* **Introduction: An explanation of the problem and the objectives of the project.**

1. Introduction

Niger state polytechnic, Zungeru is a tertiary institution with 2 colleges (college of Science and Technology (CST) and College of Administrative and Business Studies (CABS) each college has 3 schools (Environmental studies (SES), Engineering Technology (SET), Administrative Studies (SAS), General Studies (SGS), Business Studies (SBS), and Natural and Applied Sciences (SNAS)) each school has various departments and about 35 programmes form 17 departments in total. Result computation starts from departmental level, the department collect the approved list of students and courses from the Academic Planning Unit, all exams scores are submitted to the departmental coordinators by various course lecturers, then the scores are computed against each student and the responding courses they offered. After computation, the result is submitted to the exams and records unit in broad sheet format ready to be presented to the academic board, the broad sheet comprises of analysis showing all courses offered, course code, unit, grades, total, mean, standard deviation and percentage pass, it also include the result summary, showing total number of students in class, number of students that passed, number of students with Carry Over, number of students absent with Excuse, and number of students advise to withdraw[[[1]](#footnote-1)].

In view of the above, the new (automated) system will comprise of five user access role as:

* System manager: This user controls all user accounts, manage personal records of students and courses (add, update and view)
* Department Coordinator: This user manage student scores (import/input/add student exam scores, consider/edit/update) score and allocate courses to a staff in a specific department
* Student: This user will only have access to view their individual result as a whole or on semester basis.
* Teaching staff: this user can import and view student exam scores of only course allocated to them by the departmental coordinator.
* Exams and Records: This user can only view student personal information, courses, and broad sheet result of all departments

All of the users listed above will have different view when given access to the system through a login page process.

1.1 Aim and Objectives

The aim of the project is to develop a web-based system as an automation to the existing manual process of computing students’ examination scores in Niger state Polytechnic, as the existing system is prone to errors and waste of time and resources like paper. The objectives are as follows:

* Gather information about user needs of a Result Computation System
* Design a wireframe / prototype
* Develop and design a web application in PHP, using MySQL, HTML, CSS and JavaScript, with five (5) user role each with different user view as describe above.
* To Keep track of up-to-date records of the entire students in polytechnic
* Support Registration, data upload, queries to the system, generate broad sheet report and visual and graphical result analysis for academic board decisions making with user-friendly interfaces for easy interaction
* Test the application for any dysfunctionality

1.2. Organisation of the Report

This reports start by introducing the online result computation system in section 1, a brief review of relevant literature and products to establish the context of the project in section 2. A specification of the problem and an explanation of how the student arrived at this specification. An initial work schedule including an overall project plan with time-scales, deliverables and resources in section 3. Next is the Design, include the design method, design process & outcome and Design decisions are described in section 4. Implementation, testing, final evaluation and description of the functionality and interfaces of the completed system comprising; the description of production, testing and debugging and demonstration that the specification has been satisfied using formal user evaluations and reports in section 5. Finally summary, conclusion and recommendation in section 6.

* **Background: A (usually brief) review of relevant literature and products to establish the context of the project.**

Nigerian tertiary institutions follows the trend in Information and Communication Technologies in transforming their record keeping and information management operations. RMS is currently an information systems that readily attracts the attention of many tertiary institutions in Nigeria.

The review of relevant and related web applications (result computation system) is an important process to achieve a successful design and implementation of an exclusive result computation system.

Base on the research it is obvious that majority of the product didn’t have the combination of student profile, graphical and visual result analysis, and broad sheet generation as a whole. However, some of the system have the combination of two out of these functionalities. Though, many only generate broad sheets, which is one of the major component of the system required by tertiary institutions. More also, only a few of them allow importation of scores from file e.g. excel (.csv) format. Finally, only few has more than two user access roles. Some related literatures are reviewed bellow.

**2.1 Review of relevant and related literatures**

Creating and organizing information in a useful manner is known as data processing. (Udeze, 2017) Explain how the manual method of computing students result prompt to lots of errors in most tertiary institutions in Nigeria, therefore, the use of computer systems to process information is imperative and desirable as it would enable computation of students’ progress, better access to students' records (personal, and courses), updating student records, keeping track of passed and failed courses (performance). The traditional method suffers lots of setbacks; it takes a lot of time and prone to errors, leading to late publishing of examination result, sometimes with inaccurate grades and GPAs generated from scores computed, these errors could lead to wrong decision in awarding students [[[2]](#footnote-2)]. (Akpasam, 2017) says that Most of the Students’ RMS are traditional and standalone with few or basic elements of a standard Result Management System (RMS). Most of the students RMS are not web based and others do not integrate well with other information system in their institutions. Therefore, the support for RMS with an automated platform to managing result for all categories of students in a friendly and interactive manner required [[[3]](#footnote-3)].

Ashwin (2017) describe student result analysis as an online application used to analyse, store and keep track of student data and compute the mark analysis process in an educational institution, it enable the view of individual student’s result, automated computation of results from various departments. The use of computation systems that has it capabilities beyond result computation is not so much in use. Tertiary institutions today compute student result analysis manually which consume lots of time and effort. System that does analysis of student result is required as it will aid decision-making and taking in tertiary institutions [[[4]](#footnote-4)].

Eludire (2011) observed that a number of problems associated with student academic record management include improper course registration, late release of student results, inaccuracy due to manual and tedious calculation and retrieval difficulties/inefficiency. According to him, the development of database concept is the answer to these problems, where the amount of redundant data is reduced and the possibility that data contained on a file might be inaccurate because they were never updated[[[5]](#footnote-5)]. Amar, (2009) indicated that publication of students’ results in the manual system takes a very long time resulting in the students remaining idle for long. Sometimes the delay in declaration of result causes heavy losses to the students as generally they cannot join further studies, or appear in competitive exams and jobs because of the unavailability of certificates on time [[[6]](#footnote-6)].

According to E. O. Ukem et al (2012), result systems can be improved upon by using back-up policies and audit trails. Although his solution was built with java and MySql database, it did not establish any evidence for processing results batched in files to increase data integrity and mitigate fraud. The research could not also address what happens to a student after s/he is on suspension, deferment of studies, medical treatment, supplementary exams or related cases. We have analysed these scenarios and developed an audit trail for references [[[7]](#footnote-7)].

Just like the solution by Akinmosin (2014) in his study titled “The Architectural Design of an Integrated Virtual Classroom System” the software has a login form for authentication of users and Student Registration forms for registering students every semester after payment of dues, before results are uploaded using internet browsers. His solution was built with the Oracle Procedural Language/ Structured Query Language (PL/SQL) and adopts forms and reports for its implementation. However, the interface widely uses “Grades form for inputting student grade and Grades Edit form for editing erroneously entered grades.” This is a security threat and a loophole in the design that could aid intruders in accessing the database. It’s a weakness that may not properly check the mutilation of grades by users who find a way to access those forms [[[8]](#footnote-8)].

Bhatt et al (2016) proposed a Credit-based Grading Scheme (CBGS) in India. It is simply a PHP-MYSQL solution that can compile results of students. The reports of the system are generated in either excel or PDF format depending on the usage. Although the grading scheme differed from that of the Nigerian system, the formula for computing the Grade Point Average (GPA) is similar: the summation of the product of the credit hours and grade points divided by the summation of the credit hours. One major advantage of such automated systems over the manual procedure is the ease of searching and list generation [[[9]](#footnote-9)].

In view of the above related work it is clear that the requirement to have an online Result computation system in Niger state Polytechnic, Zungeru, Nigeria is important. In the institution, Students result are computed every semester from the summation of CA and Exam score of each course taken by students, given each student grades, GP, CGPA and remark. This is presented as broadsheet with analysis. The computation of result is completely manual.

This project is to create a web-based Result Computation System that would allow teachers submit/input/import students examination scores, the system compute the scores, generate a broadsheet and provide a graphical/visual analysis of the results on a semester basis. The students would be able to view their result through their profiles. Hence, in this project, we considered issues of security breaches as utmost priority and have incorporated a log file to help monitor activities that go on in the database. This will make the system more credible and enable the management account for changes made at any time by a particular user. It also keep backups of record in case of loss.

* **Requirements specification: A specification of the problem and an explanation of how the student arrived at this specification. An initial work schedule including an overall project plan with time-scales, deliverables and resources.**

**3.0 Requirement Specification**

To commence any development, it is important for any project to consider the requirements of the end users. This is to ensure that the proposed system function the way users expect it to. Therefore, requirement specifications have been gathered from sources as will be described later in this chapter. The primary focus of system requirement analysis and specification is to understand in detail what the user(s) really want the system to do. For us to know what the user needs, requirement analysis was conducted with the sole aim of gathering desirable features for the system.

This phase of the project gives an explanation of how we arrived at the specifications, including problem specification, methods of gathering requirements, online review, wireframe & prototyping, initial evaluation of the prototype, implementation, and project plan.

**3.1 A specification of the problem**

With the current trend in technology, Niger state polytechnic as a tertiary institution should have an automated means of computing student results, to enable it meet certain educational standard and requirements, an automation is required to eliminates errors in computation and reduce miss-use of resources and bring ease.

Currently the school is operating a manual system, every department computes students result in different formats in an un-centralised manner, some departments uses format such as excel sheet while some are completely paper based.

In previous years, a partial automated system of result computation has been introduced by the school authority to various departments that was used as a standalone system to enable departmental coordinators enter exam scores in text file then use the command line environment to manipulate the text files to generate a computed result, the system was not friendly, requires memorising commands and it takes a lot of time to arrange the scores in a text file and does not support editing, it is developed using Fortran. Yet with the introduction of the so called automated system every operation still seems manual, errors recorded are even more compared to the use of excel files, which has led to many departmental coordinators to dump it. Even after result is generated still analysis is done manually, given inaccurate results. It take the coordinator so many days to compute a single class, they become stressed up and sometimes frustrated as so much time is required. This problem lead to delay in producing result for student and also delay in taking and making decision.

If this problems are not taking care of it would lead to an educational degradation and inconsistent result generated continuously, although many institutions have similar issues or problems but they have often used different methods to solve the problems, but I feel developing a web based system that will be used by all department in a centralised manner, enabling result computation, adding and editing student exams score, generating broad sheet result with analysis, allowing individual student see their result on time from the comfort of their homes and also enable timely decision taking and making will make more significant difference from the methods previously used.

Finally I decided to develop the proposed system based on the problem statements stated above, leading to the requirement stated below.

**3.2. Gathering Requirements**

In order to gather the requirements for the web application a heuristic evaluation was conducted on a similar web application.

The Involvement of the researcher in the process of computing student results makes it easy to investigate the existing system and to obtain detailed about the application area to be designed. During our findings, several effective methods of information gathering, or data collection were employed which include: evaluation and inspection of relevant documents such as result broad sheet, raw score samples, and transcript formats. The data required for the development of the proposed system were from primary and secondary sources, using a survey/questionnaire and searching from the internet respectively.

With the detail idea on how the existing system work, most of the data/information required to build the proposed system is known and acquired through primary and secondary sources.

The data from primary source is the outcome of the prototype evaluation done by the participants and the secondary source include data about NBTE grading system available on internet for public view, result broad sheet and result analysis format all of which could be found online [See samples in appendix](file:///C:\wamp64\www\Compute\MSc%20Work\MSC_Project\appendixa). The data gathered is use to fulfil certain requirement in the development of the proposed system, most of the information from participants would be used for the purpose of:

1. Motivation (do the target audience want the proposed system),
2. Usability (Can the target audience work out how to use it?) and
3. Content (Does the target audience understand what the proposed system is about).

**3.3. Online Review**

An online review was done to understand how other similar systems work, to find out what is lacking and how the proposed system can be a better replacement of the existing system.

From the review an idea for creating a use and misuse case diagram representing the usage requirement and possible threats or attack to a system, this was helpful as it describes the flesh of the actual requirements. The focus of the research was mainly on the computation of scores to produce results with analysis and preventing possible threats. As an insider it was easy to get the major information’s required for the use and misuse case.

Another method used is an alternative to use case diagram known as Persona. Alan Cooper introduced Persona in 1998. Personas represent potential users of a particular product as a fictitious characters. It is very beneficial in the design process, it help the designers to better focus and understand the user experience as they use the product [[[10]](#footnote-10)].

The use persona in the study was because of its benefit and it’s more realistic nature; for instance, personas can describe multiple type of users, while actors are limited to a single user type. Equally we also uses the Use case and misuse case diagram to clearly understand and fetch out the possible attack on the functionalities of the system and preventive measures to the attacks [[[11]](#footnote-11)].

Use and misuse case diagram and persona will be detailed discussed in the chapter below.

**3.2.3 Prototyping**

Having known the requirements, wireframes were designed to lay out the functionalities and specific content on a page, it takes into account the needs of users and user journeys. It was an early approach in the development process to establish the basic structure of a page before adding the content and visual design.

Indigo Studio was used as a prototyping tool to develop a prototype that mimic the proposed system, the prototype was used for pre evaluation to collect information from users about the proposed system in other to better understand the required functionalities and better user experience.

Indigo Studio is an interaction design tool from Infragistics[[12]](#footnote-12) that enables you to explore and create functional, animated User Interface (UI) prototypes. The tool is aimed at designers who lack development skills and developers without or with few design resources, it’s an interaction design tool that lets you prototype user interfaces without writing code.

Indigo Studio makes it easy for anyone to rapidly design functional, animated UI prototypes that maximize usability and appeal. It promotes design process best practices and enables teams to stay focused on the end users and their stories.

**3.2.4 Initial Evaluation**

The design of the prototype was based on information from an observation and direct involvement of the researcher/developer as he is part of the examination committee and an instructor in the polytechnic. It is needful to have an extra opinion as it was helpful and significance in the development process.

To answer these questions; (will the target audience find satisfaction in the use of the proposed system, can they use it, and do they find ease in using it) leads to the design of the prototype for evaluation using survey question to be completed by participants recruited. The participant will fill the evaluation form responding to questions base on their experience and observation. Typically 10 participant were individually given time to interact with the prototype so as to express their opinions, thoughts and preferences about the proposed system with no negative influence through responding to the survey question in the evaluation form. Example of the evaluation question can be found in the [appendices](Apendix)

Focus group is an alternative method that would require less time to enable approximately five participant respond to the evaluation questions and the result could be negatively inflicted as few individuals could possibly be introverts while others take control of the debate and impact the end result, or possibly even introduce bias. The researcher may also have trouble controlling participant’s discussions and managing process.

The responds to the evaluation questions were analysed starting with the end in mind (check of the top research questions) using the **cross tabulating and theme & explore** approach to filter results, Interrogate the data, analyse the results then draw conclusions.

Theme and explore is the process of selecting key quotations from each respondent illustrating the themes found. It was more like a literary analysis. Theme and explore is good for smaller sample sizes and more complex subjects. It is particularly helpful when your respondents have different understandings of the same issue and you want to compare them. To ensure a credible and useful evaluation result, the data were systematically analysed.

**3.2.5 Implementation**

The two most popular methodologies for web development (Agile method and Waterfall) were studied to choose the most appropriate for the design and implementation of the proposed system.

Waterfall basically is a sequential model where software development is segregated into a sequence of predefined phases including feasibility, planning, design, build, test, production, and support. On the other hand, agile development methodology follows a linear sequential approach while providing flexibility for changing project requirements, as they occur [[[13]](#footnote-13)].

Agile principles arose in the 1990s, in part, to address perceived weaknesses of the traditional waterfall development process. Agile methodology has steadily gained credibility and adherence over the last two decades [[[14]](#footnote-14)].

In 2001, Agile codified several lightweight theories under one umbrella, which revolve around short, iterative changes and frequent builds in order to make the development process responsive and adaptive to changing requirements. Agile developers believe that making the customer/business happy is more important than executing a long range plan.

In view of the above differences between the two methodologies, we choose to implement the proposed system based on the agile method to fully leverage it advantages.

**3.3 System Requirement Analysis and Specification**

**3.3.1 Departmental Registration:**

New Student

A student whose name appears clearly in the approved list of admitted students dispatched from exams and records unit to the department through the polytechnic authority should be allowed to proceed to the department for screening.

Upon a successful screening in the department, the student is permitted to register in the system by uploading his/her academic credentials and other important documents to the school registration portal, after obtaining matriculation number.

Each department would export the records of successful registered students of its department in excel format from the registration portal, then upload or import it into the result computation system by the department examination officer or coordinator, as the systems are design independent of one another.

All student would be allowed to access their accounts by login in using their matric number and a default password, a student would be required to change their password in the first login before having full access to their profile menus.

Course Registration

In the polytechnic, student of the same department and the same year offer the same courses, the department register causes against all student.

A student should be allowed to view the course co-lecturer for every course registered per semester from the course allocation list uploaded to the system.

Extra-year student

All the courses that is failed by this student would be a resit and the scores of the courses would be treated as a special case or carry over cases.

The system should automatically select all the courses failed in past session if any.

**3.3.2 Problematic student**

Exam Malpractice and Related Offenses

Student who is currently facing the penalty of suspension/rustication should not be allowed to access any feature of the system until he/she is restored.

A student whose session has been nullified or rusticated for a session should automatically join a like class, courses before rustication on return after rustication.

Health Challenges

Supplementary examination can be granted for the lost semester(s); which could be uploaded as replacement.

**3.3.3 System Requirements Specification**

The system should accept scores uploaded by the lecturers in excel (.csv) format in the process of data input; map the results to individual records in the database; Compute the Grades, Grade Points, GPA/CGPA of each student; and generate Students Result in a broad Sheet format then prepare analysis of the result.

Functional requirements based on type users include; Students, teacher/course lecturer, Department Examination officer/Coordinator, Examination and records unit, and Academic planning Unit (System manager).

**3.3.4 Functional Requirements of Registration**

At the beginning of a new session, prior to the addition of new students to the database, student on the new admission list would undergo screaming in their various department and the successful once would be given matriculation number. In most cases, not all admitted students are cleared for studies; hence, a list of cleared students with matriculation numbers is uploaded into the system. All students uploaded are assigned password default password, and an account is created for each of them. It is at this point that they can change their password at initial login and view their profiles and view their result at the end of the semester when result is uploaded.

**3.3.5 Major Requirement**

* Keep track of each student records.
* The system considers special cases like suspension, deferment of admission, expulsion, supplementary exams and leave on health grounds.
* The system can automatically fetch all the REPEAT courses of the previous session, for student with extra year or carry overs.
* Restrict any change or modification of some sensitive information such as student registration/matriculation number, name, and level, year of admission and mode of entry.
* Authenticate all documents uploaded for the student using document clearance number system before submitting to the database.
* The system would enable system manager to upload staff list, student list and course list, manage the list of staff and students and keep the course list up to date, grant access to records such as, student record, staff record and courses, also to manage backups and logs
* The system would enable coordinator to upload/import, edit student score, can view broad sheet result and analysis and also change his/her password
* Student can view their result through their profiles and make changes to their password
* The system allow Exams and records unit to have access to the result broad sheet and visual/graphical analysis of the result through their profile and can also make changes to their password,

**3.3.6 Treatment of Special Students' Cases**

* **Offenses/Malpractice**: The system considers some offenses which may result in rustication, expulsion, suspension or nullification of some academic work.
* **Deferment of Studies:** The system provides facilities for a student who wishes to defer his studies.
* **Health Cases**: A student with approval of a medical report has such a case created on his account indicating suspension of studies until further notice.
* **Supplementary Examination:** Approval of supplementary examination is done by the HOD into the system after the Registrar must have granted it.
* **Reinstatement of Students**: On all these cases, the student is expected to report back to the department after which he is reactivated back into the system.

**3.4 Project plan**

The time management plan used in this project work is describe as shown a table with two columns, for tasks (activities) and time (duration) of each task. Some of the tasks that happen at the same time is noted next to the respective tasks.

|  |  |
| --- | --- |
| Activities | Time (Duration) |
| Defining the goals and collecting necessary information about the requirements | 2 weeks |
| Prepare records and apply for ethical approval | 3 weeks |
| Create wireframe and prototype for the web application | 1 week |
| Wait for the approval from ethics committee | 2 weeks |
| Planning the evaluation process   * Meet with participant (consent forms) * Ask them to respond to the evaluation questions, and observe them interact with the prototype | 2 weeks |
| Proposed database structure   * Draw an ER diagram * Set up a localhost server (Wamp server) * Use the ER diagram to define the database schema on the server | 1 week |
| Adopt a template (User interface design)   * Setting up GitHub * Refine the template to suit the proposed system | 1 week |
| Create personas  Create use case diagram | 1 week |
| Implementation using agile method   * Create the product backlog * Create first sprint backlog | 1 week |
| Start development (using PHP, HTML, CSS and JS)   * Test code * Create second sprint backlog | 1 week |
| Continue development and testing code   * Create 3rd sprint backlog | 1 week |
| Continue development and testing code   * Create 4th sprint backlog | 1 week |
| Final Sprint review and retrospective | 1 week |
| Final evaluation of the developed system | 1 week |
| Writing report | 2 weeks |

Figure 3. Project plan

* **Design: This should include the design method, design process and outcome. Design decisions and trade-offs should be described e.g. when selecting algorithms, data structures and implementation environments or when designing for usability.**

**4.0 Design methods**

We conducted research on similar online result computation system, where tertiary institutions manage their student result as mentioned in chapter 2, similar features of the systems that would be beneficial in the development of the proposed system are explore, during the initial designs, persona, use and misuse case diagram were created, and also prototypes, were created and were evaluated with participants.

