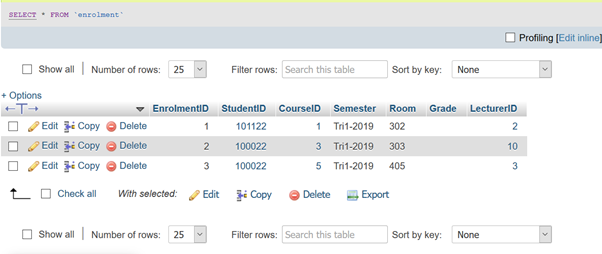
# 3. Database Implementation

# 3.1. Creation of the ERD in PhPMyAdmin

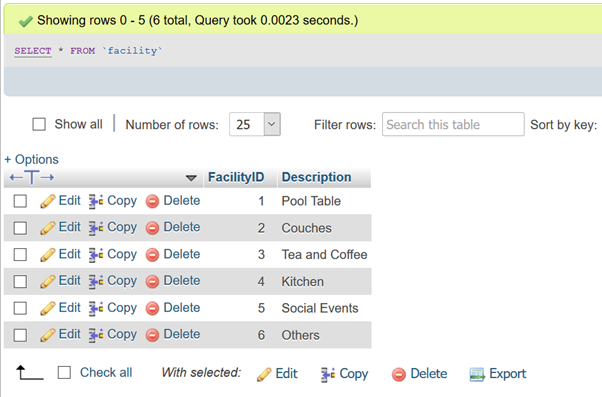
1. Course



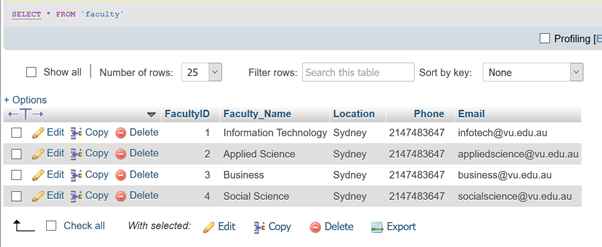
2. Enrolment



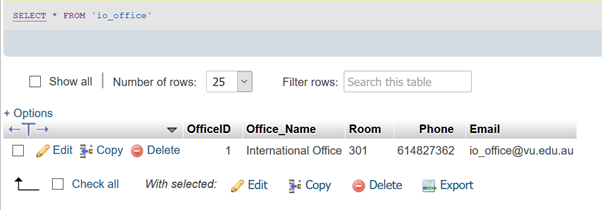
3. Facility



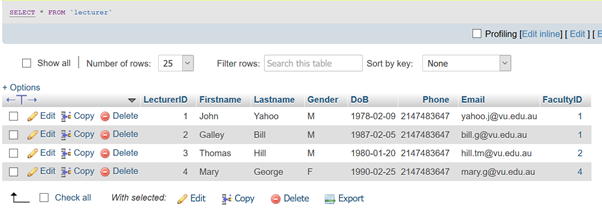
4. Faculty



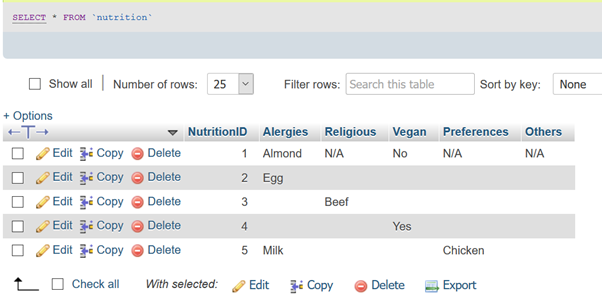
5. IO Office



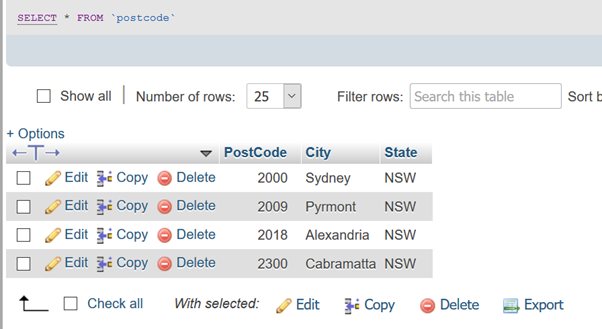
