CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

A web application or web app is a computer program that utilizes web browsers and web technology to perform tasks over the Internet.

Millions of businesses use the Internet as a cost-effective communications channel. It lets them exchange information with their target market and make fast, secure transactions. However, effective engagement is only possible when the business is able to capture and store all the necessary data, and have a means of processing this information and presenting the results to the user.

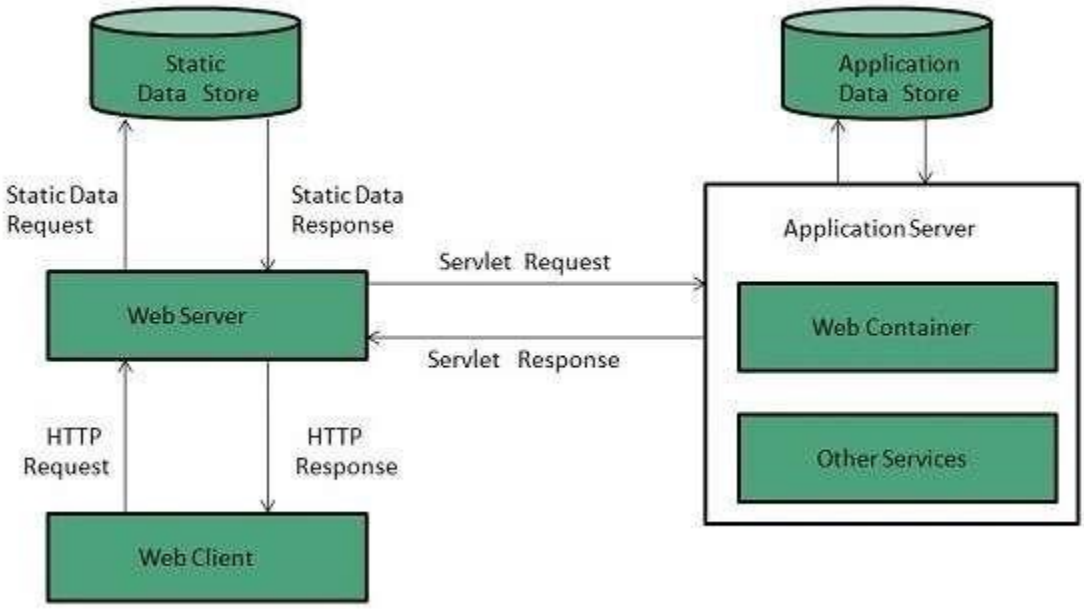

Figure 1: A web application overview (John Martin, 2017).

Web applications use a combination of server-side scripts (PHP and ASP) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users. This allows users to interact with the company using online forms, content management systems,

shopping carts and more. In addition, the applications allow employees to create documents, share information, collaborate on projects, and work on common documents regardless of location or device.

The architecture of a web application is depicted in figure 1 above.

A Portal is a private location on the internet, accessible with a unique URL (web address) and unique username and password unlike a website that is publicly accessible.

Its features are as follows:

i. Personal login is required
ii. Only portal members can see the contents
iii. Content is unique to user based on linked account information and group member settings/permissions
iv. Secure access point for personalized information
v. Communication features with other portal members or group
vi. Dynamic content changes more frequently than typical websites
vii. Interactive functionalities for portal site members

and many more. . .

An education portal is a specially designed website that provides a host of educational services. The term portal was historically used to describe a port or location of multiple loading and offloading activities. It is now used most widely to describe a multi-function website that includes public and private sections, data retrieval and submissions tools, personalized content, and often links or connections to education related systems or services.

The ability to provide personalized, user-specific content is central to the core functionality of an education portal. The technology required to support this type of website structure became widely available in the mid-1990s, with the advent of user management functionality as part of a web server.

There are two ways to provide this content: a series of template forms or the use of a content management solution.

A public website typically provides a range of general information about the school, programs, course, history, and admissions. Users are encouraged to refer to the website instead of contacting the school directly for general information. The content manager is responsible for reviewing and publishing content that is accurate, relevant, and easy to locate.

Registered students are provided with a unique user name and password to log into the private area of the website. Here, personalized content is provided, based on a database or similar tool that holds the student data. For example, students can log on to the education portal for their school and see their current courses, program of study, fees, and other relevant information. The depth and breadth of information provided varies widely and is often tied to the system architecture of the school.