**4**.1 **Personas**

Cooper and Reimann defined a persona as the collection of genuine representative information which may include fictitious details for a more accurate characterization[[15]](#footnote-15). “Persona is a user-cantered design method which sets up fictitious characters to represent the different user types within a targeted demographic group that might use a site or product”[[16]](#footnote-16).

From the definitions above two important facts are deduced as; first, personas are created based on reality. It accurately represent users in the real world. Second, most of the attributes that make up a persona are imaginary. For instance; the name, social element and photo on persona could be totally fictitious.

Particularly, personal helps the developer in user interface design and to better understand the requirements of the end user, setting up a strong figure for better, reliable and consistent understanding of end user groups. The following are personas created based on five different users of the proposed system to enable users have best interaction with the system.

**Personas – 1**

Fictional Name: Shehu Hassan Kano

Age: 21

Status: Single

Job Title: Student

Goals:

* View current and previous semester results from a profile with personal access.
* As a returning student should be able access all results anywhere anytime

Technology: Uses his phone for Facebook and Twitter, because he is an active member on many groups for vegans.

Background: Shehu is a student of School of natural and applied sciences, department of computer science in Zungeru. He is a good student performing averagely as he is always punctual and take his studies with all seriousness. He spends a lot of time studying for his HND award. But, it is difficult for him to access his results at the end of every semester when result is released. However, he access his result on the notice board or from the department examination office.

Motivations:

* Simple design and easy to use interfaces
* View results with ease and comfort any time any and any where
* Account/profile accessible by him alone

Frustrations:

* Having to leave his comfort to travel in order to check his result
* He wants to use web applications that have make all result private

**Persona – 2**

Fictional Name: Ali Nuhu

Age: 31

Status: Single

Job Title: Examination Officer (Coordinator)

Goals:

* Input, import exam scores
* View current and previous semester broad sheet results from a profile with personal access.
* Allocate and unallocated courses to staff

Technology: Uses his laptop for studying and teaching students.

Background: Ali is a teaching staff of School of natural and applied sciences, department of computer science in Zungeru. He is up and doing as a teacher and as the examination officer of the department. He spends a lot of time studying to teach. But, he finds it difficult to import / input and analyse student result as he has to it done manually at the end of every semester when student exam score are generated.

Motivations:

* Simple design and easy to use
* Input and import students score with ease for computation at his comfort any time any and any where
* Generate result analysis
* Personalise and accessible accounts

Frustrations:

* Using a paper base computation of result system, analysing student result with calculator.
* He wants to use web applications that is not stand alone.

**Persona – 3**

Fictional Name: Aminu Saira

Age: 38

Status: Single

Job Title: Deputy Registrar (Exams and records)

Goals:

* View broad sheet result and analysis
* View current and previous semester broad sheet results from a profile with private access.

Technology: Uses his laptop for composing letter for student who need a backup letter for transcript and evidence to have studied in the institution.

Background: Aminu is a non-teaching staff of the polytechnic, a deputy registrar in exams and record unit of Zungeru. He is efficient in his duties and he prepare the result of the grandaunts after confirming the eligibility of grandaunt. He present broad sheet result to the Academic board for consideration.

Motivations:

* Simple design and easy to use web page
* View broad sheet result at the comfort of his home and office any time any and any where
* Personalise and accessible accounts

Frustrations:

* Having to store pile up papers of the previous results.
* Searching for previous result and student records

**Persona – 4**

Fictional Name: John Adams

Age: 37

Status: Single

Job Title: Academic Planning officer (System Manager)

Goals:

* Add, view and edit student records
* Add, view and edit courses
* Add, view and edit programmes, department, schools and colleges

Technology: Has good working knowledge of computer, uses smart phone,

Background: Adams is an academic staff of the polytechnic, serving as the director academic planning, plan and manage academic activities like student admission, examination time table, and lecture time table, he manages all the courses student offer.

Motivations:

* Simple design and easy to use Interfaces
* Manage student records, courses, and programmes
* Have private access to account and control other user access

Frustrations:

* Add and Retrieve student data both current and previous within the comfort of his office.

**Persona – 5**

Fictional Name: Sani Kuta

Age: 35

Status: Single

Job Title: Lecturer (Teaching Staff)

Goals:

* Add student Score and view added scores
* View course allocation

Technology: Has average working knowledge of computer, uses smart phone,

Background: Kuta is an academic staff of the polytechnic, serving as course lecturer in the department of Business Administration, he has two courses allocated to him, meaning he teaches two courses to different classes in the department, set exam for the students and mark the exams.

Motivations:

* Simple design and easy to use web page
* Manage student records, courses, and programmes
* Have private access to account

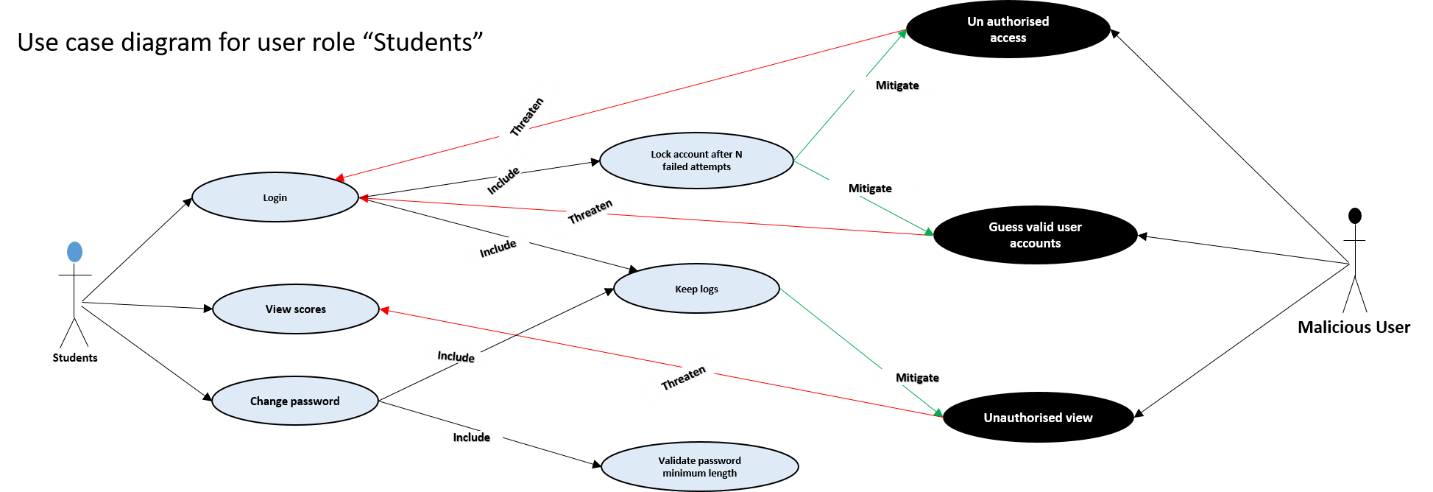
Frustrations:

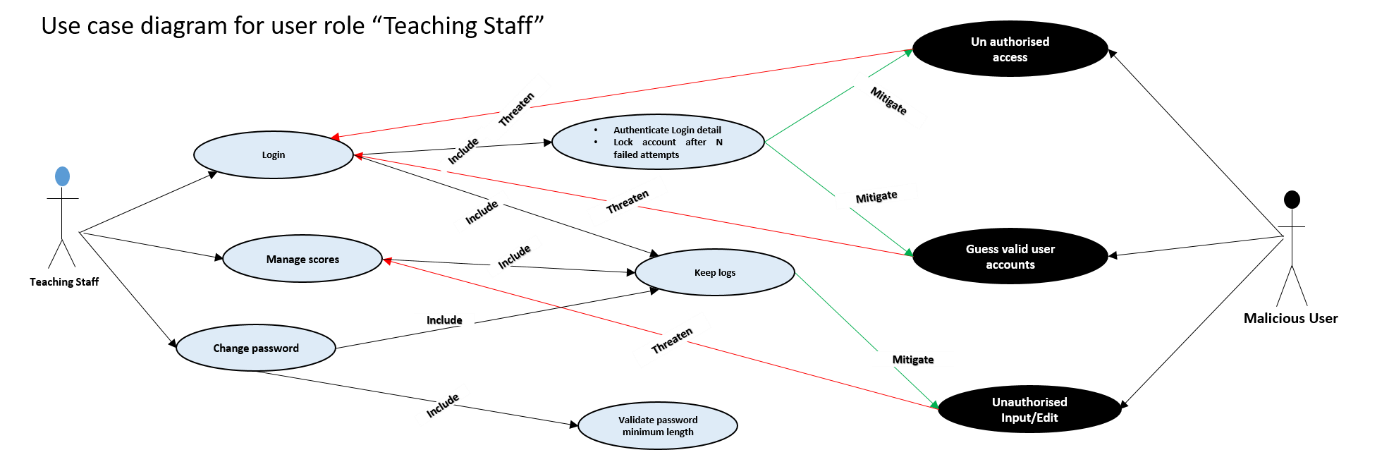
Add and Retrieve student data both current and previous within the comfort of his office.

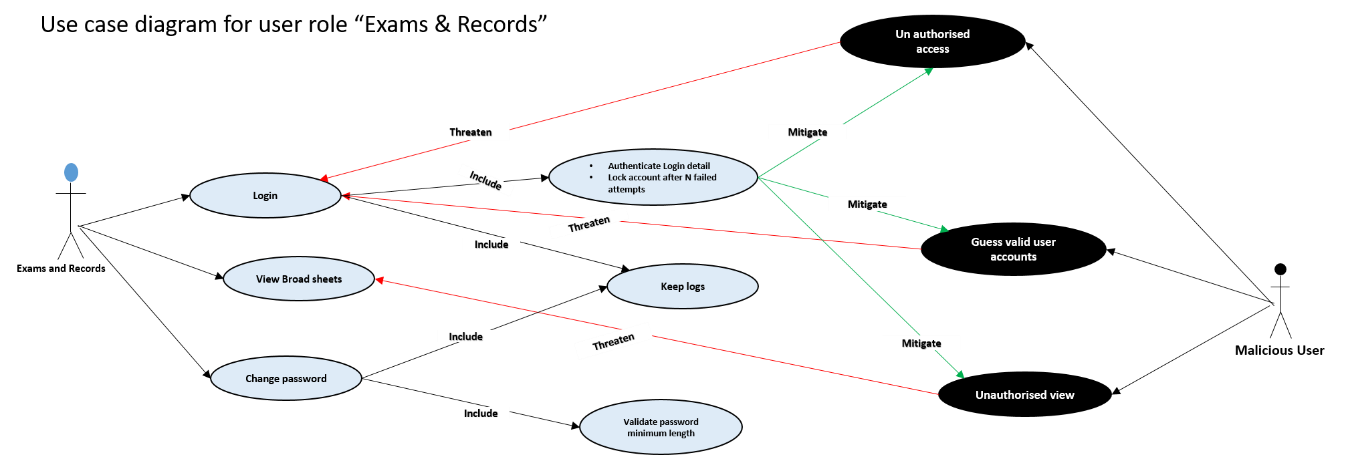
**4.2 Use and misuse case diagram**

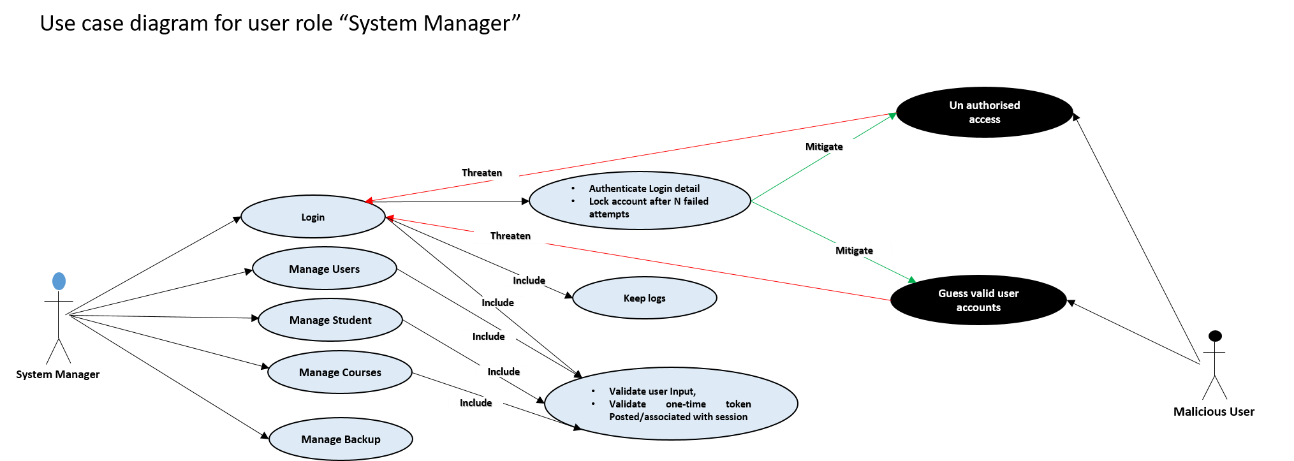
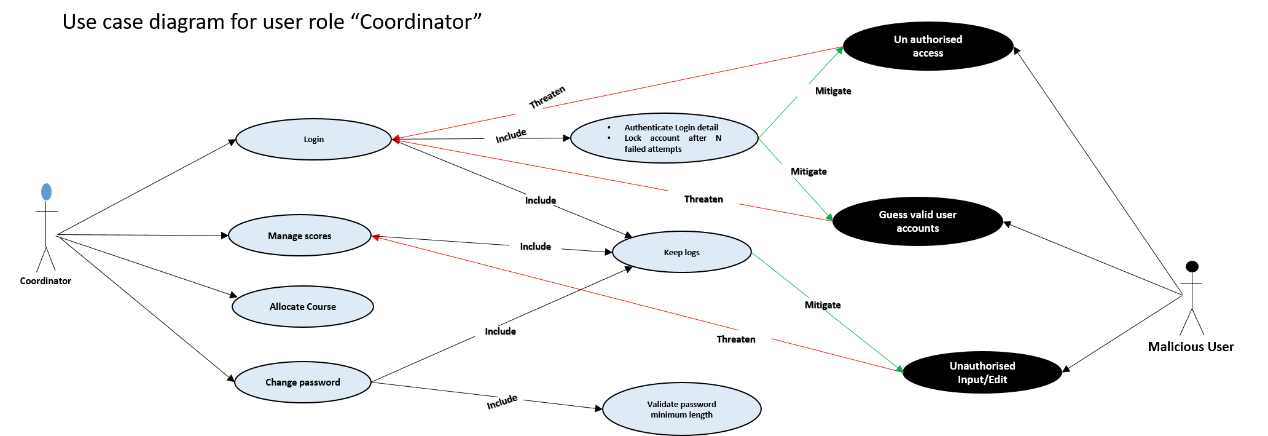
Use case diagram is a behaviour diagram in UML[[17]](#footnote-17). It is used in modelling system[[18]](#footnote-18) functionalities using use-cases[[19]](#footnote-19) and actors[[20]](#footnote-20) or used to gather the functional requirements of a system, to get an outside view of a system, identify the external and internal factors influencing the system and to show the interaction among the requirements are actors[[[21]](#footnote-21)].

Misuse cases[[22]](#footnote-22) are included in the diagram to show possible threat or abuse that may occur with the system. Misuse case is used to describe the process of executing a malicious act against the system. [Figure](usecase%20figure). shows the use and miss use case diagram of the proposed system. [See appendix](apendix) for the full documentation of the use and miss use case diagram of the proposed system.









**4.3 Prototyping**

Prototyping is described as initial phase of a software release, in which product amendment, and evolutional development may occur before the real product is release. This is sometimes known as beta phase or beta testing, it is to enable the project get evaluated by some class of users before full development [[[23]](#footnote-23)].

Prototyping is significant for the development of the proposed system to enable us take and make early decisions regarding the usability and functionalities users need. It was achieved using low and high fidelity prototyping.

**4.3.1 Low Fidelity – Sketches**

In order to get a tangible representation of the concept, sketches were drawn initially to get a feel for the design and layout; describing the proposed look of the proposed system to improve the existing system. It is characterised with low technology implementation, with materials like sheet of papers, ruler and pen. Figure ….. are the initial sketches of the proposed system, it describe the basic layout and content of proposed system’s login page, all users Home page with accessible menus, broadsheet result, student individual result, analysis page.

*Include the diagrams*

4.3.2 High Fidelity – Indigo studio

To create a higher tech visual representation of the website we uses a high fidelity prototyping tool called indigo studio, this prototyping software was adopted as it is suitable of producing a fully functional prototype of the proposed system, including interaction and dynamic features like hyperlinks and buttons. [Figure 5](figure%20five) shows the Indigo studio software interface with an example of the prototype of the login page.

Indigo Studio makes it easy for anyone to rapidly design functional, UI prototypes that maximize usability and appeal. It promotes design process best practices and enables developers to stay focused on the end users and their stories [[[24]](#footnote-24)].

*Include some diagram*

**4.4 Evaluation of High Fidelity Prototype**

4.4.1 Participants

In total 15 participants were invited, only 10 were able to turn up, they were given the porotype to interact with and give their feedback and opinions. Each participant is asked to read participant information sheet (Appendix 4) defining the nature of the task and then sign a consent form (Appendix 5) indicating that they have agreed to participate in the study. Each participant was asked to respond to evaluation questions/survey (Appendix 6) based on their interaction with the prototype as shown in Figure 6 above. The survey includes a number of questions and statements, where each participant uses a five-point Likert Scale or multiple-choice questions with options to select any choice of their opinions on each question or statement and open ended question to describe based on their opinion.

The survey first asks their opinion on the aesthetics of the website. They were then asked to go through each of the main web pages of the prototype and were asked how reliable, easy and satisfied they are with each statement using the Likert Scale or multi choice options. Participants were given the option to provide any additional feedback at the end of the survey.

In to total 10 participants evaluate the prototype and responded to the evaluation questions. The responds to the question are analysed as follows:

4.4.2 Analysis of questions

The responses on the level of reliability of the proposed system is measured using the scale of 1 to 5 as indicated below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Highly not reliable | Not Reliable | Neutral | Reliable | Highly reliable |

* **How satisfied are you with the reliability of the proposed system?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 1 | 3 | 6 | 10 |
| **%** | 0 | 0 | 10 | 30 | 60 | 100 |

From the table above, 6 respondents representing 60% are satisfied that the proposed system is highly reliable, 3 respondents representing 30% are satisfied that it is reliable and 1 respondents representing 10% a neutral view. Therefore, 90% (majority) of the respondents are of the opinion that the proposed system is reliable.

Summarised comments from respondents:

* Having different role for every user bring decentralisation in the use of the system given every user different role and different access/view makes it reliable
* This is a unique platform for making and taking decisions regarding academic issues

The response on the level of interaction of the proposed system is measured in the scale of 1 to 5 as indicated below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Very little | Little | Neutral | Much | Very much |

* **How satisfied are you with the security of the proposed system?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 2 | 2 | 6 | 10 |
| **%** | 0 | 0 | 20 | 20 | 60 | 100 |

The table above shows that, 6 respondents representing 60% are very much satisfied with the security of the proposed system, 2 respondents representing 20% are much satisfied and 2 respondents representing 20% had a neutral view. Therefore, majority (80%) of the respondents are of the opinion that the security of the proposed is satisfactory.

Summarised comments from respondents:

* The system has limited number of users and every user has a unique way of accessing the system, therefore it is somewhat secured.
* The users can also be a threat and should be thought of as potential threat.
* **How satisfied are you with the ability to import data through excel (.csv) format?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 1 | 2 | 7 | 10 |
| **%** | 0 | 0 | 10 | 20 | 70 | 100 |

From the table above, 7 respondents representing 70% are very much satisfied with importing data through file in excel format, 2 respondents representing 20% are much satisfied and 1 respondent representing 10% had a neutral view. Therefore, 90% (majority) of the respondents are satisfied with the way data is imported from a file (excel).

Summarised comments from respondents:

* Importing a file brings sanity and can serve as a softcopy backup of students’ scores entered.
* Importing a file makes it easy to have all the scores entered at once.
* Eliminate some errors that might occur as a result of inputting scores via keyboard.
* **How satisfied are you with the look and feel of the software?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 2 | 2 | 6 | 10 |
| **%** | 0 | 0 | 20 | 20 | 60 | 100 |

From the table above, 6 respondents representing 60% are very much satisfied with the look and feel of the software, 2 respondents representing 20% are much satisfied and 2 respondents representing 20% had a neutral view. Therefore, 80% (majority) of the respondents are satisfied with the look and feel of the software.

Summarised comments from respondents:

* It has clear simple and well defined menus that remain static when operating the system
* The menus can always be referred to without having to switch windows.

The response on the ease of use of the proposed system is measured in the scale of 1 to 5 as indicated below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Very Difficult | Difficult | Neutral | Easy | Very easy |

* **How easy is it to understand how the application work?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 0 | 3 | 7 | 10 |
| **%** | 0 | 0 | 0 | 30 | 70 | 100 |

From the table above, 7 respondents representing 70% opine that it is very easy to understand how the application work, and 3 respondents representing 30% opine that it easy. Therefore, 100% (all) of the respondents observed that it is easy to understand how the application work.

Summarised comments from respondents:

* The application has an uncomplicated features with fully inscribed pages to enable user understand what to do and how to use the application.