6. Lecturer



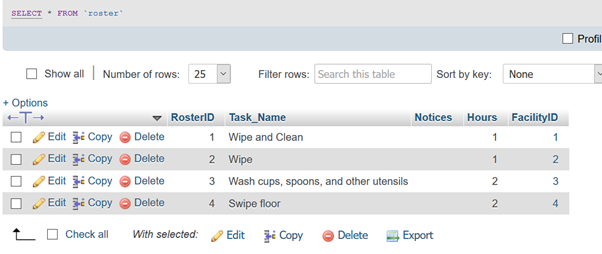
7. Nutrition



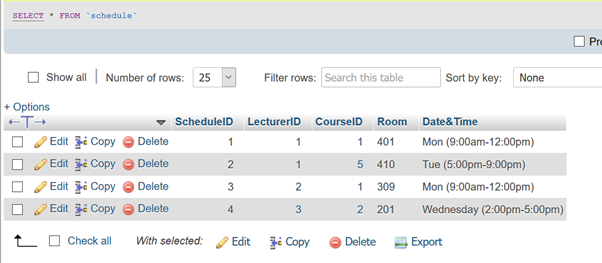
8. Postcode



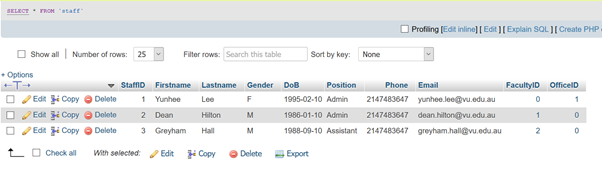
9. Roster



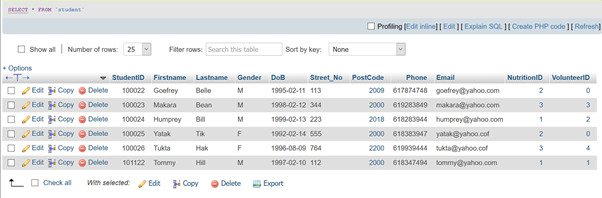
10. Schedule



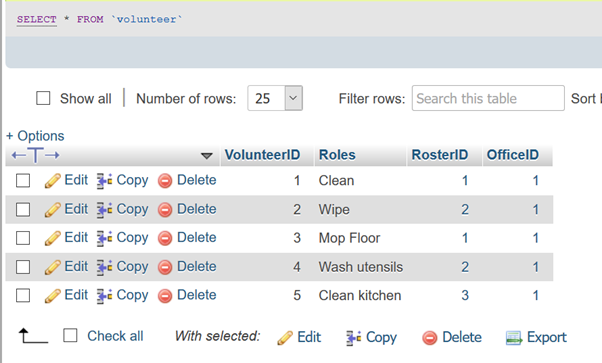
11. Staff



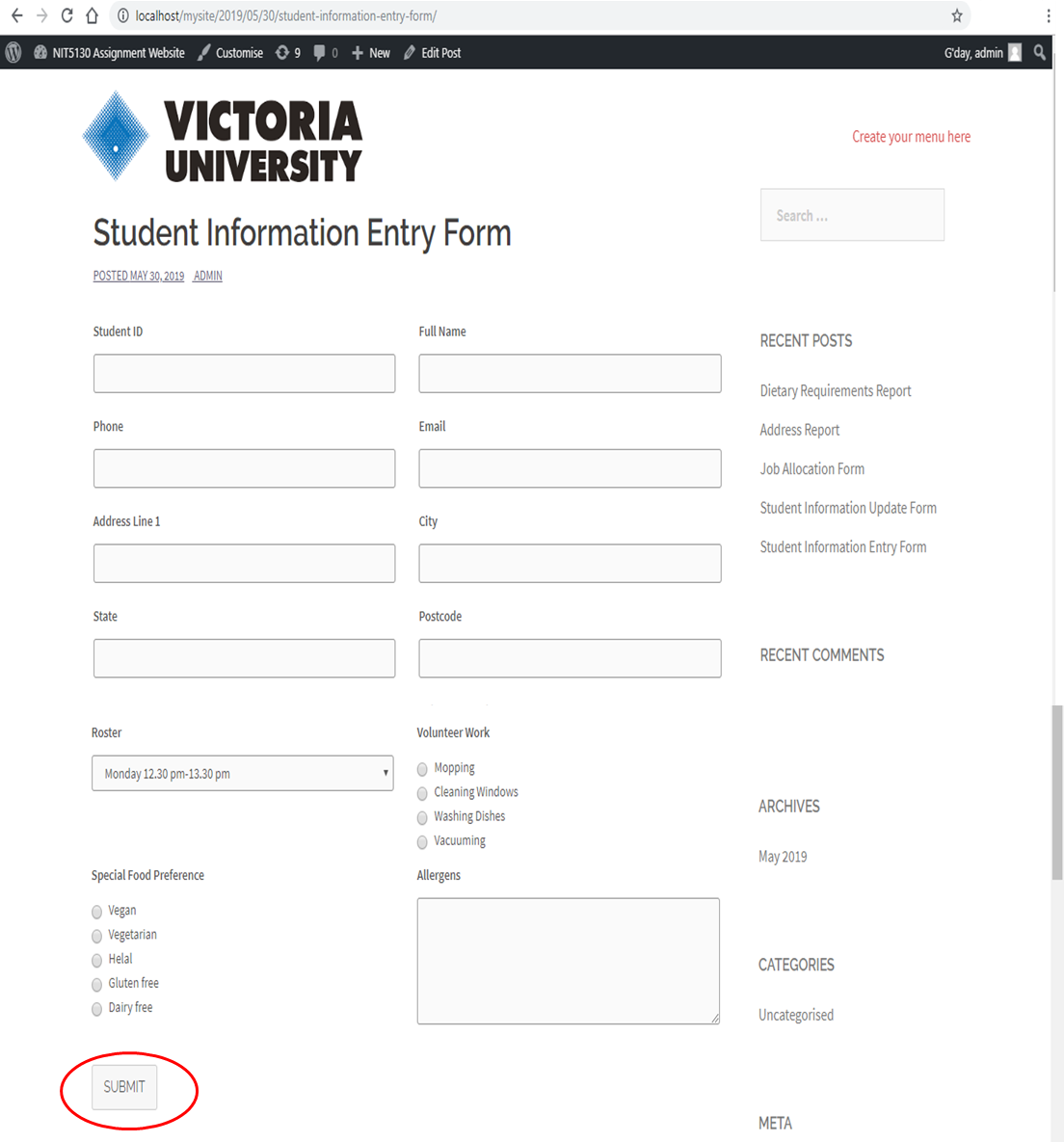
12. Student

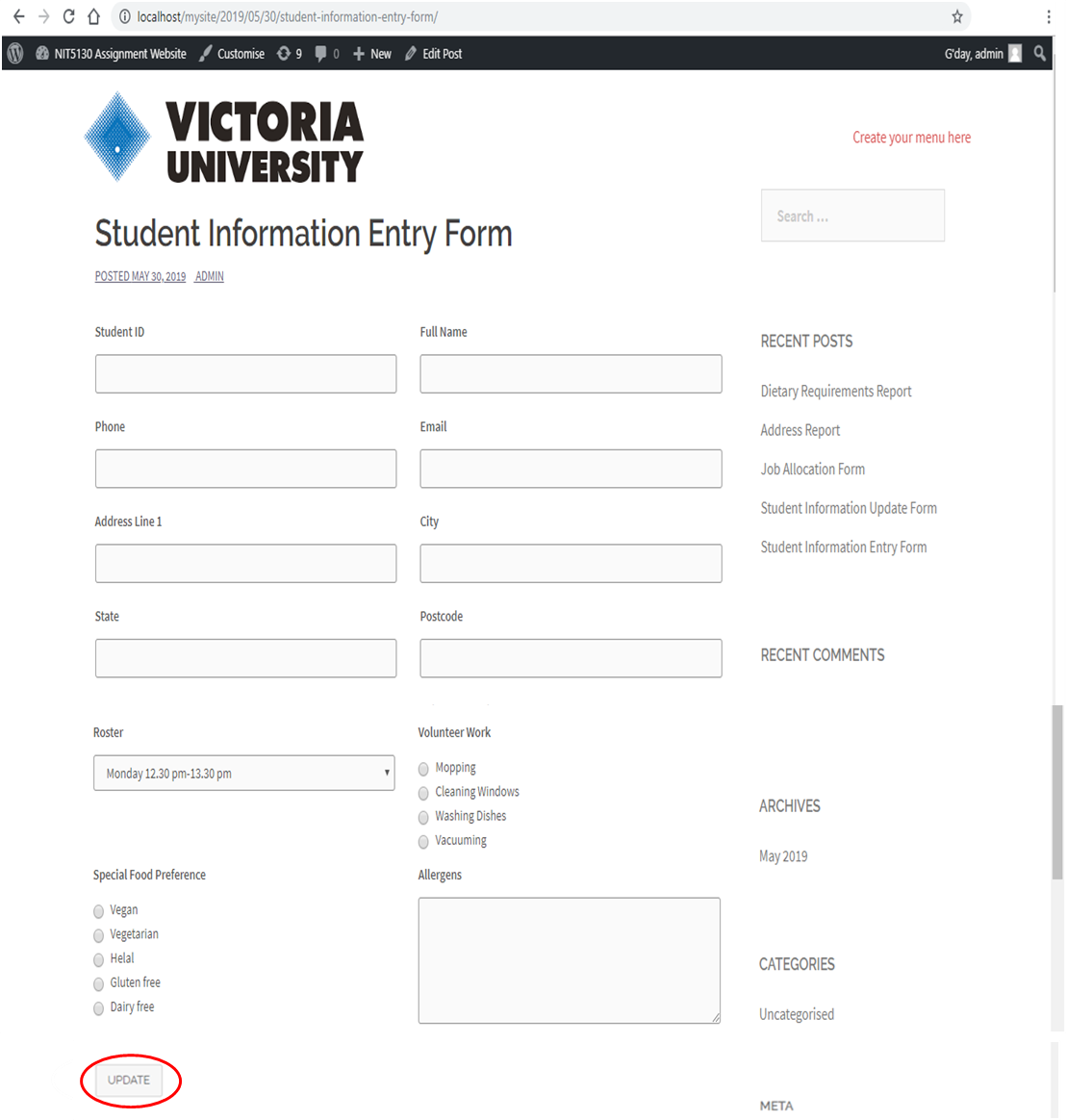


13. Volunteer