Creating and maintaining an information portal is a new technological development and has resulted in the creation of new jobs and career paths. There are two areas of skill that are required to form an educational portal: computer systems and content management. This type of website requires a dedicated web server, that can support both public and private sections. There needs to be a mechanism or program to manage requests for user identifications and passwords, as well as the ability to remind users of their passwords or allow them to reset the password.

In addition to basic user management, the private section of the website usually has connection to either the student system or data from that system. This information requires additional security to protect student privacy and prevent unauthorized access. Specially trained security and infrastructure staff are required to set up and maintain these tools.

Content management refers to all the text, pictures, files, and data provided to users through the education portal. In many organizations, a site administrator logs into the website with his unique account information and has the ability to maintain or add content to the websites, both public and private. This role is critical to keeping the website functional and meeting user's needs.

Crawford University does not host its own education portal and needs to implement to the advantage of the institution. The existing portal http://mycrawfordportal.net has been sufficient, proficient and successful in its course registration by students and approval by various units, result processing, information management and dissemination, resumption clearance and examination docket/pass, account management.

Just as a program will have its flaws, also this has its own shortcomings that eventually makes it incompetent, such as performance, as it is very inefficient, because it is written in plain contemporary PHP language and/or very high traffic. It also has other numerous imperfections such as defective user interface and poor user experience, unimplemented functionalities, persistent and offensive dialog on page load, broken, duplicate and redundant links. The system being outsourced is one of the major challenge of the system as it doesn't afford availability at all times. User cannot get a new password online but have to go to the Edu-portal office so they can manually set a new password to the account.

This project promises to provide a perfect solution to all the stated problems.

1.2 STATEMENT OF THE PROBLEM

The purpose of this study is to develop a web-based Education Portal Application as a solution to the automation problems of students processes in the institution. The purpose of this study is aimed to solve the following problems:

i. **Inefficiency**: Slow page load and response time.

ii. **Under implemented:** Persistent and offensive dialog on page load that threatens satisfactory user interaction, broken links, duplicate and redundant links, unimplemented functionality, use of anachronistic technologies and defective user interface and poor user experience, no online help. User has to go to the Edu-portal office to manually set a new password for his/her account.

iii. Unreliable and insecure as server breaks down and students are unable to access the portal and it has no active SSL (Secure Sockets Layer) certificate i.e. it doesn't use Https (Hypertext Transfer Protocol Secure) protocol.

1.3 AIM AND OBJECTIVES OF THE STUDY

The aim of this study is to proffer solution to the above stated problems being faced and to develop a web-based education portal for Crawford University using codeigniter framework to develop each web page and MYSQL as the database for storage of information on the website. The objectives are to

i. Create an embedded online course portal which provides a pre-assessment module that allow courses verification to the students when login on the website and result checking by students. In addition, the course portal can also automatically evaluate and showcase the report of courses of the students.
ii. Create course allocation, approval, course grading and result processing by lecturers.
iii. Allow position-based user restriction, bursary approval and dispensation of clearance to qualified users, mailing, admission list management, user data management.
iv. Improve the privacy, user-friendliness and to enable convenient access to the different kinds of information and services mounted on the web by users, it would be desirable to set up a portal for channeling the vast information resources to the different users in an efficient and effective manner.

1.4 SIGNIFICANCE OF THE STUDY

The study will be beneficial to the following:

i. **To the Students**

Through this system, the students will be able to use the education portal effectively for their online registration process especially for registering their personal information, courses registration, resumption registration and exam registration. Example: It will also promote students easy access to information resources, materials etc.

ii. **To the Academic officers**

Academic officers include the lecturers, deans, Heads of departments, level advisers and Edu-portal officers. Through this system, they can manage academic affairs more conveniently and proficiently and also provide all the necessary information and resource material to their students and also be able to get the student information from the website.

iii. **To the Future Researchers**

The future researchers could gain knowledge from the study on the benefits, advantages and disadvantages, impact of developing web portals which they may apply to their research in the future. By improving on the portal in such a way that is being connected with inter-switch whereby students will be able to make any necessary payment through the website, payment like school fee, acceptance fee, and departmental fee and so on.

iv. **Project Supervisors**

This will serve as a tool for the easy management of student accounts and review of project work.