The response on the level of interaction of the proposed system is measured in the scale of 1 to 5 as indicated below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Very low | Low | Neutral | High | Very high |

* **How do you find the level of interaction?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 0 | 4 | 6 | 10 |
| **%** | 0 | 0 | 0 | 40 | 60 | 100 |

From the table above, 6 respondents representing 60% propound that the level of interaction very high, and 4 respondents representing 40% propound that it high. Therefore, 100% (all) of the respondents opine that the level of interaction is high.

Summarised comments from respondents:

* The level of interaction of the proposed system makes it easy for users to explore the interface and get the best out of it functionality.

The responses on the level of satisfaction of the proposed system is measured using the scale of 1 to 5 as indicated below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Not at all satisfied | Not satisfied | Somewhat satisfied | Very satisfied | Extremely satisfied |

* **How satisfied are you with the registration processes (course registration, Student registration, staff registration) in this software?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 0 | 5 | 5 | 10 |
| **%** | 0 | 0 | 0 | 50 | 50 | 100 |

From the table above, 5 respondents representing 50% are extremely satisfied with the registration processes of the proposed system, and 5 respondents representing 50% are very satisfied. Therefore, 100% (all) of the respondents are satisfied with the process of the registration processes of the proposed system.

Summarised comments from respondents:

* The forms for registration in the system has inscription at the top to tell the user how to use the forms and also prompt users for wrong entries and empty fields when submitted.
* **How satisfied are with the broad sheet page?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 0 | 6 | 4 | 10 |
| **%** | 0 | 0 | 0 | 60 | 40 | 100 |

From the table above 4 respondents representing 40% are extremely satisfied with the format of the broadsheet page, and 6 respondents representing 60% are very satisfied. Therefore, 100% (all) of the respondents are satisfied with the format of the broadsheet page.

Summarised comments from respondents:

* The broadsheet page comprises of all details required to fully describe students results
* It shows both current and previous information of students’ results.
* **How satisfied are with the analysis page?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **1** | **2** | **3** | **4** | **5** | **Total** |
| **Respondents** | 0 | 0 | 3 | 2 | 5 | 10 |
| **%** | 0 | 0 | 30 | 20 | 50 | 100 |

From the table above 5 respondents representing 50% are extremely satisfied with the result analysis page, 2 respondents representing 20% are very satisfied and 3 respondents representing 30% are somewhat satisfied. Therefore, 70% (majority) of the respondents are satisfied with the result analysis page.

Summarised comments from respondents:

* The result analysis is tabulated showing analysis of each course per grade
* Including a tabulated summary analysis of the entire result
* It should include visualisation and also should be able to give analysis on a separate page with showing the results.
* **Were you in any point confused while exploring the software?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Yes** | **No** | **Not sure** | **Total** |
| **Respondents** | 0 | 9 | 1 | 10 |
| **%** | 0 | 90 | 10 | 100 |

From the above table 90% representing 9 respondents were not in any point confused while exploring the software and 10% representing 1 respondent were not sure. Therefore, majority (90%) of the respondent opine that they were not in any point confused exploring the system.

* **In your own words explain why you were confused**

No response, none of the respondents give their opinion

* **Does anything distract you or get in your way?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Yes** | **No** | **Not sure** | **Total** |
| **Respondents** | 0 | 10 | 0 | 10 |
| **%** | 0 | 100 | 0 | 100 |

From the above table 100% representing 10 respondents propound that there were no distractions while using the software. Therefore, all (100%) of the respondent opine that they were nothing get in there way while exploring the system.

* **In your own words explain what distract or get in your way**

All respondents suggest that there is no distraction

* **Does the information architecture and navigation make sense? (Can you find what you are looking for?)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Yes** | **No** | **Not sure** | **Total** |
| **Respondents** | 10 | 0 | 0 | 10 |
| **%** | 100 | 0 | 0 | 100 |

From the above table 100% representing 10 respondents propound that the navigation make sense. Therefore, all (100%) of the respondent find the navigation worthwhile as they can find what they are looking for.

* **Do you have any thought on how this software can be improve?**

Opinion of respondent on how the software can be improve

* Should be able to keep backups in case of loss
* Should protect the system from it users
* Let the analysis page include visualisation and graphical view of the analysis
* Provide search field for records
* **How would you describe this software using you own words?**

Participant respondent describing the system in their own words

* Good
* It almost perfect
* Simple/ plain user interfaces and helpful
* Look original
* **Any suggestions or observation?**

Suggestions made by respondents

* Make sure the software reach out the real users for implementation
* If the software is implemented, the changeover procedure should be in parallel.
* Use a plain white background, avoid dark colourful backgrounds

**4.5 Database design**

Database design is the organisation of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model [[[25]](#footnote-25)].

The proposed system was developed with a database as back end such that all the data working with the system are retained, resulting to information and knowledge. In order to access and manipulate the database a structured, standard and organised computer language, (Structured Query Language (SQL)) was used. Using SQL is nearly the only choice available when developing the database.

The development of the database was successful following the steps of modelling data in a domain; starting with the requirement analysis, conceptual, logical then physical data modelling.

4.5.1 Requirement analysis

From the requirements gathered in the earlier chapter, we were able to deduce what data to store, what data/information to retrieve, how to retrieve it, how the data would be displayed, the constraints on or between data items, and other requirements like speed

4.5.2 Conceptual Modelling

From what was deduced in the requirement stage, we were able to produce a conceptual model containing all of the major elements. Conceptual Model was created in such a way that it can be understood by the product owner. It is a framework that describes the problem space and demonstrates correct interpretation of concepts between client and designer. It is represented with an Entity Relationship (ER) diagram comprising the main data objects (entities), properties of the entities (attributes), links between entities (relationships). In this stage no consideration was made to satisfy any database design, because conceptual model is valid for implementation in any database type.

4.5.3 Logical modelling

In this stage we start by considering the type of database, taking account of the architecture of the database. Relational database was the choice of database type made.

In this model, detail of the tables, attributes, primary and foreign keys, linking tables and normalisation is described. This phase of modelling was more complex than the conceptual model. It is also represented with an ER diagram.

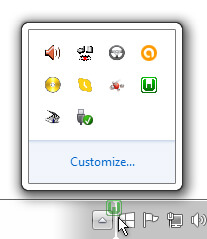
4.5.4 Physical modelling

This is final and actual blue print of the database, the actual database (Relational Database) is specified, this phase represent how data were related and structured in a specific Database Management System (DBMS). This made it important considering to the convention and restriction of the DBMS used during the design of the physical ER diagram, accurate use of data names, data type, of an entity and the use of reserved words were avoided, including additional constraint and integrity like primary keys, foreign keys were also specified.

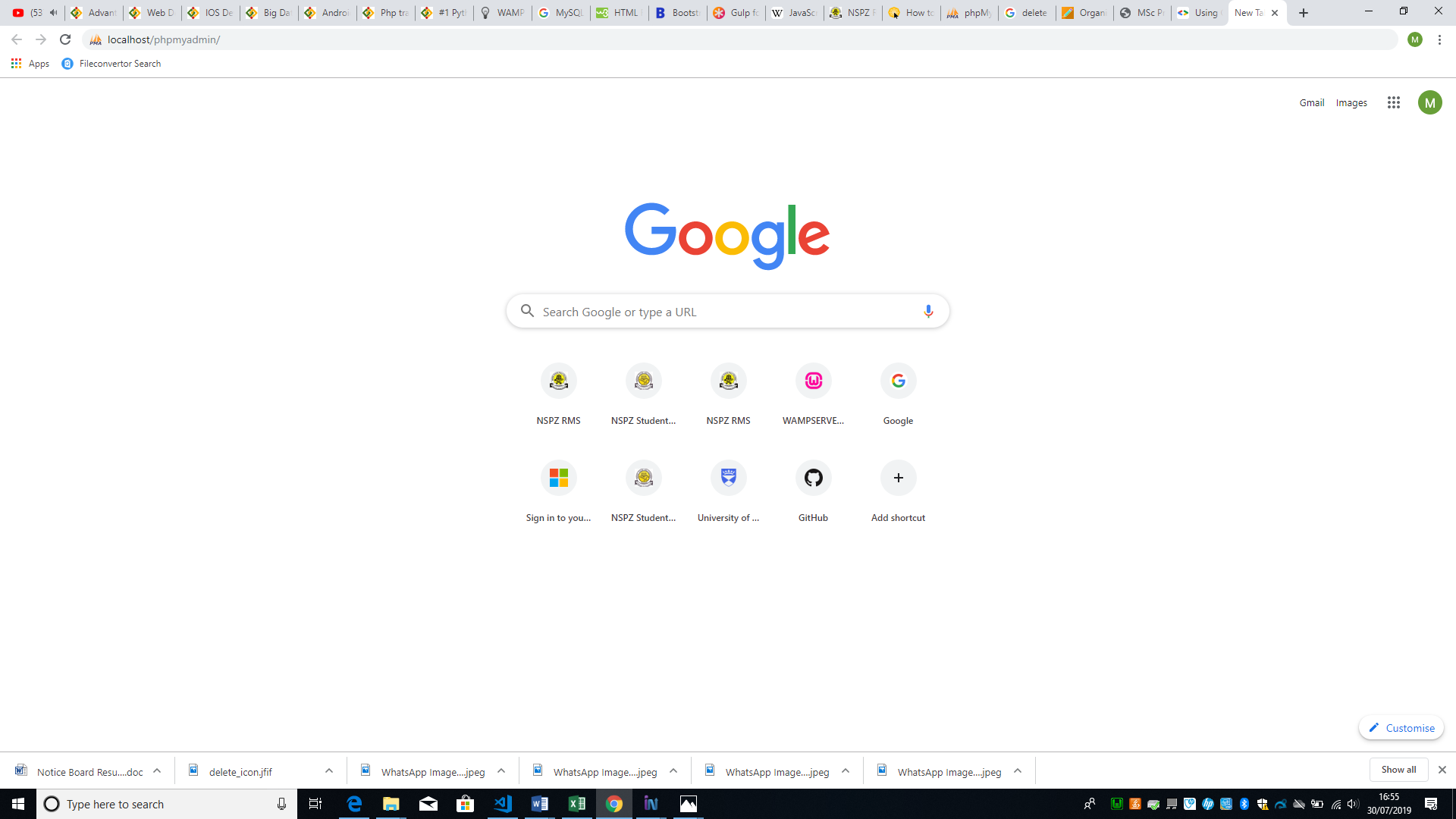
Visual paradigm was used as tool to create the ER diagram, this tool is customised to make it easy in creating the physical database. [See the ER diagram in appendix.](er%20diagram)

The database was created finally on localhost Wampserver[[26]](#footnote-26) from Phpmyadmin on my PC. After installing Wamp server. [See appendix](Wamp%20serve) … on how to do the installation. The following steps are followed to create database on localhost Wamp.

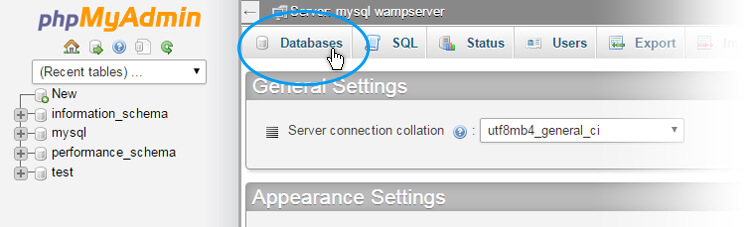
* First run the Wamp Server, as the Wamp Server icon turns green in the task bar menus,



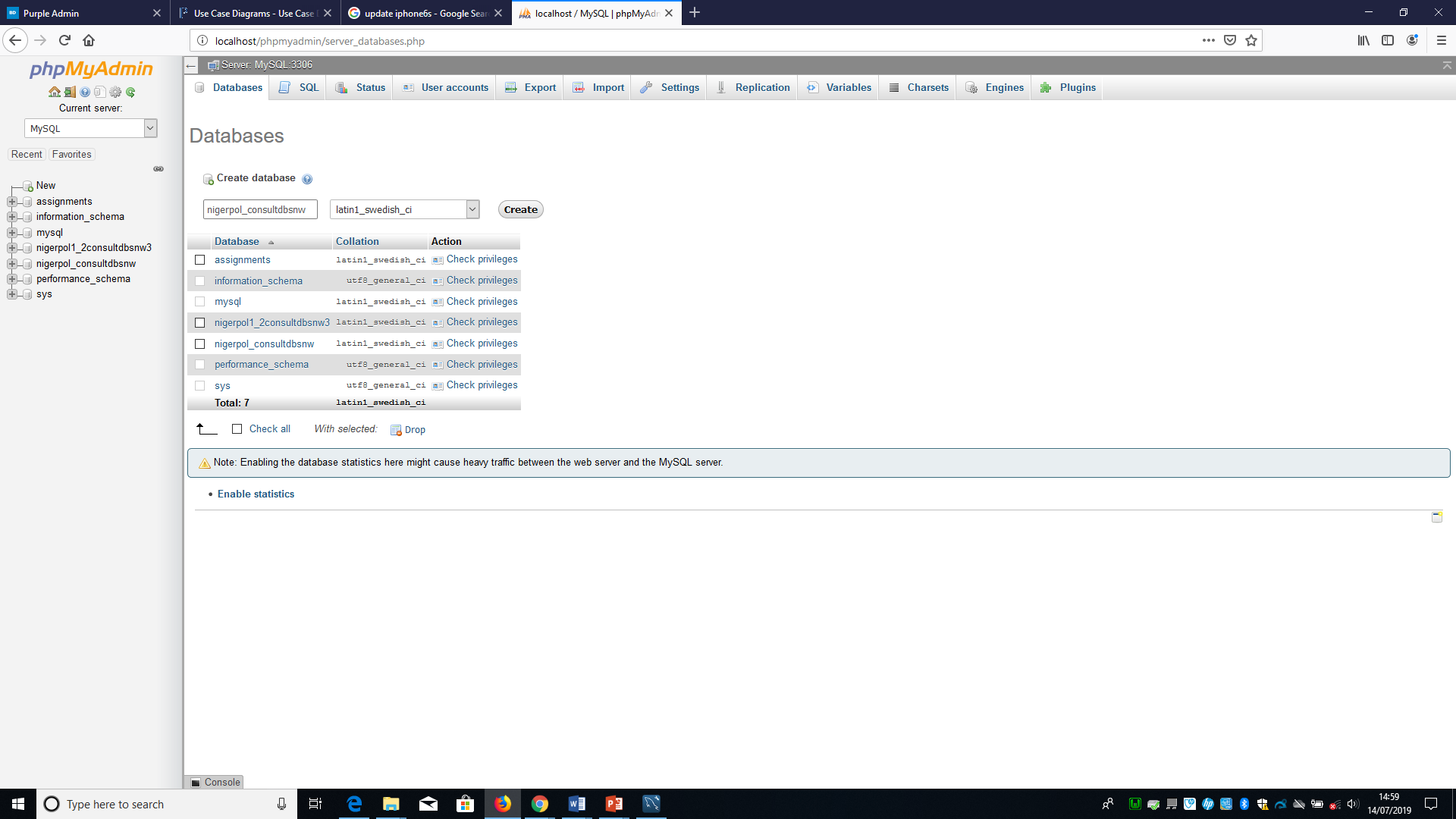
* Open the browser and type http://localhost/phpmyadmin



* Select the database menu as marked in the figure below

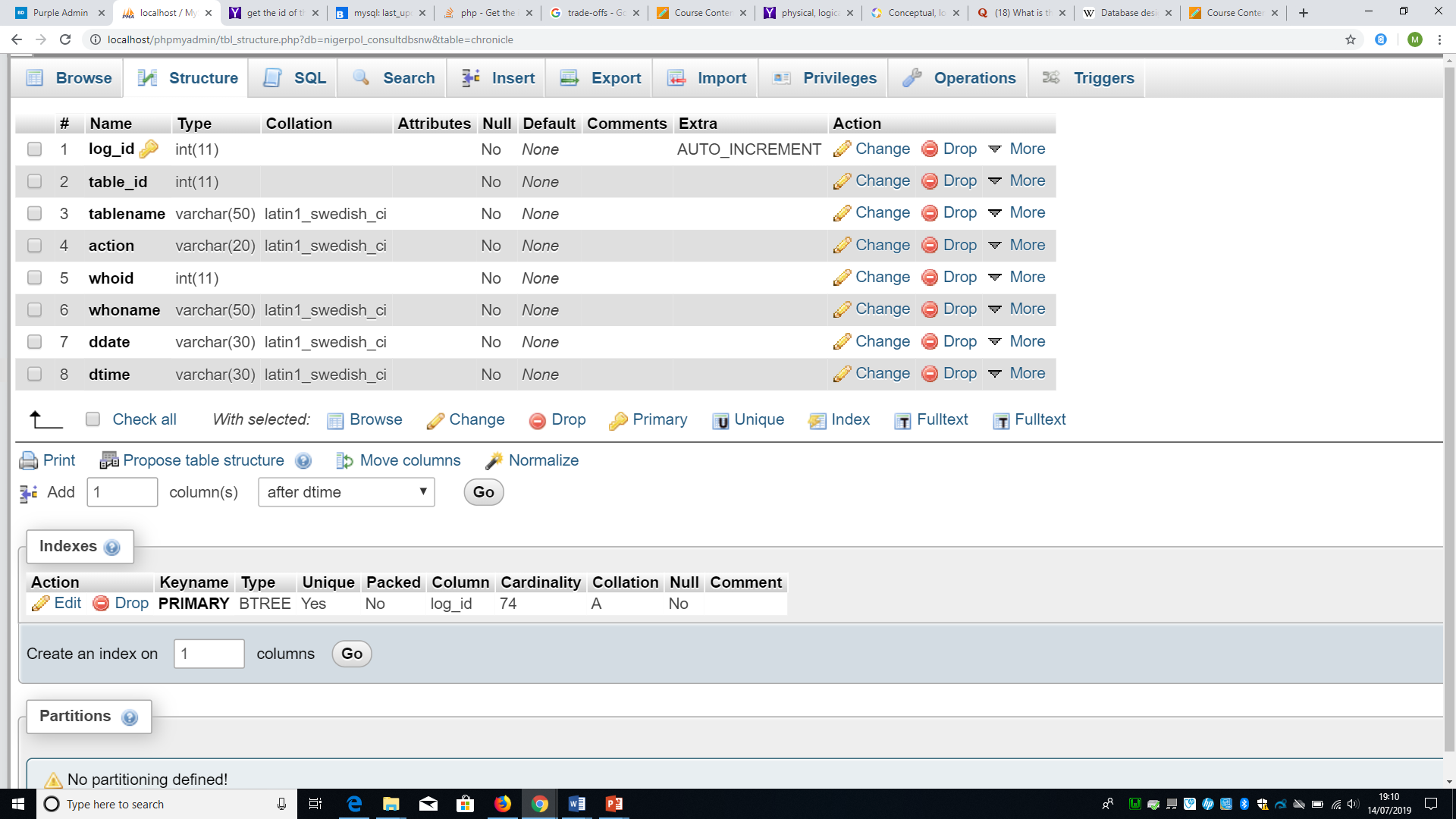


* Type the database name ‘nipoly\_consultdbsnw’, then click ‘create’ button (by clicking on the button database is created)

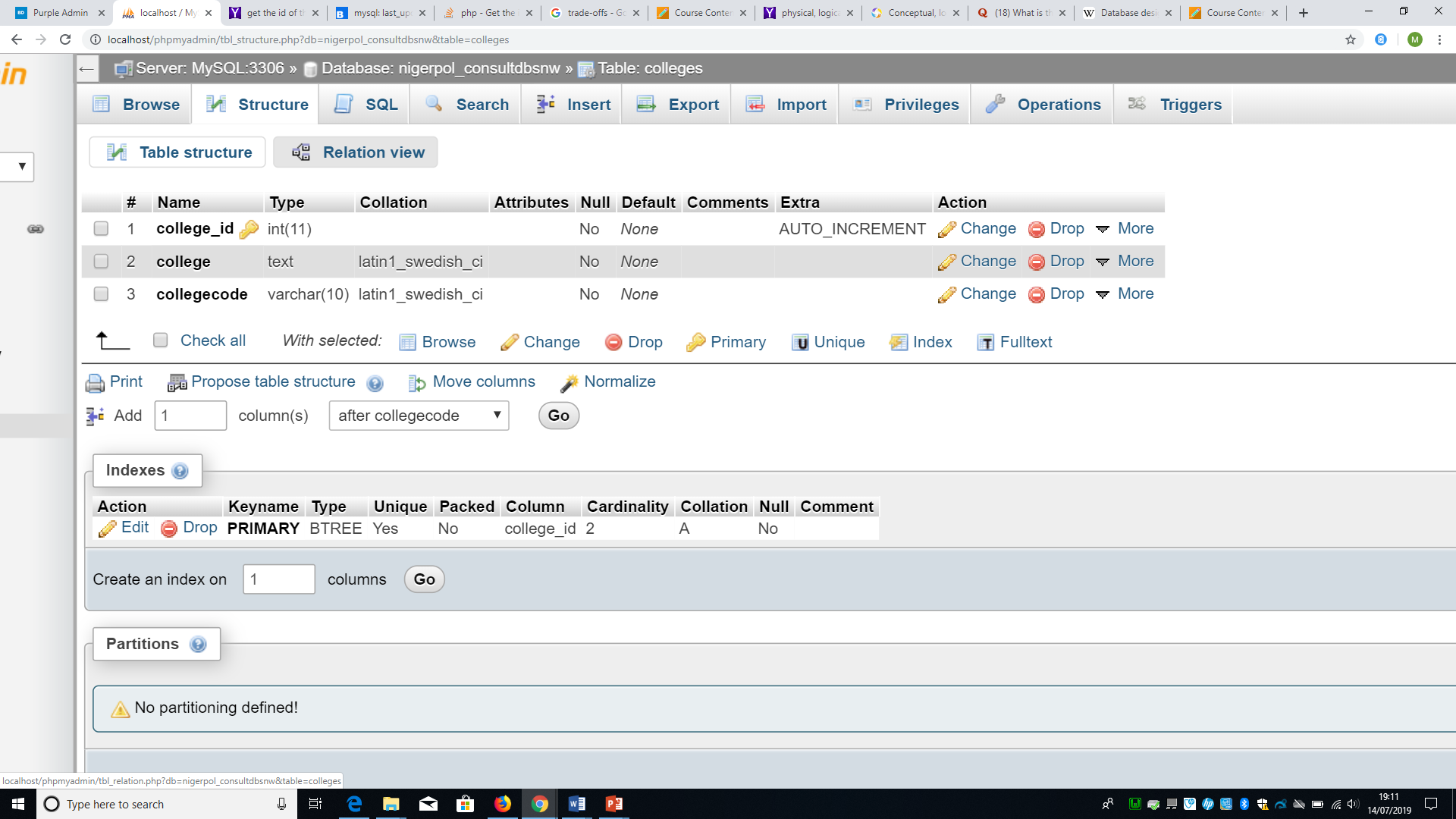


On the left side you can find the newly created database. To create table of database ‘nipoly\_consultdbsnw’, Sql statement was written and saved with “.sql” extension, the file is then imported to create tables. [See appendix](Apendix) to view the SQL statement used to create the following table; login table to hold user login credentials, colleges table to hold the descriptions of colleges, courses table to hold the courses offered by student, logs table to hold user logs, school table to hold description schools in a college, department table to hold the description of departments in a school, programmes table to hold the description of programmes in a department, student table to hold student data, staff table to hold staff data, user type table to hold the description of user types, and session table to record of academic session, below is the structure of the these tables.