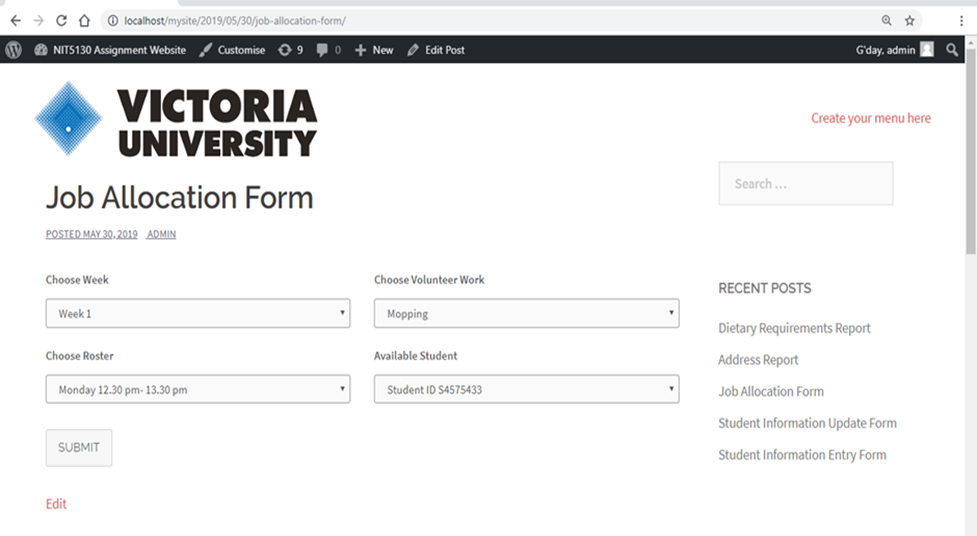


# 3.2. Forms and Reports

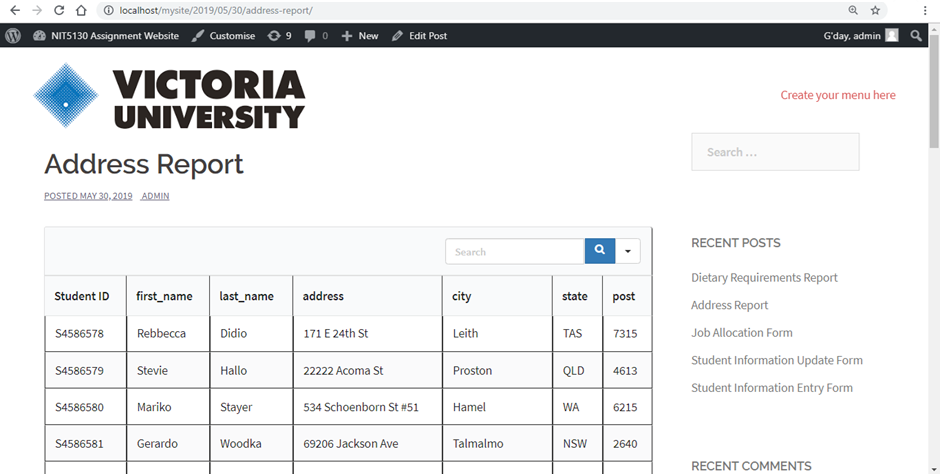
Student information entry form has been designed according to assignment requirements and necessary information for member registration. It is easy to fill and does not take much time to fill. Other student’s information such as course, lecture, semester have not been used for the forms; they have