1.5 SCOPE AND LIMITATION OF THE STUDY

The application of this software may not be limited to Crawford University alone but can be used equally by other universities, even secondary schools and colleges. This paper only covers all that is required to design and implement the education portal program and nothing more. Usually, every work has some limitations and this study is not exempted. These are the following things that limit the education portal application development:

i. **Time limitation:** A researcher cannot boast a present accuracy of the research work due to difficulties ranging from other academic work, health condition and other distraction during the period of the research work, because in the work there is limited amount of time even from the research to be carried out.

ii. **Single or too few developers:** This system requires more developers than just me.

1.6 DEFINITION OF TERMS

The following are terms that the researcher used for better understanding of the concepts in the study. The terms are defined operationally, as they are used in the study.

i. **PROGRAM:** This is a set of instruction which the computer can initiate specified or defined to produce a designed objective.

ii. **DESIGN:** it is a detail plan or arrangement to achieve a particular purpose.

iii. **IMPLEMENTATION:** is the carrying out execution or practice of a plan.

iv. **RESEARCHER**: A researcher is someone who conducts research, i.e. an organized and systematic investigation into something. Scientists are often described as researchers.

v. **DEPARTMENT:** A department is a division of a university or school faculty devoted to a particular academic discipline.

vi. **ANACHRONISTIC:** You say that something is anachronistic when you think that it is out of date or old-fashioned.

vii. **SYSTEM:** a set of computer equipment and programs used together for a particular purpose.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

As the number of users on the World Wide Web increases every day, its use in different areas is also growing. One of the most powerful aspects of the Web is that anybody who has Internet access can browse on the net. This enables sharing the information worldwide. In the early days of the Internet, the World Wide Web consisted only of web sites. These were essentially information repositories containing static documents. Web browsers were invented as a means of retrieving and displaying those documents (Jashon, 2013).

Earlier in client-server computing, each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as static documents. A web page can merely display static content and it also lets the user navigate through the content, but a web application provides a more interactive experience.

The history of web application development is quite newsworthy and uncommon. Developers had to find the most radical and intensive solutions to the existing problems. It was considerable to make web apps work on different operating systems fluently. The earliest computing models were inconvenient. Every app had its precompiled client program and it had to be separately installed on every user's PC. Furthermore, the components of client and server were tightly bound to the definite operating system and computer architecture. As a result, it was expensive to port apps to other systems. If you recall the Web of its earliest days, you will say that the client received a web page as a static document. It was difficult to have interactive experience when you worked with such a page. When you introduced any changes to

the web page, you required time to refresh this page inasmuch as conducted a round trip back to its server (Nathan, 2016).

Web applications have been around since before the World Wide Web gained mainstream popularity. For example, Larry Wall developed Perl, a popular server-side scripting language, in 1987. That was seven years before the internet really started gaining popularity outside of academic and technology circles. The first mainstream web applications were relatively simple, but the late 90s saw a push toward more complex web applications. Nowadays, millions of Americans use web application to file their income taxes online, perform online banking tasks, stay in touch with friends and loved ones and so much more (Jashon, 2013).

2.2 WEB APPLICATION

A web application (or "web app" for short) is any computer program that performs a specific function by using a web browser as its client. The application can be as simple as a message board or a contact form on a website or as complex as a word processor or a multi-player mobile gaming app that you download to your phone. It is an application program that is stored on a remote server and delivered over the Internet through a browser interface (Daniel Nations, 2019).

A web application is an application utilizing web and [web] browser technologies to accomplish one or more tasks over a network, typically through a [web] browser. (Jarel Remick, 2011). Jarel Remick a renowned freelance designer, web developer and author breaks this further:

Application [Software]: "*Application software, also known as an application or an 'app', is computer software designed to help the user to perform singular or multiple related specific tasks*."

i. **Web technologies**: Flash, Silverlight, JavaScript, HTML & CSS, Java, various other programming languages and other computer technology intended for the use across networks.
ii. **Network**: The internet or intranet.

iii. **Browser**: Firefox, Chrome, Safari, Opera, Internet Explorer and various others.

Simply put, Web Applications are dynamic web sites combined with server side programming which provide functionalities such as interacting with users, connecting to back-end databases, and generating results to browsers (Remick, 2011).