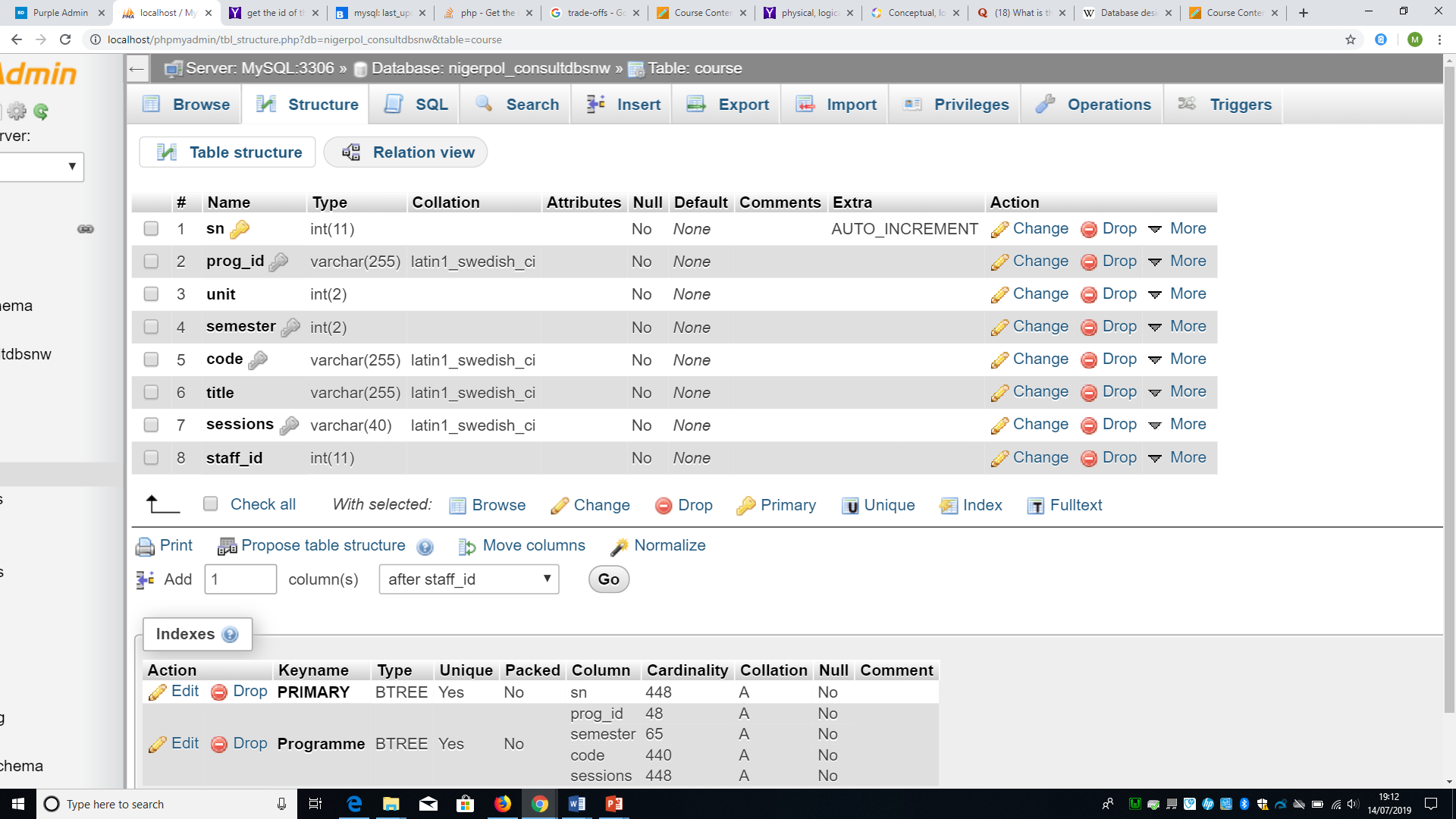
* Logs table



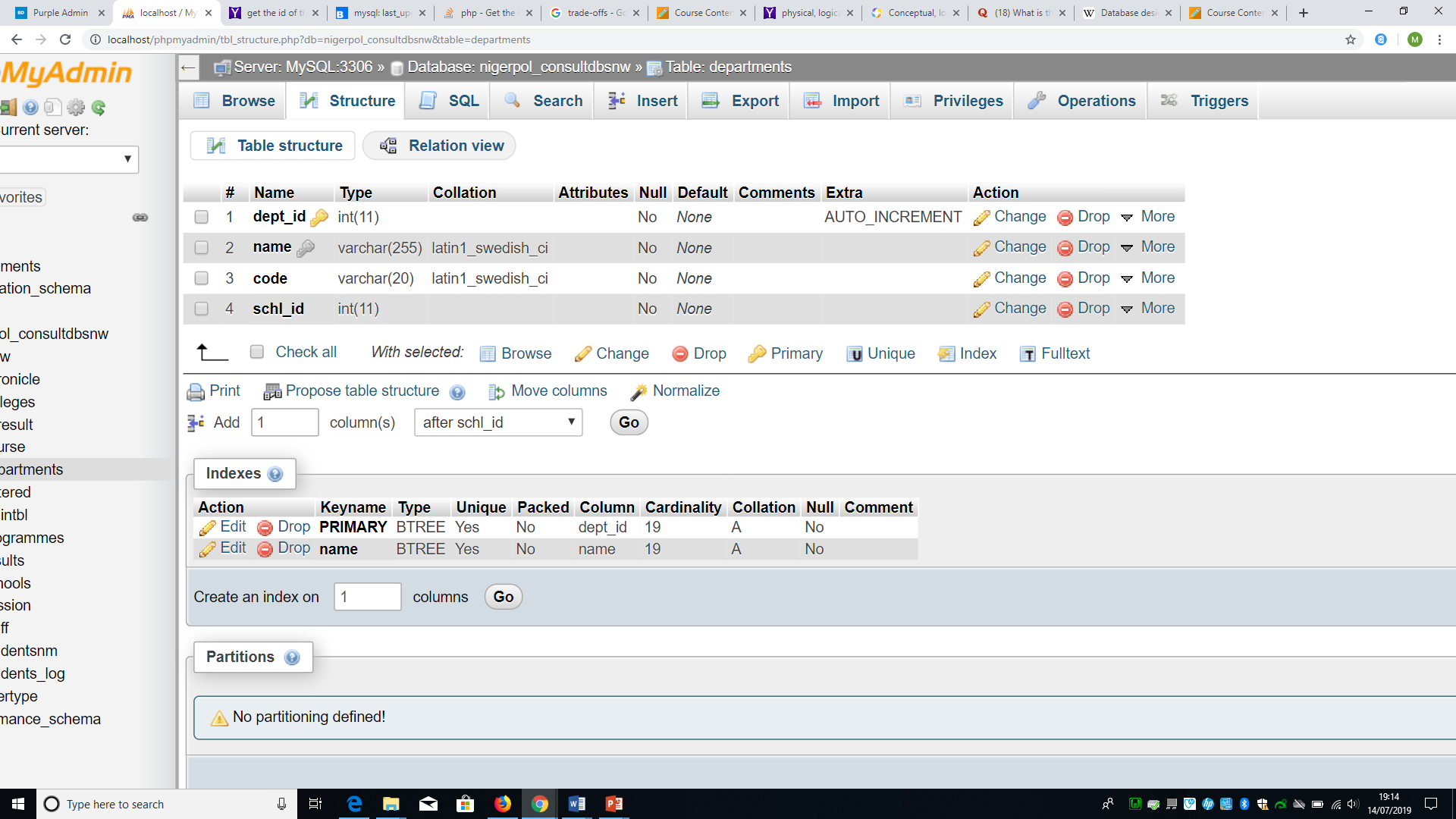
* Colleges table



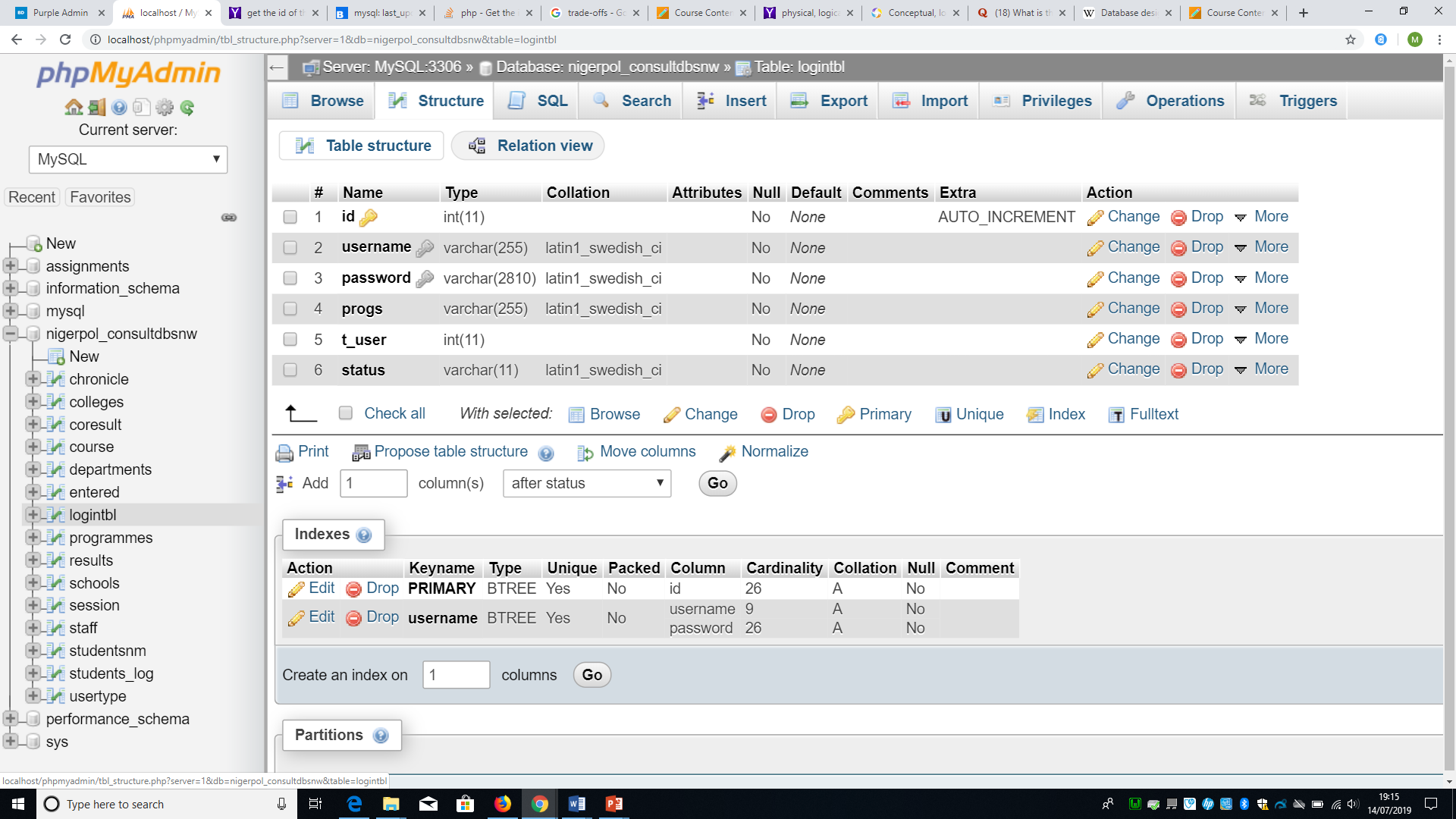
* Courses table



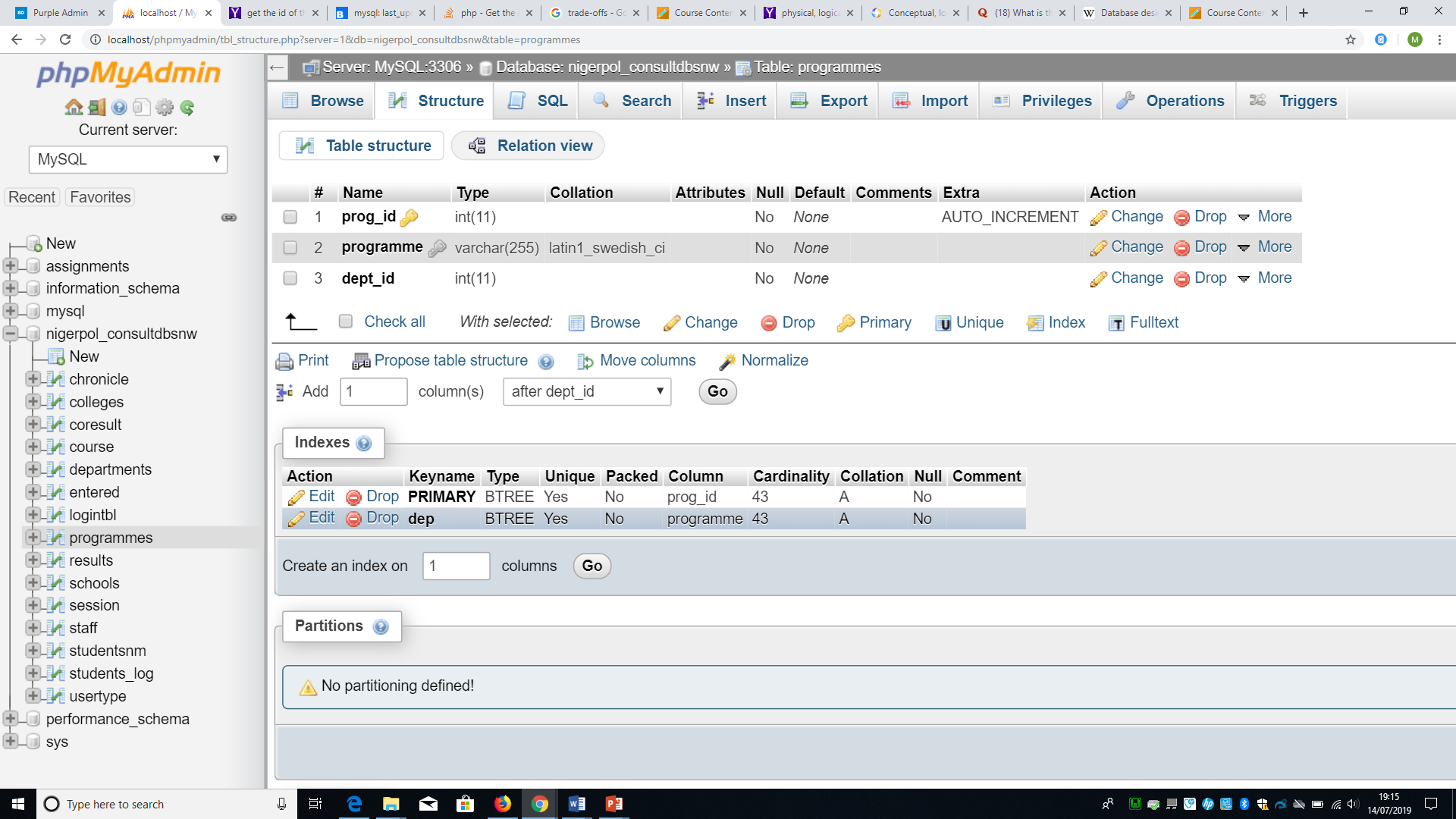
* Department table



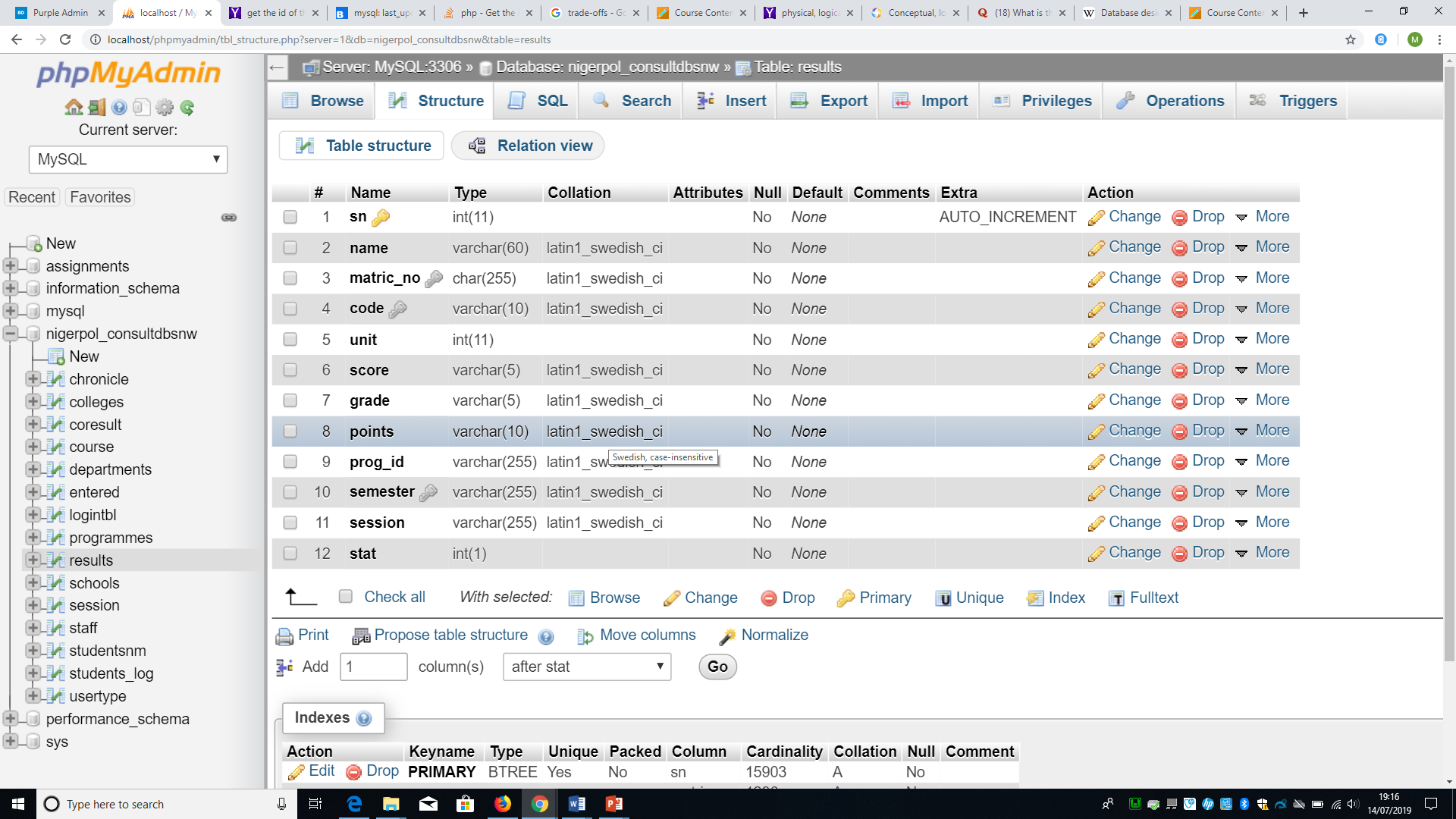
* Login table



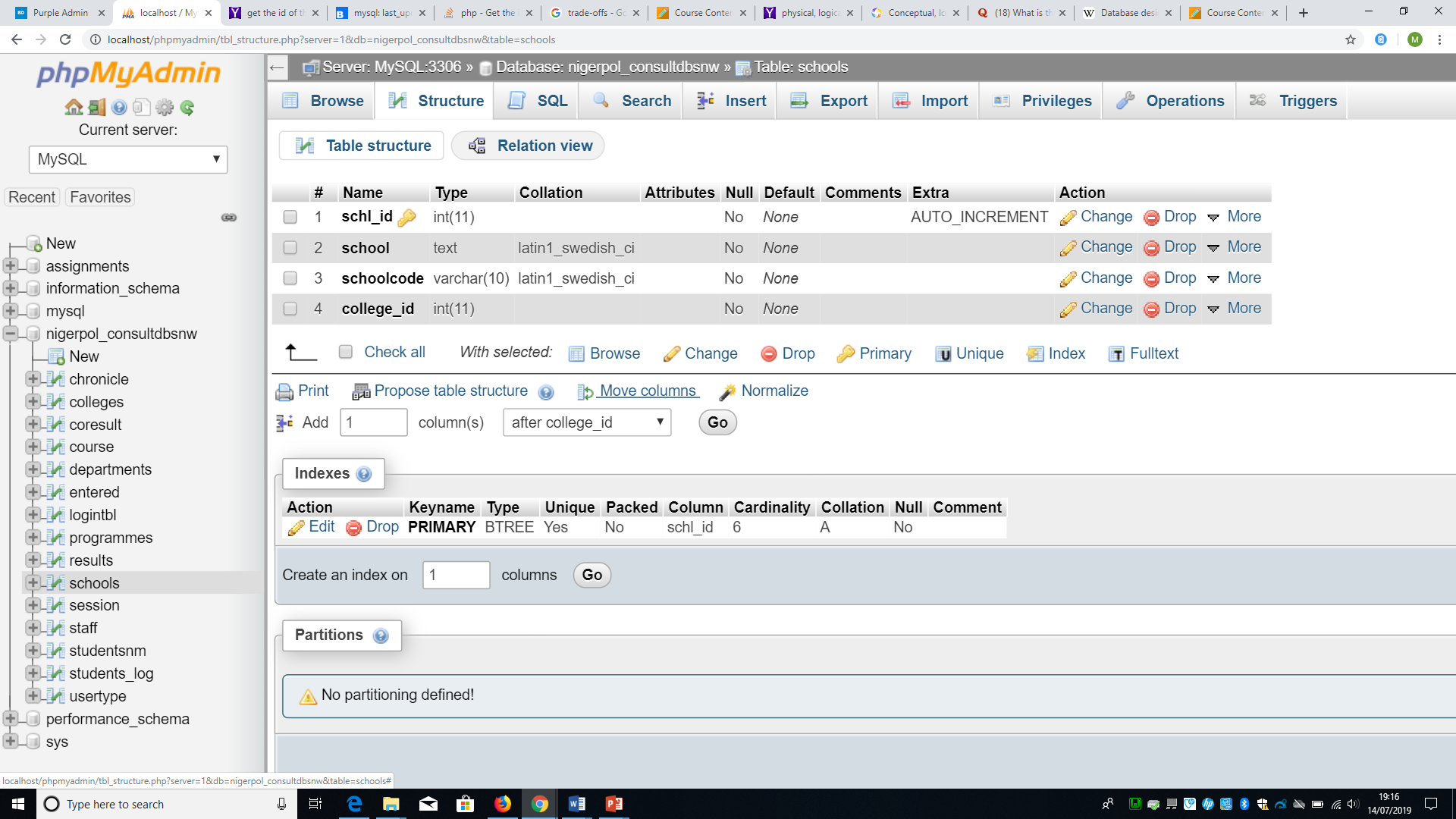
* Programmes table



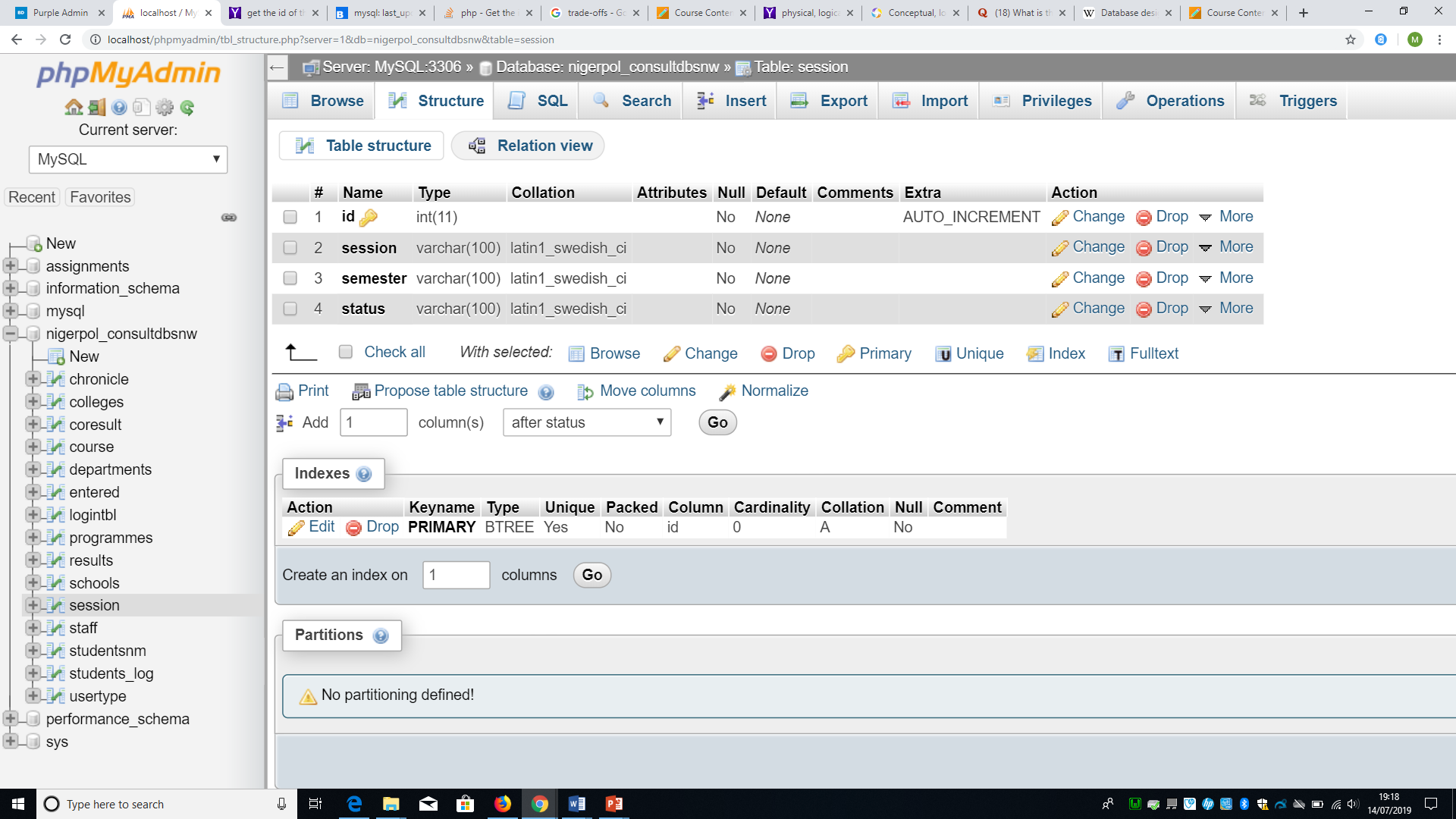
* Result table



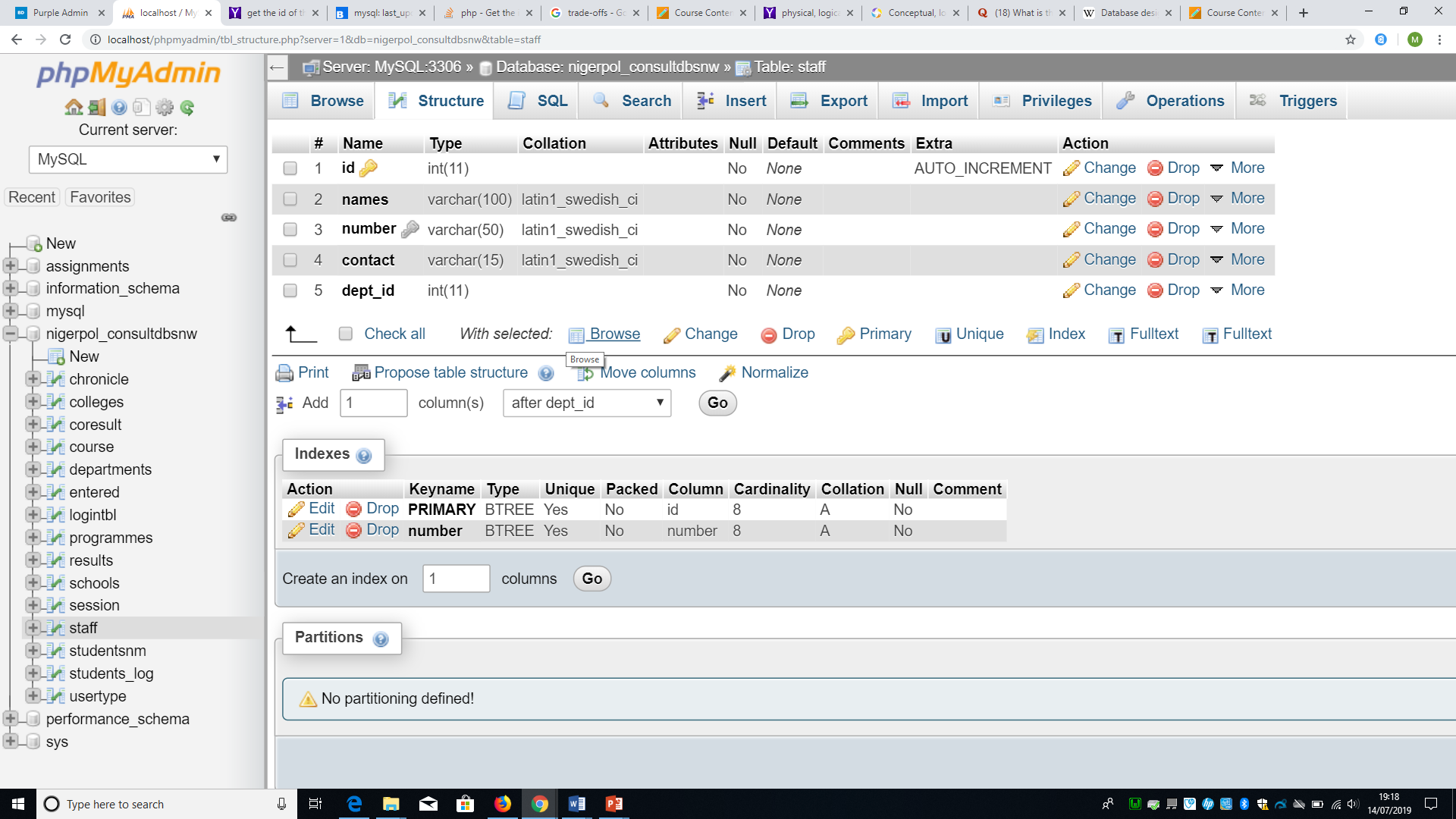
* School table



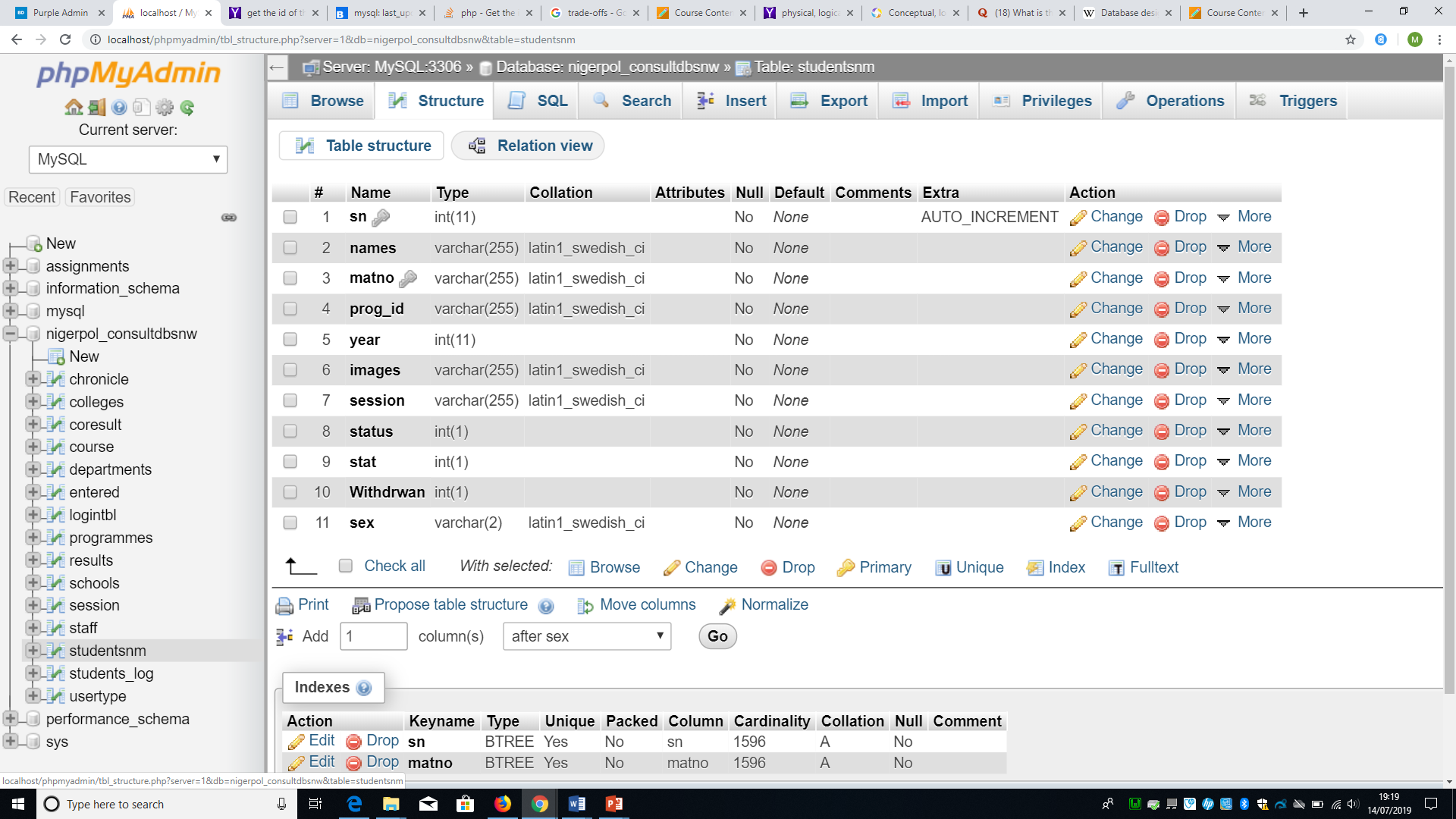
* Session table



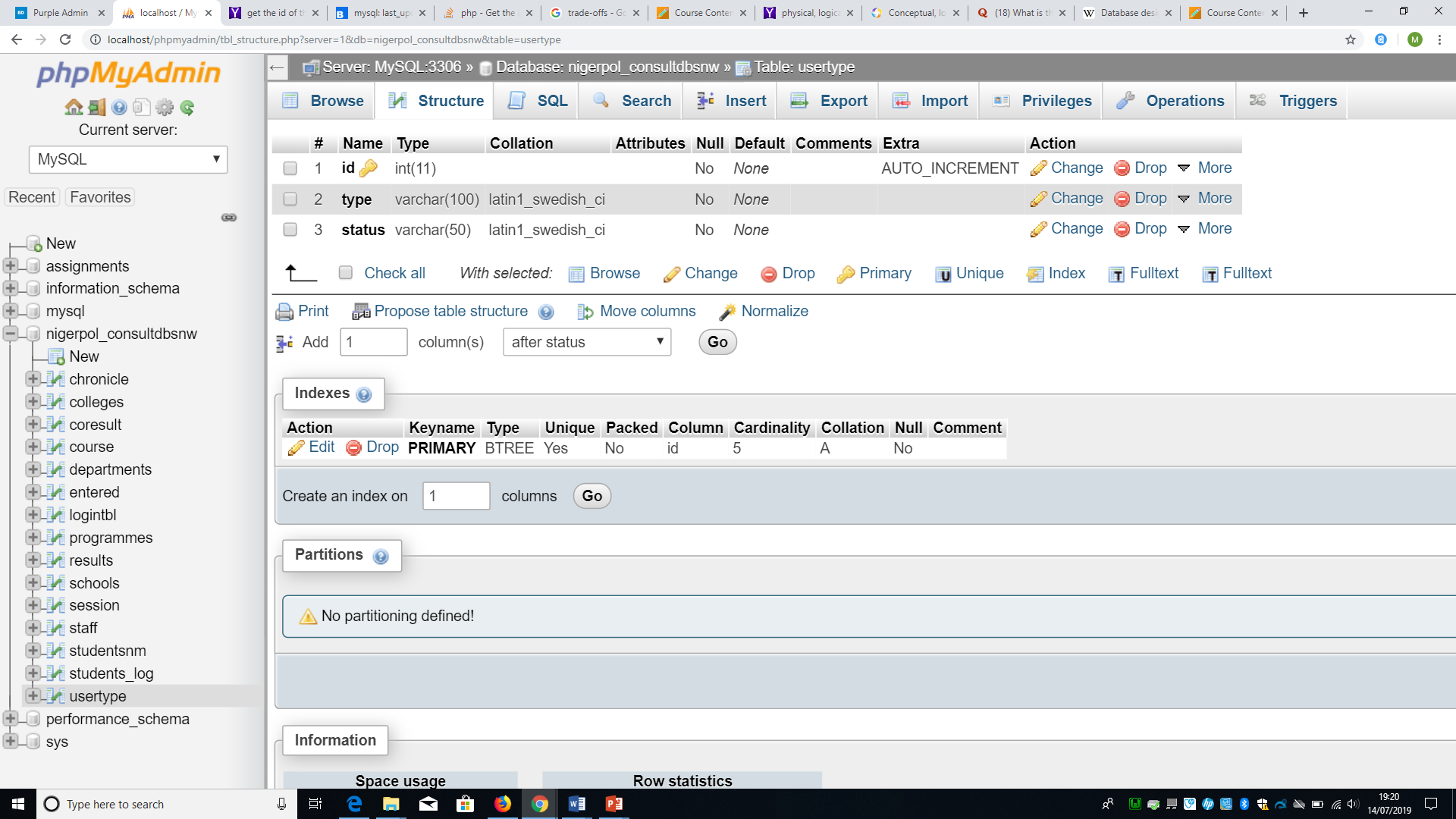
* Staff table



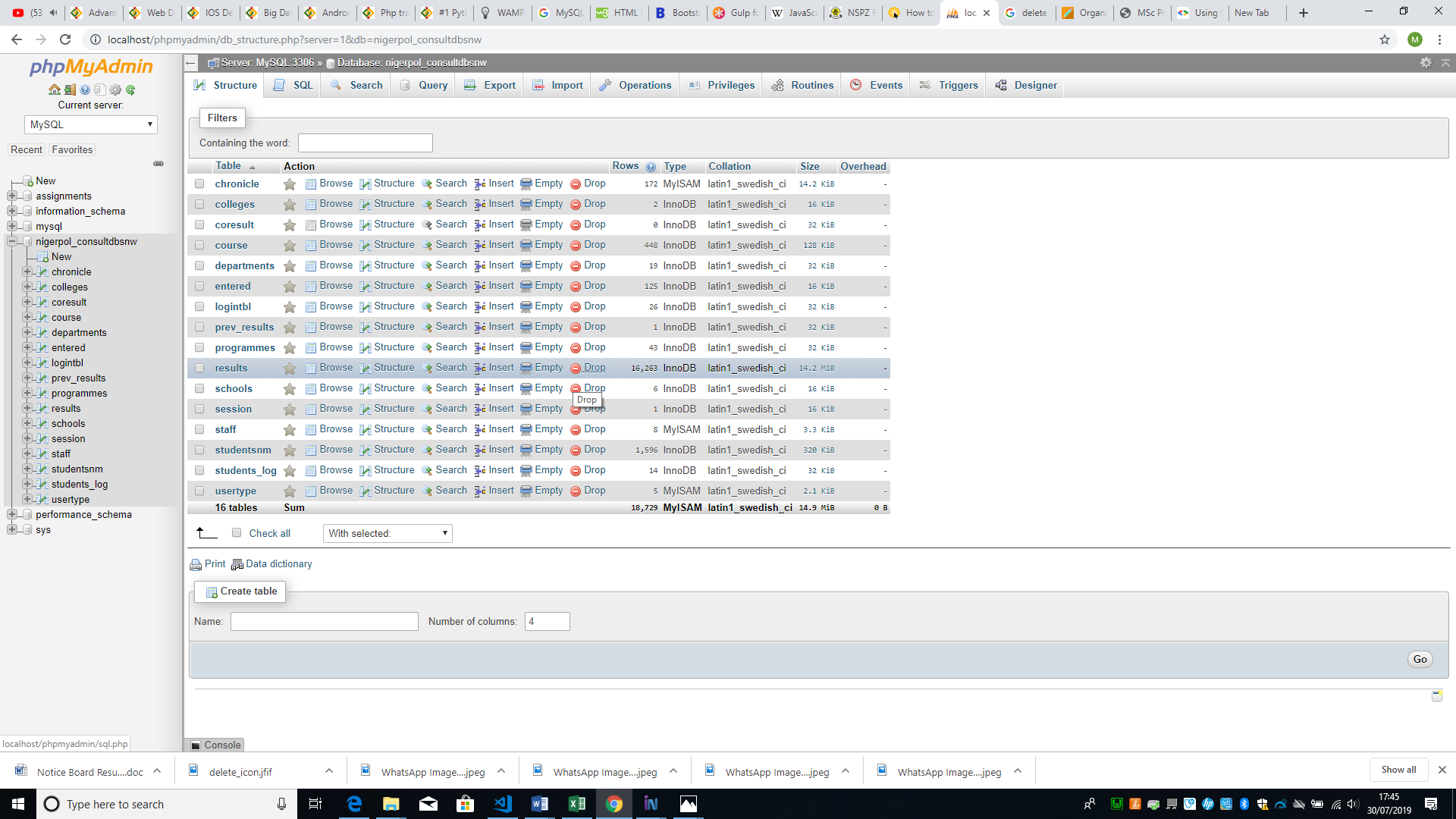
* Student table



* User type table



The structure of the entire database comprises of 16 table as shown below:



**4.6 Design process**

The design process follows an agile approach as mentioned earlier in the report. These processes lead to interpreting the requirements as list of stories[[27]](#footnote-27) prioritised in product backlog, then pick from the product backlog the stories with the highest priority into the sprint backlogs for implementation. An artefact of scrum[[28]](#footnote-28) known as product backlog and sprint backlog from sprint[[29]](#footnote-29) planning is adopted, though we are not up to the required number to form a scrum team. We adopted these artefacts because of it benefit and important in the development process of software. Product backlog consist of list of all things that need to be done within the project (“to do” list), it is created from the user story descriptions derived from requirements. From product backlog sprint backlogs are created.