been used to create ERD diagram and ready to use for design of feature forms.

Student information update form is designed for directors to update students details and change volunteer work and roster information without losing data accuracy.

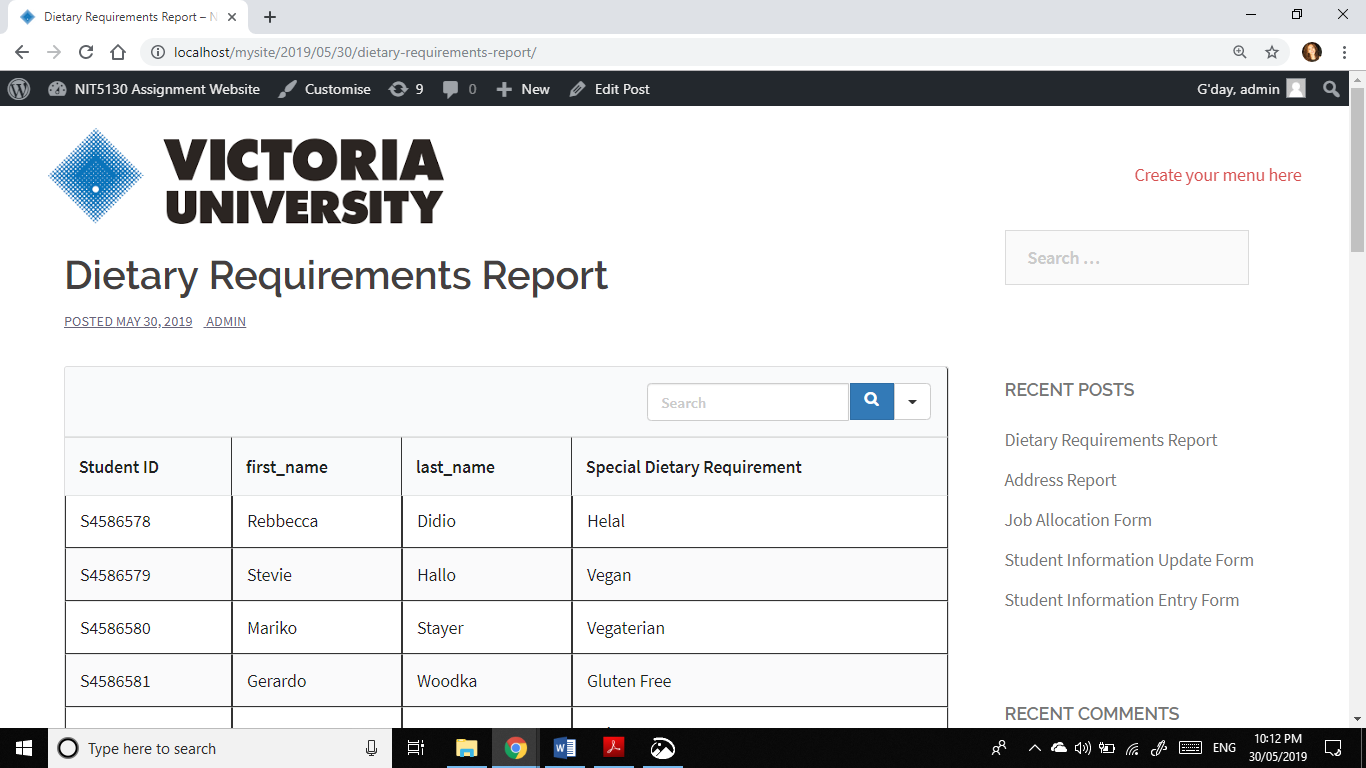
Job Allocation Form has been designed for the director to allocate the volunteer works for each week with students’ availability.