Examples of Web Applications Are Online Banking, Social Networking, Online Reservations, ecommerce / Shopping Cart Applications, Interactive Games, Online Training, Online Polls, Blogs, Online Forums, Content Management Systems, etc... (Bernard Kohan, 2017).

Web application development is the creation of application programs that reside on remote servers and are delivered to the user's device over the Internet. A web application (web app) does not need to be downloaded and is instead accessed through a network. An end user can access a web application through a web browser such as Google Chrome, Safari, or Mozilla Firefox. A majority of web applications can be written in JavaScript, Cascading Style Sheets (CSS), and HTML5 (Margarete Rouse, 2019).

Web application development will typically have a short development life-cycle led by a small development team. Front-end development for web applications is accomplished through client-side programming. Client refers to a computer application such as a web browser. Client-side programming will typically utilize HTML, CSS and JavaScript. HTML programming will instruct a browser on how to display the on-screen content of web pages, while CSS keeps displayed information in the correct format. JavaScript will run JavaScript code on a web page, making some of the content interactive. (Margarete Rouse, 2019b).

Server-side programming powers the client-side programming and is used to create the scripts that web applications use. Scripts can be written in multiple scripting languages such as Ruby, Java Python. Server-side scripting will create a custom interface for the end-user and will hide the source code that

makes up the interface. A database such as MySQL or MongoDB can be used to store data in web application development.

2.2.1 WWW

The World Wide Web (WWW) is combination of all resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP). A broader definition comes from the World Wide Web Consortium (W3C), *"The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge."*

The Web, as it's commonly known, is often confused with the internet. Although the two are intricately connected, they are different things. The internet is, as its name implies, a network - a vast, global network that incorporates a multitude of lesser networks. As such, the internet consists of supporting infrastructure and other technologies. In contrast, the Web is a communications model that, through HTTP, enables the exchange of information over the internet. (Margaret Rouse, 2017).

It is all the Web pages, pictures, videos and other online content that can be accessed via a Web browser. The Internet, in contrast, is the underlying network connection that allows us to send email and access the World Wide Web. The early Web was a collection of text-based sites hosted by organizations that were technically gifted enough to set up a Web server and learn HTML. It has continued to evolve since the original design, and it now includes interactive (social) media and user-generated content that requires little to no technical skills.

We owe the free Web to Berners-Lee and CERN’s decision to give away one of the greatest inventions of the century (Techopedia, 2015).

2.2.2 INTERNET

The Internet, sometimes called simply "the Net," is a worldwide system of computer networks - a network of networks in which users at any one computer can, if they have permission, get information from any

other computer (and sometimes talk directly to users at other computers). It was conceived by the Advanced Research Projects Agency (ARPA) of the U.S. government in 1969 and was first known as the Arpanet. The original aim was to create a network that would allow users of a research computer at one university to "talk to" research computers at other universities. A side benefit of Arpanet's design was that, because messages could be routed or rerouted in more than one direction, the network could continue to function even if parts of it were destroyed in the event of a military attack or other disaster(Margarete Rouse, 2019a).

Today, the Internet is a public, cooperative and self-sustaining facility accessible to hundreds of millions of people worldwide. Physically, the Internet uses a portion of the total resources of the currently existing public telecommunication networks. Technically, what distinguishes the Internet is its use of a set of protocols called TCP/IP (for Transmission Control Protocol/Internet Protocol). Two recent adaptations of Internet technology, the intranet and the extranet, also make use of the TCP/IP protocol.

For most Internet users, electronic mail (email) practically replaced the postal service for short written transactions. People communicate over the Internet in a number of other ways including Internet Relay Chat (IRC), Internet telephony, instant messaging, video chat or social media.

The most widely used part of the Internet is the World Wide Web (often abbreviated "WWW" or called "the Web"). Its outstanding feature is hypertext, a method of instant cross-referencing. In most Web sites, certain words or phrases appear in text of a different color than the rest; often this text is also underlined. When you select one of these words or phrases, you will be transferred to the site or page that is relevant to this word or phrase. Sometimes there are buttons, images, or portions of images that are "clickable." If you move the pointer over a spot on a Web site and the pointer changes into a hand, this indicates that you can click and be transferred to another site.