Sprint backlog is a subset of product backlog, is more detailed than the product backlog. See product backlog and sprint backlog in [appendix](backlogs)

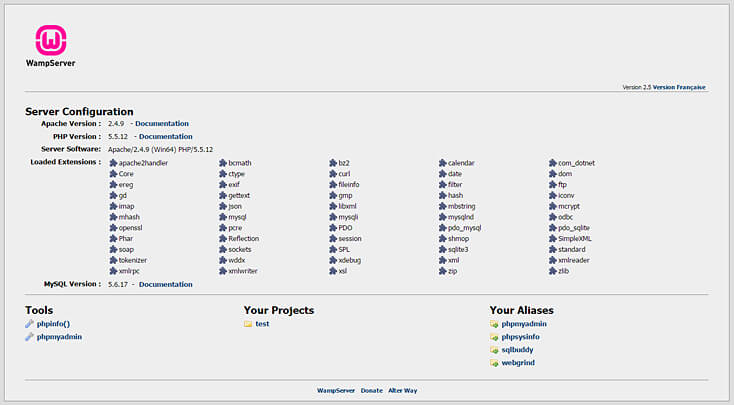
The total number of sprint in this project is 5, in each sprint we made different sprint backlogs until all the stories in the product backlog were exhausted. Though due to some unforeseen challenges some of the stories that we could not complete in any sprint is moved into the next sprint with the highest priority. Every sprint last for 1 week and at the end of every sprint we do retrospective and sprint review together with my supervisor to answer the 3w questions, what went well? What went wrong? What should be done next?

**4.7 Technology used**

The technologies used in this project are to enable the developer develop the proposed through frameworks/environments including APIs and scripting/programing languages. These technologies are explained as follows.

**4.7.1 Localhost Wamp server**

Wamp is a technology used for development and testing the website on a server locally hosted on user PC. Other alternatives include XAMP and LAMP. The decision to use Wamp was because it allows experiment with code, design, plugins and frameworks safely[[30]](#footnote-30). It could be dangerous to design and maintain tasks on a live website. There is more to lose depending on how large the website. Therefore, it is better to use localhost to avoid nasty surprises. After WAMP is downloaded[[31]](#footnote-31) and installed in user PC, there will be no need to install MySQL database. Wamp have MySQL included and is accessed through PhpMyadmin as show above.



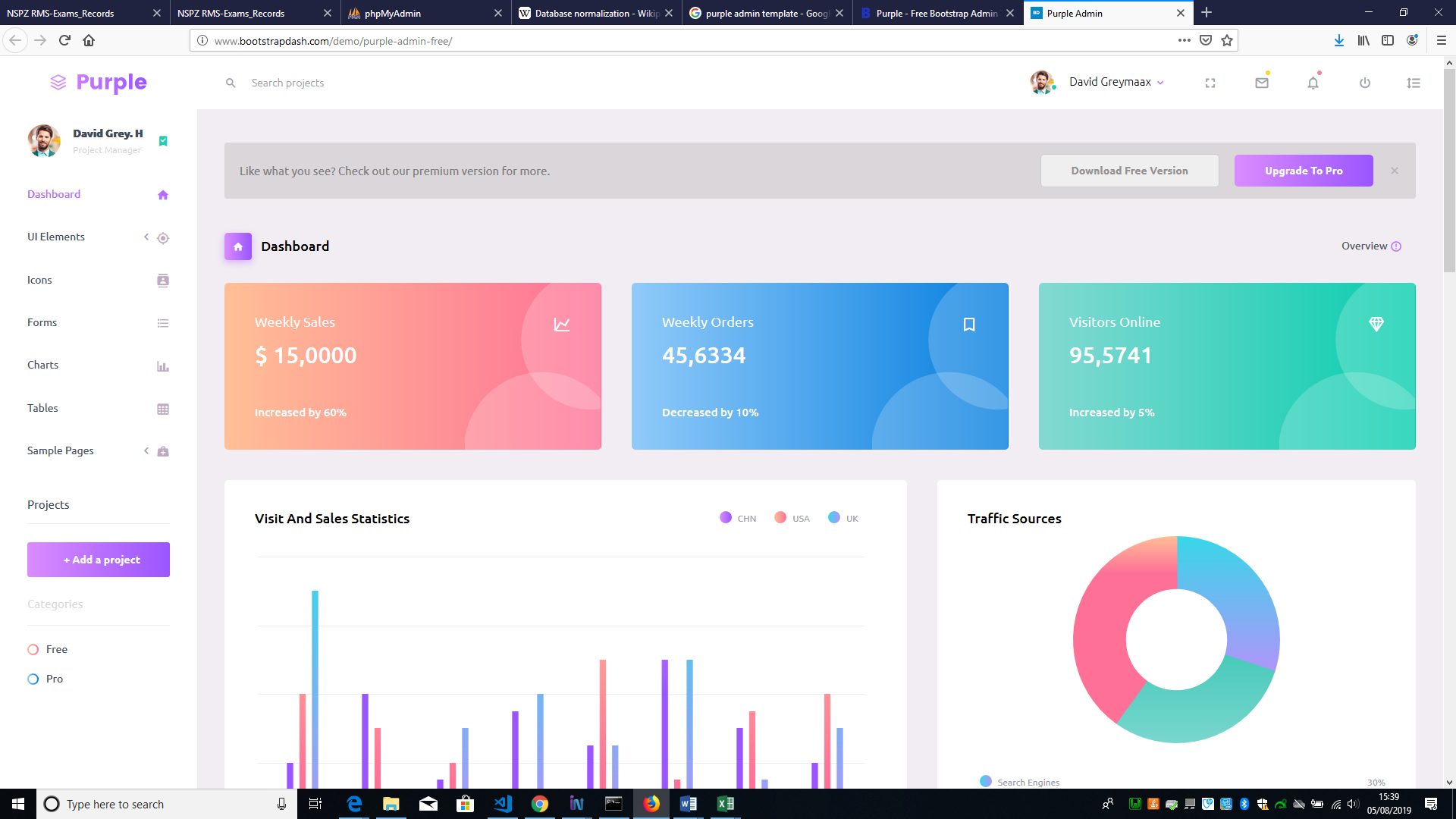
4.7.2 Server side scripting language

The Server side scripting language used in the development of the proposed system is PHP embedded in HTML. It was originally designed in 1994 by Rasmus Lerdorf, it is a general purpose server side programming language used primarily for the purpose of developing websites. Initially known as Personal Home Page, now called Hypertext Pre-processor. One of the most popular social networking sites such as Facebook and a well-known educational institution are both based on PHP, therefore, making it popular and increasing its credibility. Sites developed using PHP can be maintained, improved updated with ease periodically[[32]](#footnote-32). The reasons behind PHP been popular is the following advantages; cross platform[[33]](#footnote-33), easy database connection[[34]](#footnote-34), easy to use, and an open source[[35]](#footnote-35).

4.7.3 Back end and front end

The technology used for the back end is MySql, provided by Oracle, it is an open source relational database management system (RDBMS[[36]](#footnote-36)) based on Structured Query Language (SQL)[[37]](#footnote-37). In virtually all platform MySQL can be used, these include Windows, UNIX, and Linux. It is usually associated with online and web application. Mysql is a significant component of LAMP, LAMP is a web development environment that uses Linux OS, a web server (Apache), relational database management system (MySQL) and PHP as the scripting language.

The technology used for the development of front-end is HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets), these are two of the core technologies used for designing websites[[38]](#footnote-38).



In order to achieve a responsive design[[39]](#footnote-39), a purple-free-admin template was adopted. Purple-free-admin is designed by bootstrap Admin Dashboard Templates, built on Bootstrap; a popular JavaScript framework for developing responsive pages. Using npm[[40]](#footnote-40), Gulp[[41]](#footnote-41), Bower, Sass and Git, development was made easy, application are build faster.[[42]](#footnote-42)

However, the template is used by web developer, because of it ready to use features, having structures that houses all other structures introduced during the process of the development. The template came with pages that can be customised to fit developer need. In the process of the development the developer uses only the index page of the template including the attached CSS page to retain it look. Out of the hundreds of templates designed by bootstrap admin dashboard templates, the developer choose the purple admin template, because of it simple touch, colour combinations, side bar navigational menu, most importantly it complete responsive nature.. Figure… above is the adopted template.

In addition, JavaScript was used to add some dynamic features such as; [event driven](https://en.wikipedia.org/wiki/Event-driven_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) (including [object oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [prototype based](https://en.wikipedia.org/wiki/Prototype-based_programming)) [programming styles](https://en.wikipedia.org/wiki/Programming_paradigm). JavaScript is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) used alongside with HTML and CSS to enables interactive [web pages](https://en.wikipedia.org/wiki/Web_page). To extend JavaScript features on the web application jQuery was used.

4.7.4 Visualisation / Charts

The technology used for visualisation in this project is known as Google charts tools (API). Google Chart API is an interactive web service used in creating [graphical](https://en.wikipedia.org/wiki/Graphics) [charts](https://en.wikipedia.org/wiki/Chart) from data supplied by users. It is conveniently embedded in the web page by simply using JavaScript and HTML tags (div or image tag) with an id in the web page. It is used by loading some google libraries, outline the data to visualise, customise the chart then finally create a chart object with an id used in any of the HTML tags

Google chart tools are powerful, simple to use, and free. It has a rich gallery of interactive charts and data tools like line charts, complex hierarchical tree maps. The gallery provide a varieties of ready to use charts types[[43]](#footnote-43).

4.7.5 GitHub

GitHub is an online tool that is used to store files basically code on the cloud. It is not just a storage, but a full and advance version control system powered by Git. Linus Trovalds the founder of Linux, started as an open source version control system similar to version control systems like Mercurial, CVS and Subversion.

Git is helpful to developer when they create applications, they make changes to the code constantly to release new versions of the application after official release. Git as a version control system keep these revisions, storing the changes in a central repository which allow developers to cooperate enabling them to download new version of the application, make changes and upload the newest revision. All developer can see and have access to the new changes, downloads and make contributions.[[44]](#footnote-44)

However, GitHub is used in this project by the developer to keep records of all files including changes made during development process, keep track or get access to the files anywhere anytime. Among all version control system, Github is chosen due to the fact that most developers preferred using it and it has many advantages compared to other version control systems. It keep file changes more efficiently and better integrity is ensured. [See explanation on how Git works.](http://git-scm.com/book/en/Getting-Started-Git-Basics)

4.7.6 Accessibility Checker (AChecker)

AChecker is a free online open source web accessibility evaluation tool that was developed in University of Toronto in 2009 by the Inclusive Design Research Centre. On this tool, a user can upload a web page through its URL or by chosen uploading its HTML file locally from the user PC and can also select the guidelines in which the evaluate could follow such as; WCAG 2.0, WCAG 1.0, the HTML Validator, Stanca Act, Section 508, and BITV[[45]](#footnote-45).

However, there are many more accessibility checkers online, but the developer chose to use AChecker because it gives user different option of website submission methods, where user can either submit through the URL, upload the file or paste the row code and also has more accessibility guideless reference than other known tools.

**4.8 Design Decisions**

During the design of the proposed system decision where made leading changes that results to adding and correcting some features of the system. These decision are made for better development of the proposed system.

In the front end design, the proposed system was meant to have three user access roles which are; Students, Coordinator/Examination officer and teacher/course lecturer. To have proper account and control of the system for security reasons more roles were added, dividing roles of the coordinator into; System Manager and Examination & records. System manager control the entire menus accessible by other users, add, view and edit student, courses, programs, and staff records. The decision of creating more roles was made to reduce the task on user and allow proper management and control of the system. More also a decision was made to enable user generate a blank worksheet in .csv format for entering courses, student and exam score data. This decision was thought of as result of an observation made about user importing incorrect file format and wrong data column order.

In the back end design, the database initially comprises fourteen tables, but due to a decision made to create menu control and user logs, four more tables (user logs, menu control, submenu control, and user type tables ) where added. User logs table was created to keep logs, and keep track of all changes made by users in the database, and this will control miss use of the system by it users, mitigating unwanted changes that a user might make. The decision to create user logs was thought of when a use and misuse case diagram was drawn and documented. Menu and sub menu control tables were created to enable the system manager have full control of user menus, the system would be able to enable or disable user menu to deny them access to the functionality of the menu. User type table was created to ensure normalisation in the database to reduce data redundancy and improve data integrity by ensuring that dependencies are enforced properly[[46]](#footnote-46).

* Implementation and Testing: A description of production, testing and debugging. A demonstration that the specification has been satisfied.

**5.0 Implementation and testing**

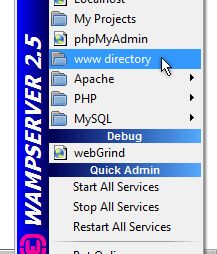
The development of the software follows an agile method as describe in previous chapters, all the items in the sprint backlog are implemented, strictly following the stories based on their priorities in the backlog. The developer uses Test Driven Development (TDD) to be able detect early problems and get quicker feedbacks in form of regressive testing. This type of test is also known as test first programming/development. However, the developer work in a very short cycle adding failing test[[47]](#footnote-47) and then making it work.

Beck, 2003 give two rules to be followed in testing as; “Never write a single line of code unless you have a failing automated test” and Eliminate duplicates.

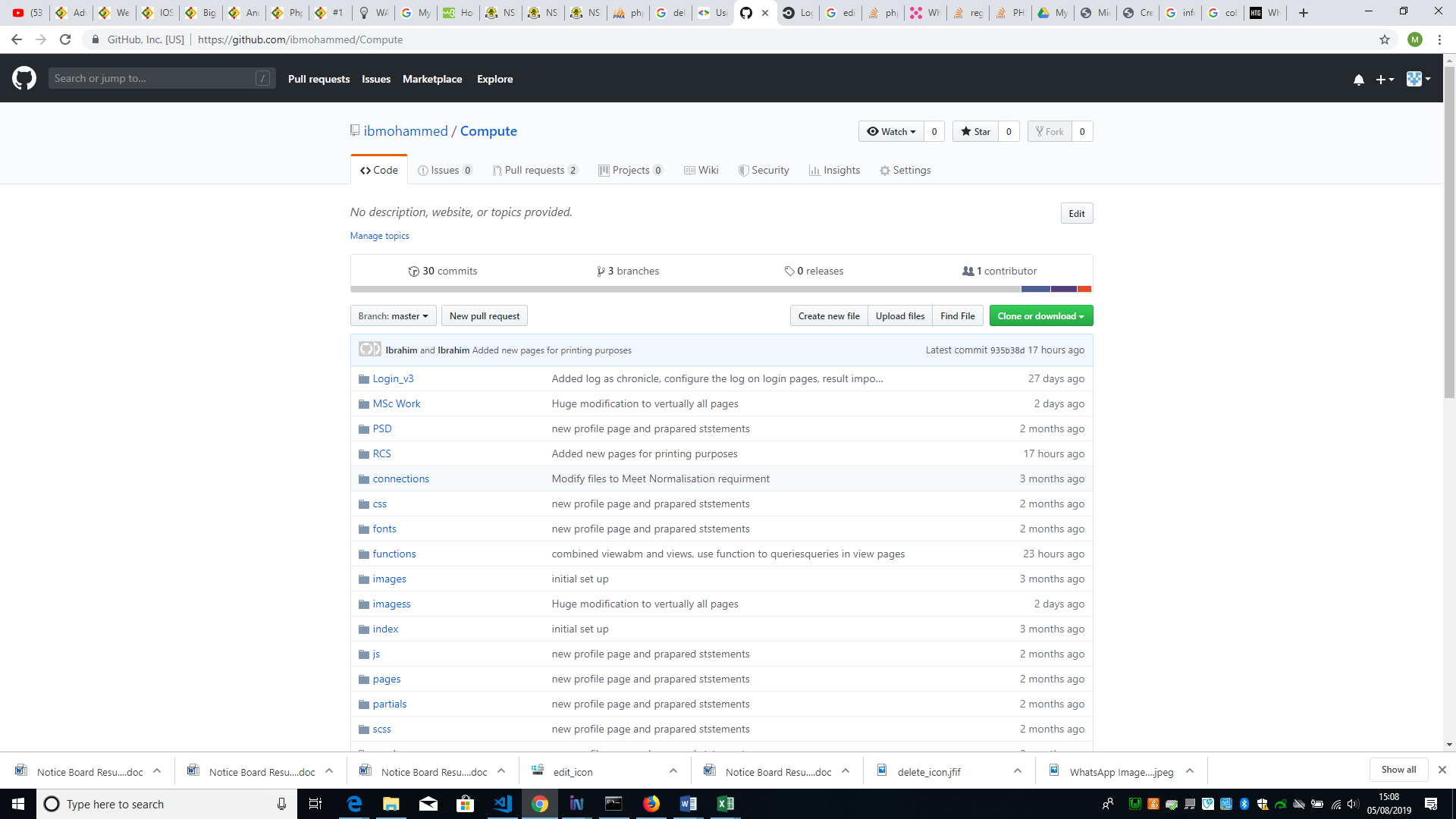
In TDD every pies of code is tested in the following process; building an automated failing test before writing a single line of code, only required code is written for the test to arrive as fast as possible at a working code, rewrite the code as soon as the code work then refactor[[48]](#footnote-48) it to eliminate code duplicate.

**5.1 Software development**

The development started with creating directory on the server in the www directory. The www directory is installed to **C:\wamp\www\** by default. The directory is accessed via the WAMP menu in the system tray as shown in the figure below.



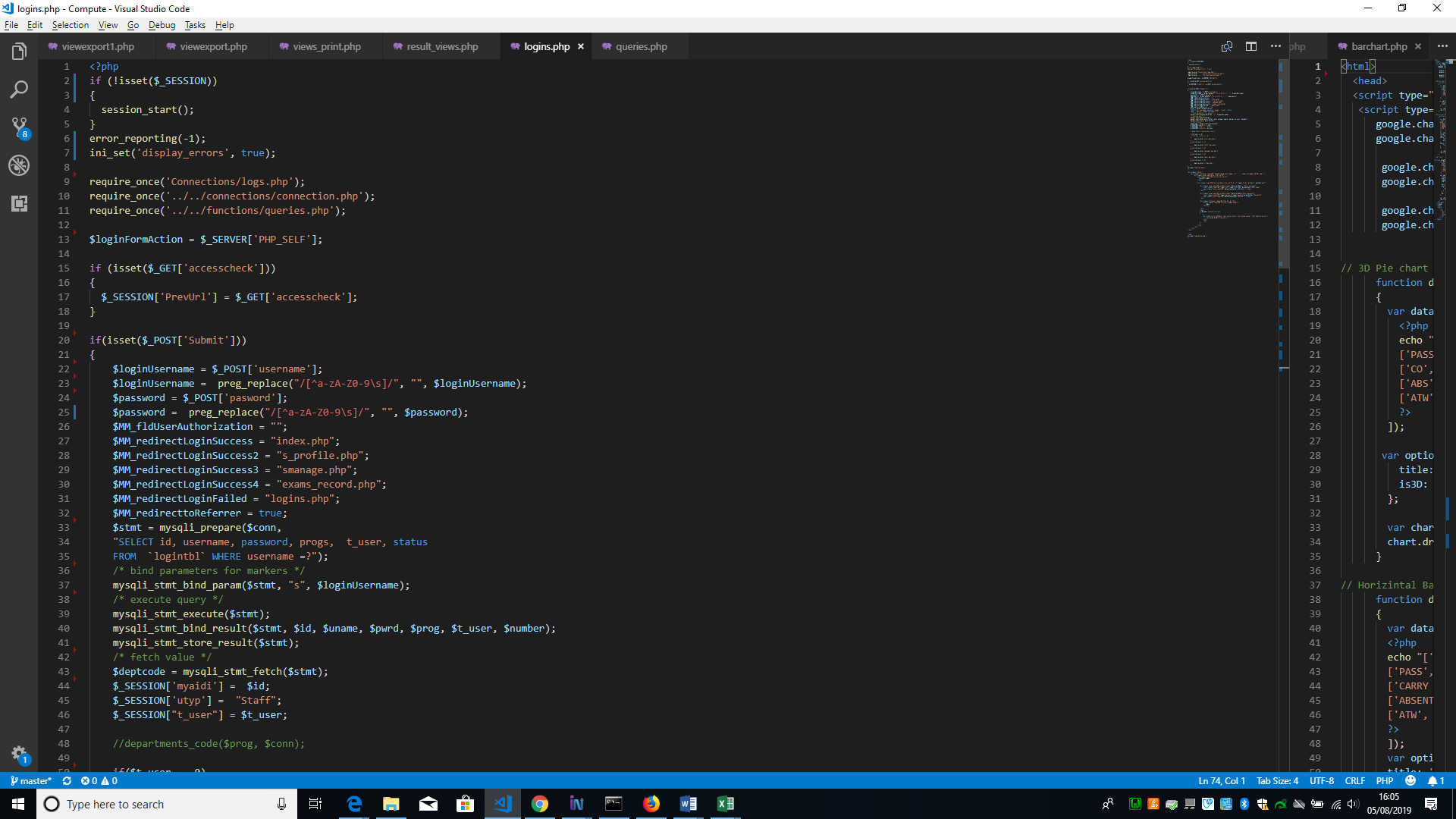
A new folder named “compute” is created in the **www directory**. All other pages and folders that comprise the project reside in the folder “compute” on the server. A repository of GitHub was created to keep all related files and folders save as shown in the figure below.