Address report has been created for directors to print the students’ address details.

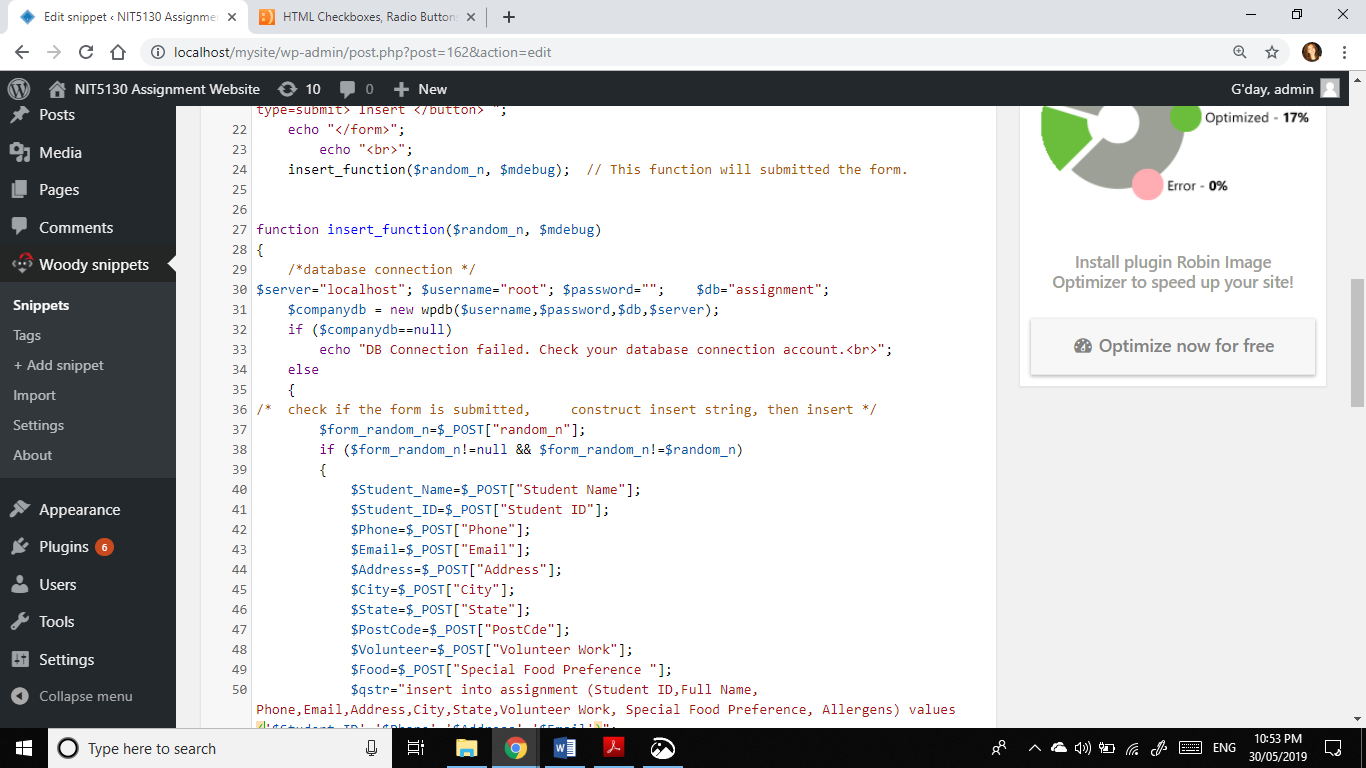
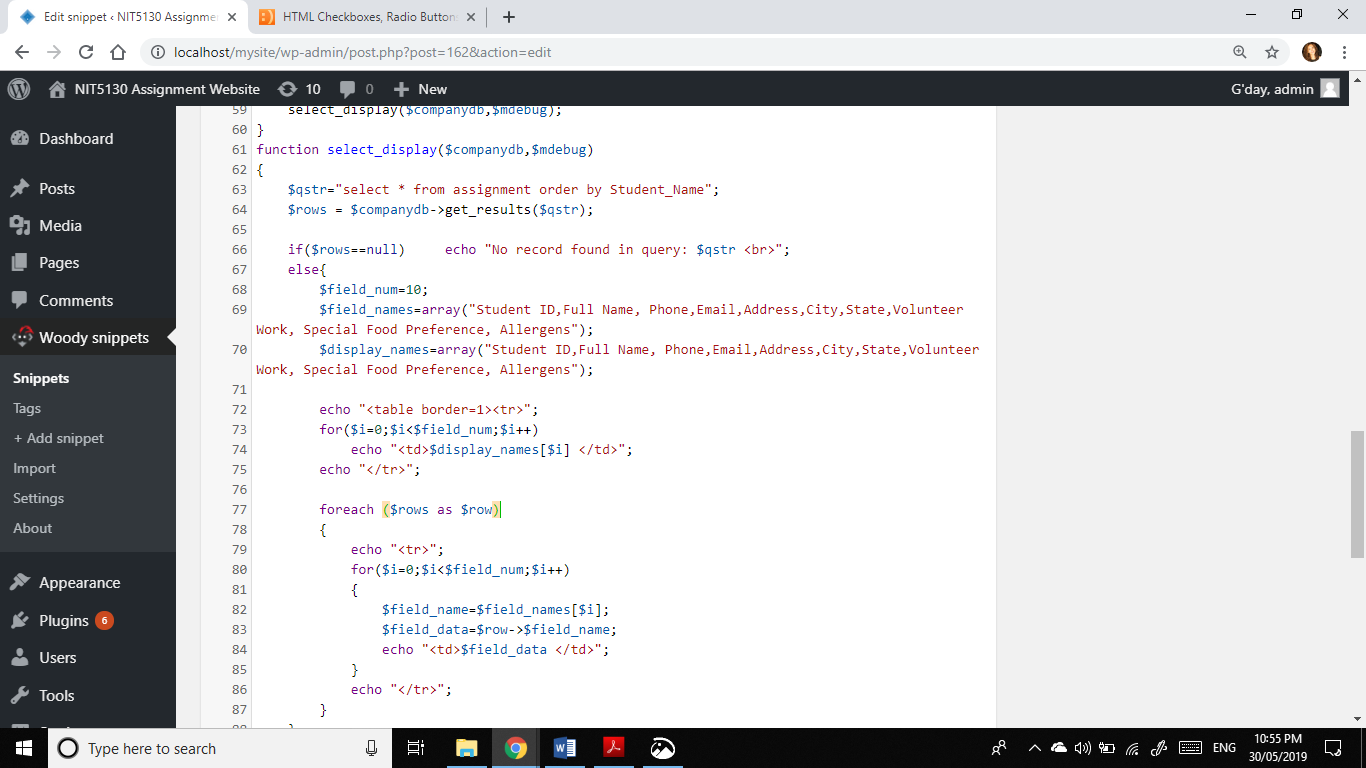