Using the Web, you have access to billions of pages of information. Web browsing is done with a Web browser, the most popular of which are Chrome, Firefox and Internet Explorer. The appearance of a particular Web site may vary slightly depending on the browser you use. Also, later versions of a particular browser are able to render more "bells and whistles" such as animation, virtual reality, sound, and music files, than earlier versions.

The Internet has continued to grow and evolve over the years of its existence. IPv6, for example, was designed to anticipate enormous future expansion in the number of available IP addresses. In a related development, the Internet of Things (IoT) is the burgeoning environment in which almost any entity or object can be provided with a unique identifier and the ability to transfer data automatically over the Internet (Margaret Rouse, 2019).

2.2.3 CLIENT

The "client" is used in client-server environment to refer to the program the person uses to run the application. A client-server environment is one in which multiple computers share information such as entering information into a database. The "client" is the application used to enter the information, and the 'server' is the application used to store the information. (Elise Moreau, 2019).

A client is the receiving end of a service or the requestor of a service in a client/server model type of system. The client is most often located on another system or computer, which can be accessed via a network. This term was first used for devices that could not run their own programs, and were connected to remote computers that could via a network. These were called dumb terminals and they were served by time-sharing mainframe computers. (Techopedia, 2019)

Techopedia explains a client further "*A client can be a simple application or a whole system that accesses services being provided by a server. A client can connect to a server through different means like domain*

sockets, named, shared memory or through Internet protocols, which is the most common method being used since the wide adoption of the Internet".

Clients are classified into three types:

i. **Thin Client**: A client application with minimum functions that uses the resources provided by a host computer and its job is usually just to display results processed by a server. It simply relies on a server to do most or all of its processing.

ii. **Thick/Fat Client**: This is the opposite of the thin client. It can do most of its processing and does not necessarily rely on a central server, but may need to connect to one for some information, uploading, or to update data or the program itself. Anti-virus software belongs to this category because they do not really need to connect to a server to do their job, although they must connect periodically to download new virus definitions and upload data.

iii. **Hybrid**: Exhibits characteristics from the two above types. It can do most processes on its own but may rely on a server for critical data or for storage.

2.2.4 BENEFITS OF WEB APPLICATIONS

A web application relieves the developer of the responsibility of building a client for a specific type of computer or a specific operating system, so anyone can use the application along as they have internet access. Since the client runs in a web browser, the user could be using an IBM-compatible or a Mac. They can be running Windows XP or Windows Vista. They can even be using Internet Explorer or Firefox, though some applications require a specific Web browser.

Web applications commonly use a combination of server-side script (ASP, PHP, etc.) and client-side script (HTML, JavaScript, etc.) to develop the application. The client-side script deals with the presentation of the information while the server-side script deals with all the hard stuff like storing and retrieving the information. (Elise Moreau & Daniel Nations, 2019)

2.2.5 EXISTENCE OF WEB APPLICATIONS

Web applications have been around since before the World Wide Web gained mainstream popularity. For example, Larry Wall developed Perl, a popular server-side scripting language, in 1987. That was seven years before the internet really started gaining popularity outside of academic and technology circles.

The first mainstream web applications were relatively simple, but the late 90s saw a push toward more complex web applications. Nowadays, millions of Americans use web application to file their income taxes online, perform online banking tasks, stay in touch with friends and loved ones and so much more. (Elise Moreau & Daniel Nations, 2019)

2.2.6 WEB APPLICATION EVOLUTION

Most web applications are based on the client-server architecture where the client enters information while the server stores and retrieves information. Internet mail is an example of this, with companies like Google's Gmail and Microsoft's Outlook offering web-based email clients.

Over the past several years, there's been a big push for web applications to be developed for functions that do not normally need a server to store the information. Your word processor, for example, stores documents on your computer, and doesn't need a server.

Web applications can provide the same functionality and gain the benefit of working across multiple platforms. For example, a web application can act as a word processor, storing information in the cloud and allowing you to 'download' the document onto your personal hard drive.

If you've been using the web long enough to witness how popular web applications like Gmail or Yahoo mail clients have changed over the years, you have seen how sophisticated web applications

have become. Much of that sophistication is because of AJAX, which is a programming model for creating more responsive web applications.