Following the list of requirements in the sprints, the sprint backlog has the list of task, estimated complexity, task, and estimated effort. The development start with the task from top to bottom of the table in order of complexity, all task completed are coloured in green, the task in progress are coloured in orange while task not started are coloured red.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Story | Est. Complexity | Done? | Tasks | Estimated Effort (Person hours) |
| **Sprint 1** |  |  |  |  |
| 1. Spike .csv file format upload to MYSQL |  |  | Spike |  |
|  |  |  |  |  |
| 2. As System manager, upload student records | 3 | Y | Create HTML upload form | 10 |
|  |  |  | Upload student record into database from file |  |
|  |  |  | Set up database structure |  |
|  |  |  |  |  |
| 3. As System manager, upload courses | 2 | Y | Create HTML upload form | 2 |
|  |  |  | Upload courses into database from file |  |
|  |  |  |  |  |
| 4. As System manager, create users | 2 | Y | Create form to add accounts, implement the add function | 1 |
|  |  |  |  |  |
| 5. As system manager, create programmes | 5 | Y | Create HTML forms |  |
|  |  | Y | Add programmes to database |  |
|  |  | Y | Add department to database |  |
|  |  | Y | Add school to database |  |
|  |  | Y | Add colleges to database |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 6. As exam officer upload exams score |  | Y | Create HTML form |  |
|  |  | Y | Upload scores into database |  |
|  |  | Y |  |  |
| 7. As a teaching staff, upload exam score |  | Y | Create HTML upload form |  |
|  |  | Y | Upload scores into database |  |
|  |  | Y | Display the Scores uploaded |  |
|  |  |  |  |  |
| **Sprint 2** |  |  |  |  |
| 1. create login for all user type |  | Y | Create form |  |
|  |  | Y | Connect to database to check |  |
|  |  | Y | Set session |  |
|  |  | Y | Hash user password |  |
|  |  | Y | Verify user hashed password |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 2. Spike Bcrypt password hash and verify |  | Y | spike |  |
|  |  |  |  |  |
| 3. As system manager, input student record |  | Y | Create HTML form |  |
|  |  | Y | Add records to database |  |
|  |  |  |  |  |
| 5. As exam officer, view broadsheet |  | Y | Create HTML Table |  |
|  |  |  | Map score to grade |  |
|  |  |  | Map score to point |  |
|  |  |  | Generate GPA, CGPA |  |
|  |  |  | Create remarks |  |
|  |  | Y | Implement Academic broadsheet result |  |
|  |  |  |  |  |
| 6. As system manager, create backup |  | Y | Create backup |  |
|  |  | Y | Import backup |  |
|  |  |  |  |  |
| 7. As Student, view my semester result |  | Y | Create html formatted page |  |
|  |  | Y | View student semester result | 1 |
| **Sprint 3** |  |  |  | 4 |
| 1. Allocate course |  | Y | View all department courses |  |
|  |  | Y | Choose and allocate course to staff |  |
|  |  |  |  |  |
| 2. Spike web visualisation API |  | Y | spike |  |
| 3. As exams & record, View result analysis |  | Y | Create table |  |
|  |  | Y | View analysis by grade in table result |  |
|  |  | Y | View analysis summary in table |  |
|  |  | Y | View visualise analysis |  |
|  |  | N | Visualise analysis comparing semesters |  |
|  |  |  |  |  |
| 4. As System manager, control user menu |  | Y | View all user menu |  |
|  |  | Y | Disable and enable user menu |  |
|  |  |  |  |  |
| 5. Form validation |  | Y | Validate user input, strip unwanted xters. |  |
| 6. Upload CSV format |  | Y | Ensure file format is .csv |  |
|  |  |  |  |  |
| 7. User logs |  | Y | Create user logs for input, edit, and logins |  |
|  |  |  |  |  |
| **Sprint 4** |  |  |  |  |
| 1. Single course analysis |  | Y | Visualise analysis of single course |  |
|  |  |  |  |  |
| 2. As System manager, view student | 3 | Y | Make a page with a table | 5 |
|  |  | Y | View student records | 3 |
|  |  |  |  |  |
| 3. As System manager, view courses | 2 | Y | Make a page with a table | 1 |
|  |  |  | View courses |  |
|  |  |  |  |  |
| 4. As system manager, modify user access |  | Y | View user | 6 |
|  |  |  | Modify user account |  |
|  |  |  |  |  |
| 5. As system manager, modify programme |  | Y | View programmes |  |
|  |  | Y | Modify programmes |  |
| 6. As system manager, generate file format |  | Y | Course upload .csv file format |  |
|  |  | Y | Student upload .csv file format |  |
|  |  | Y | Ensure the require data is uploaded |  |
| **Sprint 5** |  |  |  |  |
|  |  |  |  |  |
| 1. As Exam officer, generate file format |  | Y | Score upload .csv file format |  |
| 2. Change all user login detail |  | Y | Compare old user detail to new |  |
|  |  | Y | Make changes |  |
|  |  |  |  |  |
| 3. As System manager, view all user | 2 | Y | Make a page with table |  |
| 4. Create user views |  | Y | Create user views with different menus |  |
|  |  |  |  |  |
| 5. Logout |  | Y | Live page and destroy all sessions |  |
|  |  |  |  |  |
| 6. Export files to word |  | Y | Implement result export to word |  |
| 7. Export file to PDF |  | N | Implement result export to word |  |

After adopting the template the development of the login page start, it was completely independent of the adopted template as it was fully coded by the developer as shown below.



5.2 Security

Security is an integral part and important issue to any software especially when it involve storage of data or information that could be sensitive. Taking this into consideration the developer make provisions of possible mitigating or preventive measures to tackle issues involving threat or attack from malicious users such as the following:

5.2.1 Cross-Site Request Forgery (CSRF)

This is the type of hacking that occurs when a hacker tricks a user to load into a webpage from a site on which they are currently authenticated. Therefore, it becomes advantageous to the hacker, due to the user’s authenticated state. The developer use onetime token associated with a session in a post requests from a user. Ensuring mitigating or preventive measure for this type of attack is important.

5.2.2 Password Hashing

The hashing function use in this project is the “bcrypt password hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX in 1999”[[49]](#footnote-49). If this function is used on any phrase, it turns to a mathematically transformed random strings of characters, unreadable by human.

On the system whenever user is created, the passphrase submitted by the user is hashed using the bcrypt hashing function before submitting it to the database so that it cannot be misused.

The developer choses to use bcrypt because it can be implemented in PHP and is a function that is adaptive, it give room to increase it iteration count which makes it slower, therefore, remains resistant to brute-force searching attacks no matter the increase in computation power of an adversary.

5.2.3 SQL Injections

SQL Injection is “a code injection technique that might destroy your database” and “…is one of the most common web hacking techniques”[[50]](#footnote-50). This usually occurs when a malicious user input SQL Statements into a text field in a form, for instance username or email field, instead of username or email which will be run on the database. The developer mitigate this by validation of user inputs, striping unwanted characters from users input and ensure that only valid data has been inputted by the user. For example, a form has a date input field and if a user attempts to insert an SQL statement instead of a valid date, the system will prompt the user with a generic message like “Please enter correct data for all fields” automatically.

5.2.4 Cross-Site Scripting (XSS)

Cross-Site Scripting occurs when the content user submit is not escaped or cleared by the website before it is rendered into HTML. An attack of this nature can lead to users disposing vital or sensitive information to malicious user or hacker unknowingly. This is prevented by the developer using Contextual output encoding/escaping of string and validating untrusted HTML input.

5.3 Testing

As described earlier on in this chapter, every piece of story was tested as the developer uses a Test driven development approach, though the developer keep track of all test carried out during the development which is reported as shown below:

**Test Type: Unit Testing following agile method**

Unit testing is done to test the smallest variable or every functionality in the system during development.

**Test Summary**

|  |  |  |
| --- | --- | --- |
| Executed | Passed | 31 |
| Failed |  |
| Total test Executed  (Passed + Failed) | | 32 |
| Pending | | | 1 |
| In progress | | | 0 |
| Blocked | | | 0 |
| (Sub-total) test planned | | | 32 |
| (Pending + in progress + Test executed) | | |

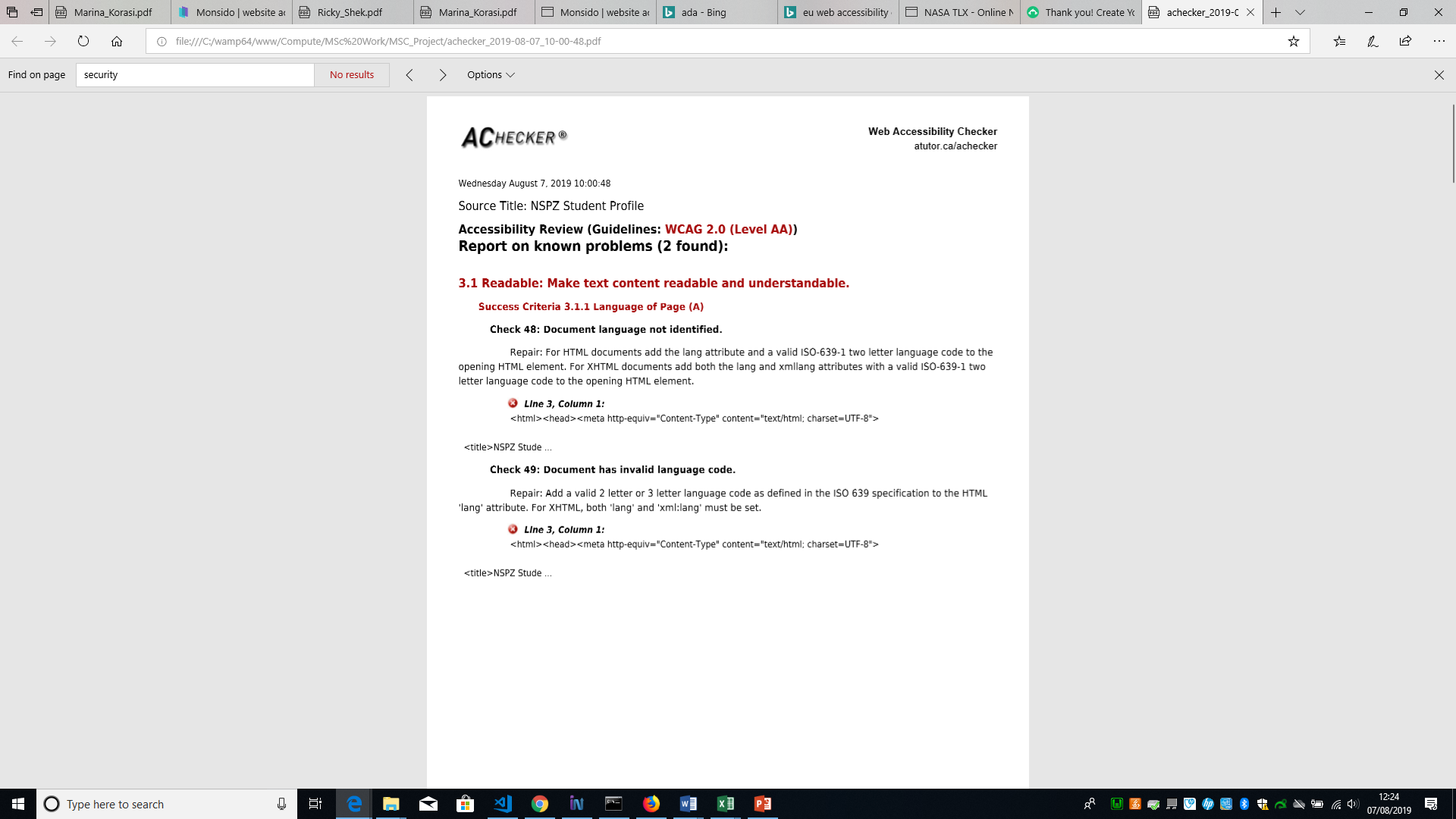
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Functions | Description | %TCs Executed | %TCs Passed | TCs pending | Priority | Remarks |
| upload student records | Check if record is added | 100% | 100% | 0 | High |  |
| upload courses | Check if record is added | 100% | 100% | 0 | High |  |
| create users | Check if user is created | 100% | 100% | 0 | High |  |
| Add new programmes | Check if added | 100% | 100% | 0 | High |  |
| upload exams score | Check if uploaded | 100% | 100% | 0 | High |  |
| login users | Verify if successful or not | 100% | 100% | 0 | High |  |
| password hash and verify | Verify if hashed and can be verified | 100% | 100% | 0 | High |  |
| input student record |  | 100% | 100% | 0 | High |  |
| Generate score Grades |  | 100% | 100% | 0 | High |  |
| Generate score point |  | 100% | 100% | 0 | High |  |
| view broadsheet |  | 100% | 100% | 0 | High |  |
| view broadsheet |  | 100% | 100% | 0 | High |  |
| create backup |  | 100% | 100% | 0 | High |  |
| view my semester result |  | 100% | 100% | 0 | High |  |
| Allocate course |  | 100% | 100% | 0 | High |  |
| web visualisation API |  | 100% | 100% | 0 | High |  |
| View result analysis |  | 100% | 100% | 0 | High |  |
| Enable or Disable user menu |  | 100% | 100% | 0 | High |  |
| Check file format |  | 100% | 100% | 0 | High |  |
| Create user logs |  | 100% | 100% | 0 | High |  |
| Single course analysis |  | 100% | 100% | 0 | High |  |
| View student |  | 100% | 100% | 0 | High |  |
| View courses |  | 100% | 100% | 0 | High |  |
| Modify user access |  | 100% | 100% | 0 | High |  |
| Modify programme |  | 100% | 100% | 0 | High |  |
| Generate file format |  | 100% | 100% | 0 | High |  |
| Edit user login detail |  | 100% | 100% | 0 | High |  |
| View all user |  | 100% | 100% | 0 | High |  |
| Control user views |  | 100% | 100% | 0 | High |  |
| Logout |  | 100% | 100% | 0 | High |  |
| Export files to word |  | 100% | 100% | 0 | Low |  |
| Export files to PDF |  |  |  | 1 | Low |  |

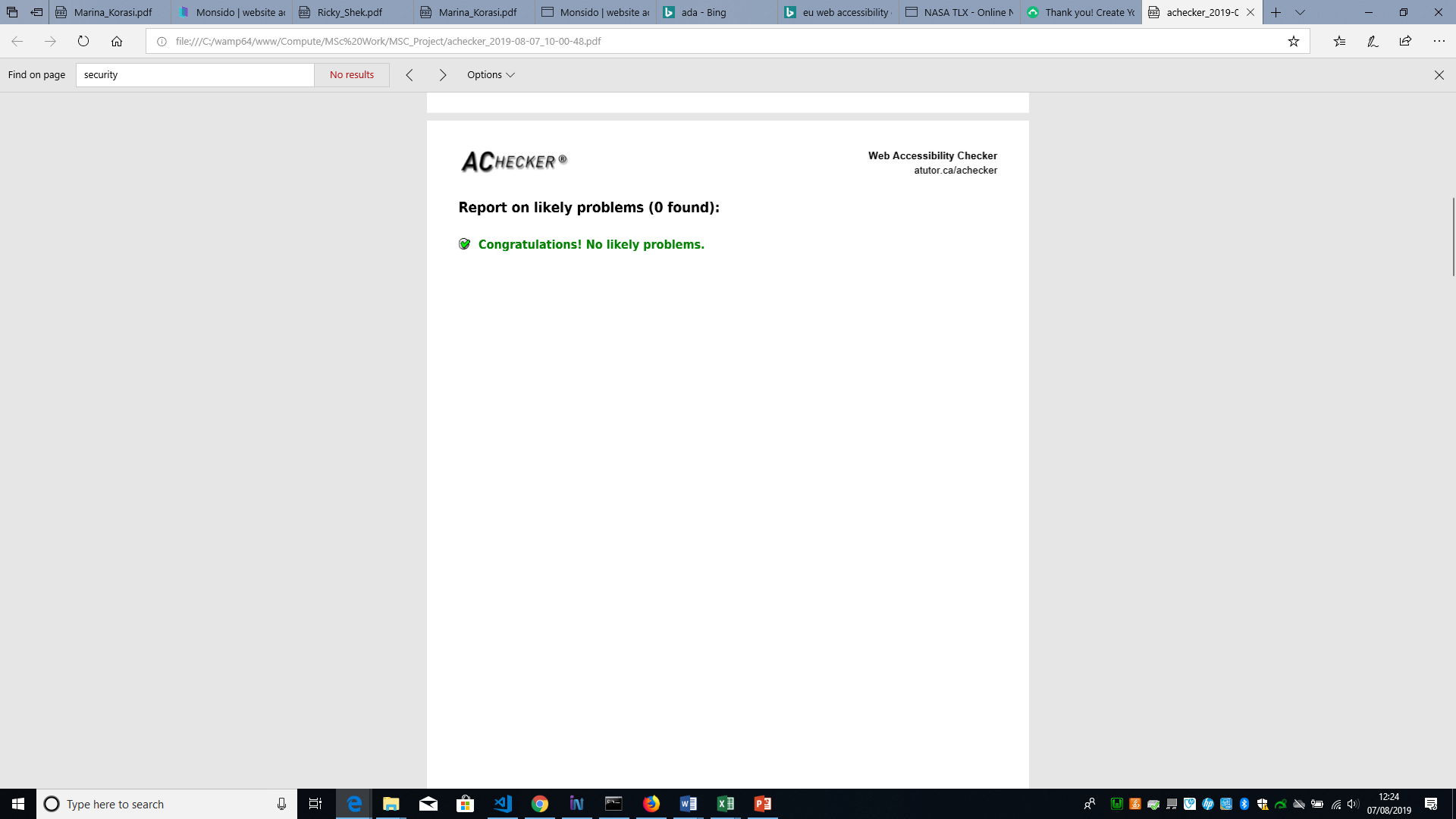
**Defects**

* The defect density is 1 defects/50 lines of code average this is because the testing follows an agile principle that says “no much code written than it is required” during the test.
* 98% defects fixed, this is as a result of following strictly following the agile principle by not implementing any function without testing and fixing possible defects.
* The detail of the bugs are described in this bug tracker in [appendix](https://docs.google.com/spreadsheet/ccc?key=0Au2ff2t4xXuGdFZlek9zbHhUcnVleHVyOVp0VmFHMmc&usp=drive_web#gid=0)

The report serve as document ensuring that the current status of project and quality of the product is informed, to enable corrective action to be taking if necessary, and finally determine whether the product is ready for release.

As previously stated in chapter 4 Accessibility Checker (AChecker) was use as a tool to test the web pages following the Web Content Accessibility Guidelines (WCAG) to improve Website Accessibility and the result is as follows:





See detailed report in <appendix>

The tool indicated the above “known” problems. Recommended that the document languages is added and should follow the “valid ISO-639-1 two letter language code” to the opening HTML element. It was added as shown below. Though this is not really a problem that necessarily frustrate user in terms of accessibility and by default the language is set to en meaning English.

<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

5.3 Final product

5.3.1 Evaluation of final product

The final evaluation of the web application was conducted using an online questionnaire targeted mostly to initial participants in the evaluation of the higher fidelity porotype. The questions were transferred to an online survey tool (SurveyMonkey5) to have better layout arrangements and easily administered to the participants. The questionnaire was open to participants for one week.