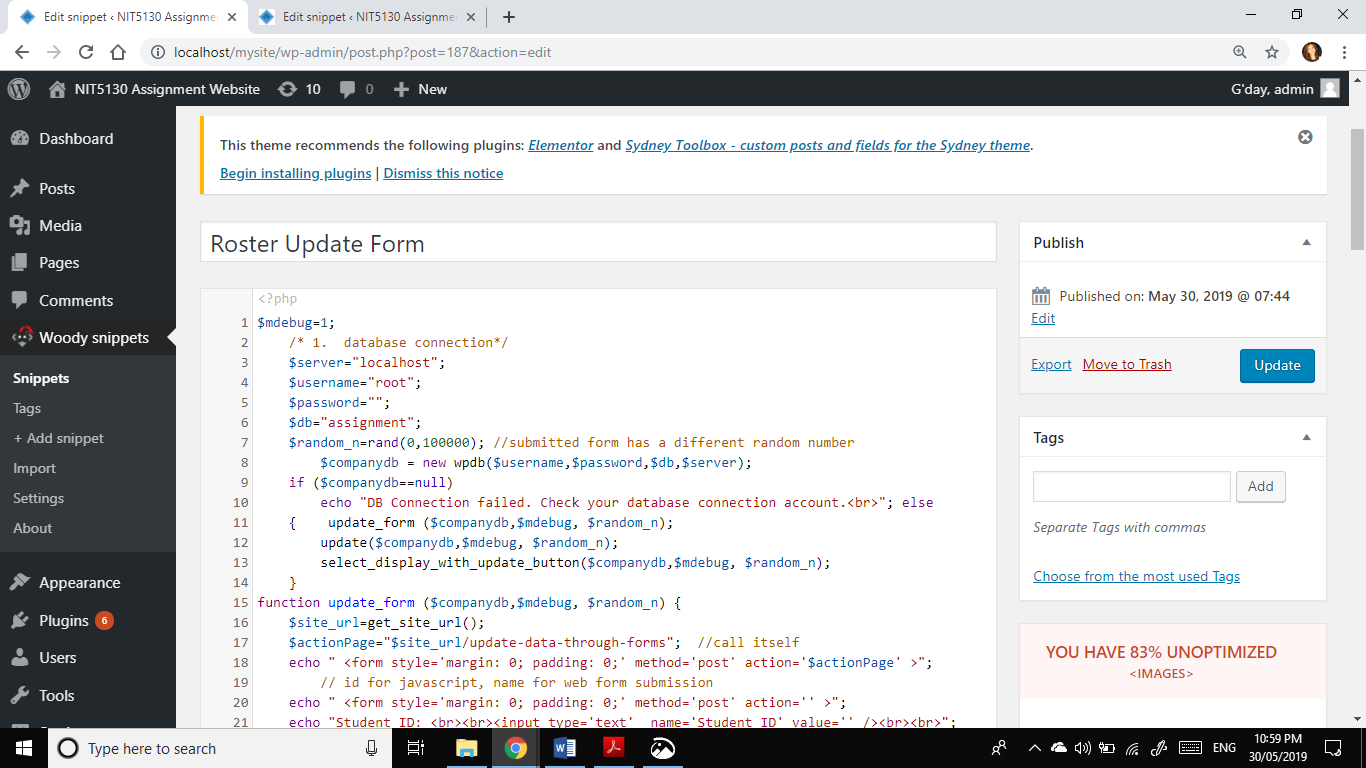
Dietary Requirements Report has been created to list all of the students with special dietary requirements.

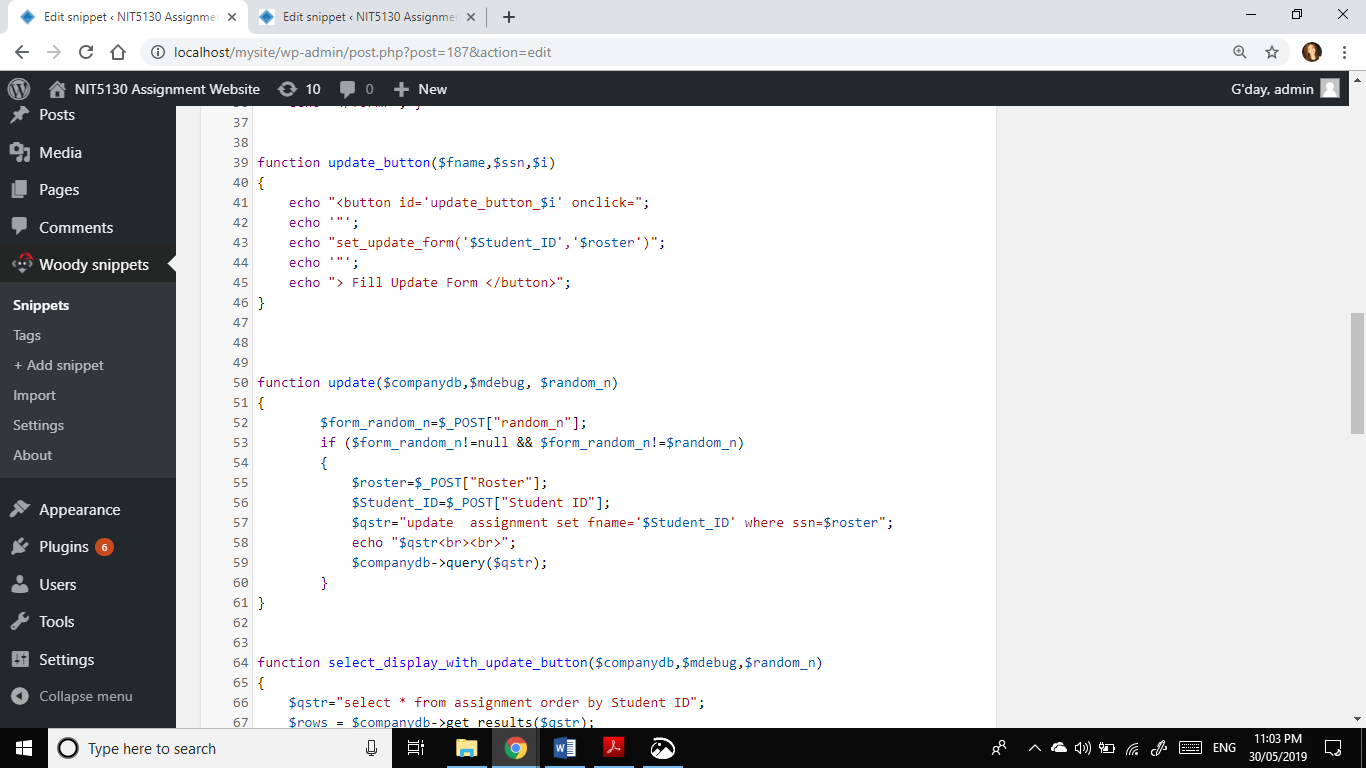


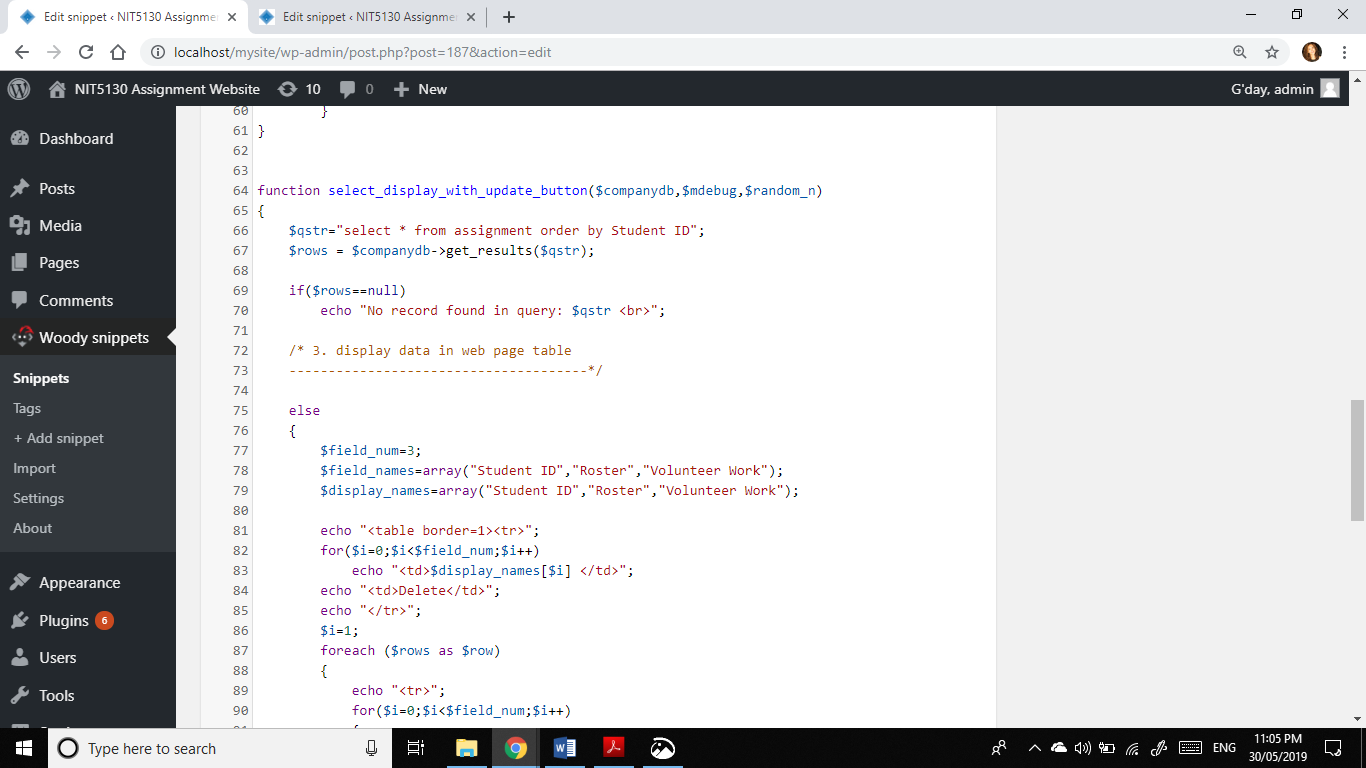
# 3.3. Codes

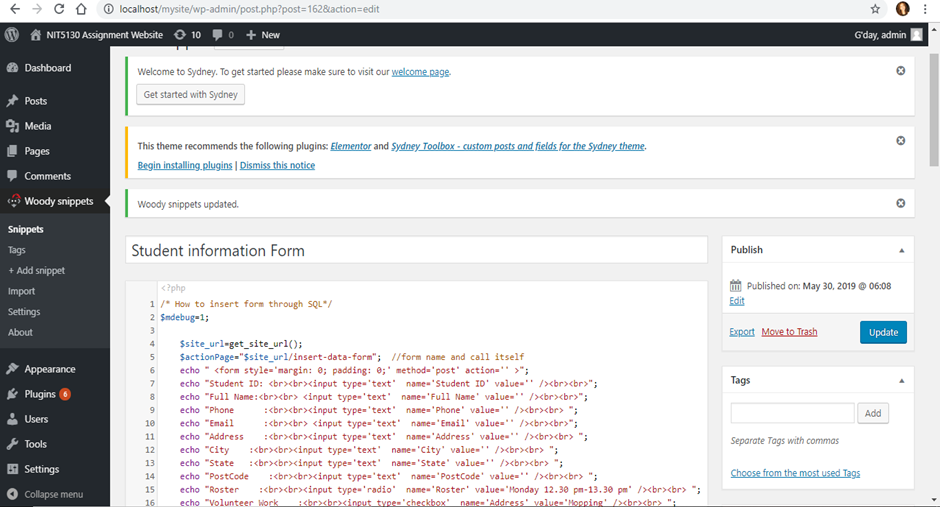
All codes for forms has been shared for feature benefits.











# 4. Appendix

Declaration Form

Subject Name: Database Analysis and Design

Subject Code: NIT5130

Campus: Sydney

Lecturer: Yousefi, Ashkan

Student Id \_s5698386\_\_\_ Name \_\_Jinxi Han\_

Student Id \_s4584593\_\_ Name Nguyen Truong Anh\_

Student Id \_s4580349\_\_ Name\_\_\_Makara Vann\_

Student Id \_\_s4584569\_\_ Name \_\_Alev AKTURK\_

Students’ Declaration

I hereby certify that I am an author of the submitted work bearing my name and student identification number.

Signature \_\_Han\_\_\_\_\_\_\_\_\_\_\_ Date \_\_8/6/2019\_\_\_\_\_\_\_\_

Signature \_\_\_\_\_\_\_ Anh \_\_\_\_\_ Date \_8/6/2019\_\_\_\_

Signature \_\_\_\_\_ Alev \_\_\_\_\_\_ Date \_\_\_\_8/6/2019\_\_\_\_\_\_

Signature \_\_\_\_makara\_\_\_\_\_\_ Date \_\_\_8/6/2019\_\_\_\_\_\_\_

# 5. References

phpMyAdmin, 2019. phpMyAdmin. [Online]

Available at: https://www.phpmyadmin.net/

[Accessed 30 May 2019].

WampServer, 2019. WampServer. [Online]

Available at: http://www.wampserver.com/en/

[Accessed 30 May 2019].

WordPress, 2019. WordPress. [Online]

Available at: https://wordpress.com/

[Accessed 30 May 2019].

TechTarget, 2019. TechTarget. [Online]

Available at : https://searchsqlserver.techtarget.com/definition/database-management-system

[Accessed 7 June 2019].