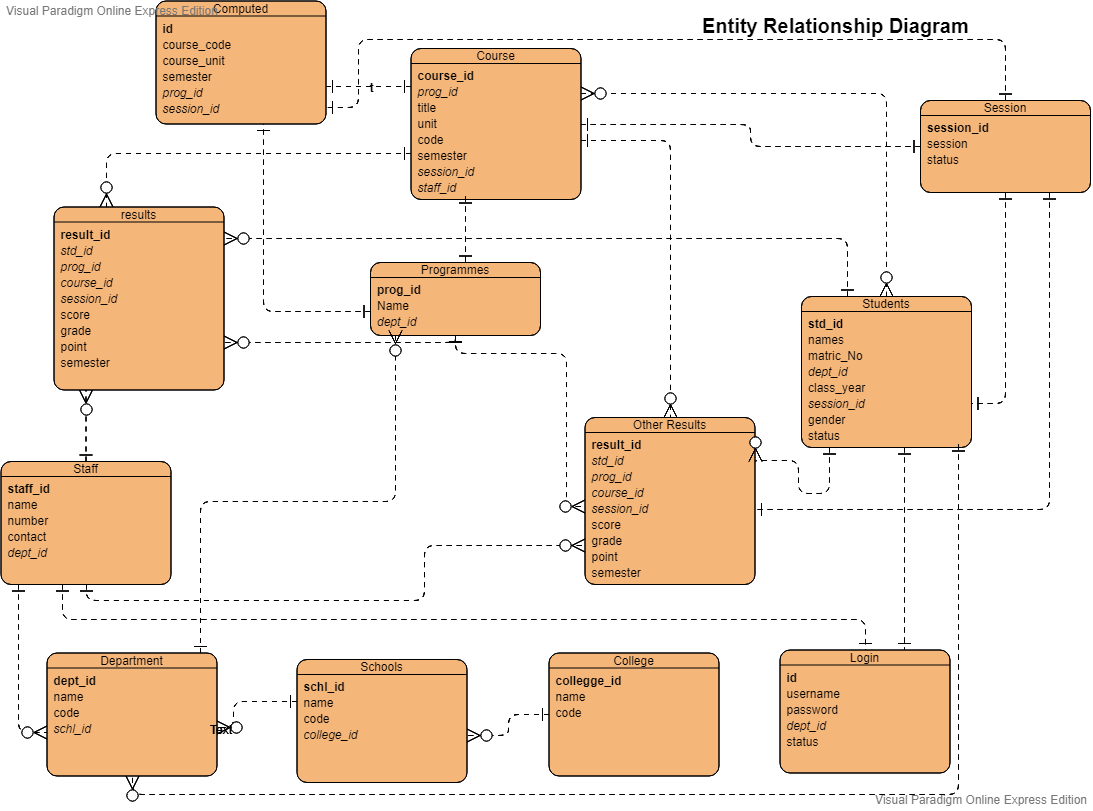
Moreover, the Nasa TLX questionnaire (see appendix IV) was used. The Nasa TLX was not targeted to vegans, as it was only used to measure the effort that the users put during the tasks that they had to complete.

* A description of the functionality and interfaces of the completed system.
* Appraisal: A critical appraisal of the project indicating the rationale for design/implementation decisions, lessons learnt during the course of the project and an evaluation (with hindsight) of the final product and the process of its production (including a review of the plan and any deviations from it). The project should be placed in a wider context and this could include the scientific, technical, commercial, social and ethical context.

In general, the web development of the application went fairly smoothly. It was a good decision to choose PHP and MySQL for the application because they were more flexible and easier to learn since I had already introduced to the basics, during the Database module for my course. Now, I feel more confident using them and I am ready for one step ahead. I also gained some knowledge of the security methods, ways to avoid shrewd hackers and ways to encrypt passwords. Moreover, I learnt how encryption is working and the significant importance of encrypting the passwords. Moreover, I learnt how to implement the APIs for Facebook and Google+ login. I was concentrated on the backend development because that was really important for me; to get experience in that field, so I can gain the skills to work as a backend web developer in the future. As a result, my design is over-simplified. Although I had positive comments from the evaluation, I would like to improve it by adding more features in the future. Furthermore, I learnt about Materialize CSS framework, which was very easy to learn. It has many functions, such as drop-down menu, side menu and cards which were used in the web app. These functions helped me design responsive nice-looking material in a short period of time. I should point out that my skills in time management have been improved a lot since I started my course and the use of the timetable plan helped me to complete most of the tasks on time. Also, I used two participants to help me with the testing because I needed someone else to test the application from the front end environment. That helped me with the responsiveness, as well as with the “debugging” my code, which is very important when the project is big. The general feedback from the evaluation of the final product was really positive and that made me more confident about my work.

In summary, this project has been very useful for me, and since it was my own idea, I am glad that I gained all these knowledge and experience.

Appendix



**Appendix: Use case documentation**

**UC1 Login to account**

Enable users to login into the application to access certain menus.

**Basic flow of events**

The use case begins when the system present user a login form.

|  |  |
| --- | --- |
| System: | Ensures that users enter username and password. **(A1)**  Verify user input. **(A2)** |
| User: | Enter username and password. **(A3)** |
| System: | Verifies the username and password entered is correct. **(A4)**  Determine the type of user and Display the user Dashboard and menus  Keep logs |

**Alternative flows**

**A1-Username and password field cannot be empty**

The user submit an empty field, both username and password field must be empty.

|  |  |
| --- | --- |
| System: | Prompt the user that username and password field cannot be empty |

**A2 - Unwanted character entered**

The system cannot allow unwanted character. User must have entered unwanted character.

|  |  |
| --- | --- |
| System: | Prompt the user that unwanted characters are not allowed. |

**A3 – Lock account**

System detect that login details are entered more than the required number of times

|  |  |
| --- | --- |
| System: | Lock account after N failed attempts |

**A4 – Incorrect username and password**

System verifies the login details to be incorrect

|  |  |
| --- | --- |
| System: | Display incorrect username and password. |

**UC2 Manage users**

Enable system manager (user) to add new user, view and edit user detail.

**Basic flow of events**

The use case begin when the user logged in successfully.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu options. e.g. “Manage user” menu |
| System manager: | Click on the manage user menu. |
| System: | Display the sub menu options for managing user, these are: Add, edit, and view user |
| System manager: | Selects one of the sub menu to Add user. **(A1, A2)** |
| System: | Present user registration form for user to input full name, username, number, select department, select user type with submit button. |
| System manager: | Enter name, username, number, select department, select user type and click on submit button. **(A3)** |
| System: | Ensure no field is left empty. **(A4)**  Verify user input. **(A5)**  Add the new user. **(A6)** |

**Alternative flows**

**A1 – Edit user**

**A2 – View user**

**A3 - Session Timeout**

The system detects a session timeout

System: Displays timed out message and/or re-directs to login page.

**A4 – Empty fields not allowed**

System: detect empty fields submitted

System: Prompt/Display all field are required, no field can be left empty

**A5 – Unwanted input (character)**

System detect that special character are inputted

System: Special Characters not allowed

**A6 – User registration successful**

System: prompt/display user registration successful

**UC3 Manage Students**

Enable system manager (user) to add, edit and view student records.

**Basic flow of events**

The use case begin when the user logged in successfully”.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu options. e.g. “Manage students” menu. |
| System manager: | Click on the manage students menu |
| System: | Display the sub menu option for managing student record, These are: Add, Edit, and view student. |
| System manager: | Select one of the sub menus to add new student. **(A1, A2)** |
| System: | Present student registration form for user to enter student name, number, department, gender, session with a submit button. |
| System manager: | Enter student name, number, department, session, gender then click on submit button. **(A3)** |
| System: | Ensure no field is left empty. **(A4)**  Verify user input. **(A5)**  Register and Create account for student**. (A6)** |

**Alternative flows**

**A1 – Edit student**

**A2 – View student**

**A3 - Session Timeout**

The system detects a session timeout

System: Displays timed out message and/or re-directs to login page.

**A4 – Empty fields not allowed**

System: detect empty fields submitted

System: Prompt/Display all field are required, no field can be left empty

**A5 – Unwanted input (character)**

System detect that special character are inputted

System: Special Characters not allowed

**A6 – student registration successful**

System: prompt/display students registration successful

**UC4 Manage courses**

Enable system manager (user) to add, edit and view Courses.

**Basic flow of events**

The use case begin when the user logged in successfully.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu options. e.g. “manage courses” menu. |
| System manager: | Click on the manage courses menu |
| System: | Display the sub menu option for managing courses, These are: Add, Edit, and view courses. |
| System manager: | Select one of the sub menus to add new course. **(A1, A2)** |
| System: | Present course registration form for user to enter course title, code, department, unit, semester, session with a submit button. |
| System manager: | Enter course title, code, department, unit, semester, session, then click on submit. **(A3)** |
| System: | Ensure no field is left empty. **(A4)**  Verify user input. **(A5)**  Register courses **(A6)** |

**Alternative flows**

**A1 – Edit courses**

**A2 – View courses**

**A3 - Session Timeout**

The system detects a session timeout

System: Displays timed out message and/or re-directs to login page.

**A4 – Empty field not allowed**

System: detect empty fields submitted

System: Prompt/Display all field are required, no field can be left empty

**A5 – Unwanted input (character)**

System detect that special character are inputted

System: Special Characters not allowed

**A6 – student registration successful**

System: prompt/display course registration successful

**UC5 manage backups**

Enable system manager (user) create backup and import backup

**Basic flow of events**

The use case begin when the user logged in successfully”.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu options. e.g. “manage backup” menu. |
| System manager : | Click on the “manage backup” menu |
| System: | Display the sub menu option for managing backup, these are create backup and import backup |
| System manager: | Select the sub menu option to create backup. **(A1)** |
| System: | Create Backup. **(A2)** |

**Alternative flow**

**A1 – Import Backup**

**A2 – Backup created**

System: prompt/Display backup created successfully.

**UC6 Manage results**

Enable coordinator (user) to input/import, edit, and view results.

**Basic flow of events**

The use case begin when the user logged in successfully.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu options. e.g. “manage result” menu. |
| Coordinator: | Click on the menu “manage result”. |
| System: | Display the submenu option to manage result, these are input/import result, Edit result, input CO, View result |
| Coordinator: | Select the option Import result. **(A1, A2, A3)** |
| System: | Present the form to for user to select programme, session and semester, with a submit button |
| Coordinator: | Select programme, session and semester, then click on submit. |
| System: | Ensure programme, session and semester is chosen. **(A4)**  Display form to select course code, select file to import with submit button |
| Coordinator: | Select course code, select file to import, then click on submit button. **(A5)** |
| System: | Ensure course code and file to import is chosen. **(A6, A7)**  Import records. **(A8)**  Keep logs |

**Alternative flows**

**A1 – Edit result**

**A2 – Input Carry - Over Result**

**A3 – View Results**

**A4 – No empty field is allowed**

System detect that empty files submitted

System: Prompt/display empty fields cannot be submitted.

**A5 - Session Timeout**

The system detects a session timeout

System: Displays timed out message and/or re-directs to login page.

**A6 – No empty field is allowed**

System detect that empty files submitted

System: Prompt/display empty fields cannot be submitted.

**A7 – file chosen must be in “.csv” format**

System detect file chosen is not “.csv” file

System: Display/prompt incorrect file format

**A8 – records imported**

**UC7 Import student scores**

Enable Teaching staff (user) to import students score

**Basic flow of events**

The use case begun when the user logged in successfully.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu options. e.g. menu “import score” |
| Teaching staff: | Click on the menu “import score”. |
| System: | Present a form for user to select course code and choose a file to import with submit button. |
| Teaching staff: | Select course code, choose a file then click on submit button. **(A1)** |
| System: | Ensure course code and file is chosen. **(A2)**  Import score. **(A3)**  Keep logs |

**Alternative flows**

**A1 - Session Timeout**

The system detects a session timeout

System: Displays timed out message and/or re-directs to login page.

**A2 – Empty fields not allowed**

System detect that empty files submitted

System: Prompt/display empty fields cannot be submitted.

**A3 – records imported**

System: Prompt/Display records imported

**UC8 Student profile**

Enable student (user) to View results

**Basic flow of events**

The use case begun when the user logged in successfully.

|  |  |
| --- | --- |
| System: | Display user dashboard and menu option. e.g. “View result” menu. |
| Student: | Select “view result” |
| System: | Display sub menu option for view result, these are Semester1, semester2, semester3, and semester4. |
| Student: | Select menu option “semester1”. **(A1, A2, A3)** |
| System: | Display the result for semester 1. |

**Alternative flows**

**A1 – Semester 2**

**A2 – Semester 3**

**A3 – Semester 4**

**UC9 Change password**

Enable User to change password

**Basic flow of events**

The use case begun when the user logged in successfully and user selects the option to change their password.

|  |  |
| --- | --- |
| System: | Present password change form and request user to enter their new password. |
| Teaching Staff | Enters their new password and username and clicks submit. **(A1, A2)** |
| System: | Validate the password to ensure it is correct. **(A3, A4, A5, A6)**  Displays confirmation to the user.  Keep logs |

**Alternative Flows**

**A1 - Cancel**

User decides to cancel the process of changing password

System: Returns to dashboard.

**A2 - Session Timeout**

The system detects a session timeout

System: Displays timed out message and/or re-directs to login page.

**A3 - Password Incorrect**

The system detect incorrect login details

System: display/prompt incorrect login details

**A5 - Unwanted characters in the password**

System detect that special character are inputted

System: display/prompt special characters are not allowed

**A6 – minimum length exceeded**

System detect that the minimum length of character required is exceeded

System: validate password minimum length

1. Niger State Polytechnic, Zungeru, Student and Staff Handbook (2018/2019) [↑](#footnote-ref-1)
2. *UDEZE, C. L., UMOREN, P. U., OHERI, H. E., & ATTAH H. H. (2017) Automated Students' Results Management Information System (SRMIS), Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 4, ISSN: 2458-9403,* [↑](#footnote-ref-2)
3. *. Akpasam J. E., Simeon O., Afolayan J. J. (2017). Development of Students Result Management System: A case study of University of Uyo. Mathematical and Software Engineering, Vol. 3, No. 1, 26-42* [↑](#footnote-ref-3)
4. *. Ashwin M., Jugal P., Aditya M. (2018). Student Result Analysis System. International Research Journal of Engineering and Technology (IRJET), vol.5, e-ISSN: 2895 -0056* [↑](#footnote-ref-4)
5. Obasa, A.I., Eludire, A.A. and Isaac, M. (2011) The Architectural Design of an Integrated Virtual Classroom System. Research Journal of Information Technology, 3, 43-48. [↑](#footnote-ref-5)
6. Amar, E. and Mohini (2009). Single portal for integrated examination system, emerging technologies in e-governance, E-Governance, pages 287-293 [↑](#footnote-ref-6)
7. Ukem, E. O.-I. (2011). A Software Application For The Processing Of Students Results. Global Journal of Pure and Applied Sciences, Volume 17 No. 4 [↑](#footnote-ref-7)
8. Akinmosin, J. (2014). Automated Student Result Management System using Oracle Database, Forms and Reports, Journal of information Engineering and Application Vol.4, ISSN 2225-0506(online) [↑](#footnote-ref-8)
9. Bhatt J., Jain R., Kadge S. and Parate P. (2016). Result Generation System for CBGS Scheme in Educational Organization. International Journal of Research in Engineering and Technology (IJRET), Vol. 5, Issue 2 from http://www.ijret.org [↑](#footnote-ref-9)
10. ahttps://**en.wikipedia.org**/wiki/**Alan\_Cooper** [↑](#footnote-ref-10)
11. <http://www.agilemodeling.com/artifacts/personas.htm>, [Copyright](http://www.ambysoft.com/licensing.html) 2003-2018 [Scott W. Ambler](http://www.ambysoft.com/scottAmbler.html), [↑](#footnote-ref-11)
12. Infragistics “is a global software company, founded in 1989, that publishes user interface development tools and components for a range of developer applications, across all platforms. The company is also a provider of developer support, testing tools, and UI and User Experience training and consulting services”. [Wikipedia](https://en.wikipedia.org/wiki/Infragistics) [↑](#footnote-ref-12)
13. <https://flatworldsolutions.com/10-differences-agile-waterfall-methodology/>, ©2019 Flatworld Solutions Pvt. Ltd. All Rights Reserved. [↑](#footnote-ref-13)
14. .<https://www.informatica.com/services-and-training/glossary-of-terms/agile-methodology-definition.html#fbid=w-gCxvCHf2g>, © 2019 Informatica All Rights Reserved [↑](#footnote-ref-14)
15. Plinio Thomaz Aquino, Lucia Vilela, Leite Filgueiras. (2005). User Modeling with Personas, Proceedings of the 2005 Latin American conference on Human-computer interaction, Cuernavaca, Mexico, pp. 277-281 [↑](#footnote-ref-15)
16. Wikipedia, “Personas”, http://en.wikipedia.org/wiki/Personas [↑](#footnote-ref-16)
17. UML is a way of visualizing a software program using a collection of diagrams. [↑](#footnote-ref-17)
18. system is something being developed or operated, such as a web site [↑](#footnote-ref-18)
19. Use cases are a set of actions, services, and functions that the system needs to perform. [↑](#footnote-ref-19)
20. Actors are people or entities operating under defined roles within the system [↑](#footnote-ref-20)
21. [] <https://www.smartdraw.com/use-case-diagram/> ©1994-2019 SmartDraw, LLC [↑](#footnote-ref-21)
22. The term Misuse Case or mis-use case is derived from and is the inverse of [use case](https://en.wikipedia.org/wiki/Use_case). [↑](#footnote-ref-22)
23. https://www.techopedia.com/definition/13136/prototyping , Copyright © 2019 Techopedia Inc. [↑](#footnote-ref-23)
24. [*https://www.i-programmer.info/news/146/5149.html*](https://www.i-programmer.info/news/146/5149.html) Copyright © 2009-2019 i-programmer.info. By David Conard, November 2012 [↑](#footnote-ref-24)
25. Teorey, T.J., Lightstone, S.S., et al., (2009). Database Design: Know it all.1st ed. Burlington, MA: Morgan Kaufmann Publishers [↑](#footnote-ref-25)
26. WAMP, also known as WAMPserver is a free [localhost](https://en.wikipedia.org/wiki/Localhost) server stack comprising Apache, MySQL and PHP for Windows. It is ideal for learning, testing and developing websites without having to use a remote web server. [↑](#footnote-ref-26)
27. Story is a descriptions of discrete functionality, it used to describe user goals. [↑](#footnote-ref-27)
28. Scrum is a framework for developing and sustaining complex products, a collaborative…team that is delivering business value in a tightly coordinated fashion [↑](#footnote-ref-28)
29. Sprint is one time-boxed iteration of a continuous development cycle. Within a Sprint, planned amount of work has to be completed by the team and made ready for review. The term is mainly used in Scrum Agile methodology. [↑](#footnote-ref-29)
30. <https://www.smallbizgeek.co.uk/tools/wamp/> [↑](#footnote-ref-30)
31. <http://www.wampserver.com/en/>  [↑](#footnote-ref-31)
32. <https://www.w3trainingschool.com/php-advantages> [↑](#footnote-ref-32)
33. PHP is supported by most Operating Systems such as; Windows, UNIX, and Linux. [↑](#footnote-ref-33)
34. There are built-in module in PHP that helps it to connect easily with database, this makes it demanding in the field of web development to develop websites that are data driven. [↑](#footnote-ref-34)
35. To learn PHP no intensive manual or studying is required, it commands and function can easily be understood and can easily figure out what is does from the name. [↑](#footnote-ref-35)
36. <https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system> [↑](#footnote-ref-36)
37. <http://searchoracle.techtarget.com/definition/MySQL> [↑](#footnote-ref-37)
38. <https://www.w3.org/standards/webdesign/htmlcss> [↑](#footnote-ref-38)
39. Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones) [↑](#footnote-ref-39)
40. <https://www.npmjs.com/> [↑](#footnote-ref-40)
41. **Gulp** is a tool that helps you out with several tasks when it comes to web development. It's often used to do front end tasks [↑](#footnote-ref-41)
42. <https://www.bootstrapdash.com/product/purple-free-admin-template/> [↑](#footnote-ref-42)
43. <https://developers.google.com/chart/interactive/docs/> [↑](#footnote-ref-43)
44. <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/> by [KORBIN BROWN](https://www.howtogeek.com/author/korbinbrown/) UPDATED SEPTEMBER 6, 2017, 2:21PM EDT [↑](#footnote-ref-44)
45. <https://usabilitygeek.com/10-free-web-based-web-site-accessibility-evaluation-tools/> August 22, 2011 by [Justin Mifsud](https://usabilitygeek.com/author/usab8904/) UsabilityGeek [↑](#footnote-ref-45)
46. “Normalization entails organizing the columns (attributes) and tables (relations) of a database to ensure that their dependencies are properly enforced by database integrity constraints. It is accomplished by applying some formal rules either by a process of synthesis (creating a new database design) or decomposition (improving an existing database design)” [↑](#footnote-ref-46)
47. Test represent requirement that must be fulfilled by the code. If there is no requirement (i.e no test), there is no need for implementing anything, therefore no need for any code. [↑](#footnote-ref-47)
48. “Process of making changes to the code, such that it does not alter the external behaviour of the code yet improves its internal structure”. Martin Flower, “Refactoring”, 1999 [www.refactoring.com](http://www.refactoring.com) [↑](#footnote-ref-48)
49. Provos, Niels; Mazières, David; Talan Jason Sutton 2012 (1999). ["A Future-Adaptable Password Scheme"](http://www.usenix.org/events/usenix99/provos/provos_html/node1.html). *Proceedings of 1999 USENIX Annual Technical Conference*: 81–92. [↑](#footnote-ref-49)
50. https://www.w3schools.com/sql/sql\_injection.asp [↑](#footnote-ref-50)