GSuite (formerly Google Apps), Microsoft Office 365 are other examples of the newest generation of web applications. Mobile applications that connect to the internet (such as your Facebook app, your Dropbox app or your online banking app) are also examples of how web applications have been designed for the ever increasingly popular use of the mobile web (Marcus & Dafydd, 2010).

2.2.7 WORKING PRINCIPLE OF A WEB APPLICATION

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some of the applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server. The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information. Application server technology ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.

Here's what a typical web application flow looks like: User triggers a request to the web server over the Internet, either through a web browser or the application's user interface. Web server forwards this request to the appropriate web application server

Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data. Web application server sends results to the web server with the requested information or processed data. Web server responds back to the client with the requested information that then appears on the user's display (Robert Gibb, 2016).

2.3 EDUCATIONAL PORTAL

Portals build on the same technology used for Web sites, but enhance the functionality and flexibility to cater for the demands of specific classes of user. According to Gerd Waloszek, "Portals are a special

breed of external or internal Websites offering a blend of information, applications and services. This implies that portals always have more than just information to offer, as many Websites do" (Gerd Waloszek, 2015).

According to Rob Allan, "Put simply a portal is a presentation layer which aggregates, integrates, personalizes and presents information, transactions and applications to the user according to their role and preferences" (Rob Allan, 2018).

From both perceptions portal can be concluded as a gateway to web access which users can locate all the web content they commonly need which required personalization, search, channels, links, and customization base on role and workflow.

According to Ramona Winkler "Traditionally, a portal denotes a gate, a door, or entrance. In the context of the World Wide Web, it is the next logical step in the evolution to a digital culture". Formerly, portals are defined as search engines where it offers a full text index of document content. Today's Internet portals offer a more structured, navigable interface compares to Internet search engines. Internet portal is more focused on better delivery of specific information among a group with the same interest. (Ramona Winkler, 2008).

According to Robert Moskowitz of Matrix: The Magazine for Leaders in Education, when you hear the word "portal" you might immediately think of one of the many commercial Websites, such as Yahoo or Excite, that populate the Internet today. As the term implies, these services are the gateway to the Internet for many people, offering news, search functions, sports scores, shopping and movie reviews. (Robert Moskowitz, 2015).

An education portal, is the entry point for a college or university. It provides a centralized source of information and services for students, prospective students, faculty, suppliers, administrators, alumni and friends. The term education portal can mean anything from a relatively simple set of Web-based

application and payment services, to a comprehensive online interface offering highly personalized and customizable access to nearly all the features and benefits of campus life and work.

Education portals can provide entry points to anything from e-shopping for books and campus regalia, to Web access, student activity information, class resources, syllabus and assignment listings, and homework submission, as well as online class registration and tuition payment.

Shivani, a computer engineer and management post graduate, with writing as passion expresses that an education portal is a specially designed website that provides a host of educational services. The term portal was historically used to describe a port or location of multiple loading and offloading activities. It is now used most widely to describe a multi-function website that includes public and private sections, data retrieval and submissions tools, personalized content, and often links or connections to education related systems or services. One of the best education portal in India is Leading Education Portal | School & Higher Education in India which provides plethora of information related to educational sector and over its various aspects such as educational news, study abroad etc. (shivani, 2017).

Online education portal is a forum for you to get all your education related needs catered to. You can search for and apply for different degrees and online courses. This portal features an extensive list of thousands of educational institutes for potential students.

Education portals were pioneered by UCLA in 1999, to be followed by similar systems at the University of Washington and the University of Buffalo. Advancements in software simplify the work involved in building comprehensive portals from scratch. As a result, most schools can launch a pilot portal within 12 months. Expansion and roll out takes longer, however, and a continually evolving portal is an unending process.(Rick, n.d.).

Only about 5 percent of universities currently have campus portals, but many other are on the way. Some 80 percent of US. colleges with enrollments of more than 1,000 will have registration portals by 2010,

predicts the Connecticut-based Gartner Group (Gartner, 2010) "They will become as essential to the campus experience as the quad," predicts Matthew Pittinsky, chairman of Blackboard Inc., the Washington, DC.-based vendor of enterprise software for about 1,400 universities and schools. (Gartner & Pittinsky, 2010).

"A portal supports staff, alumni, students, faculty and others," says Blackboard's Pittinsky." It's the common interface for all the services each of these constituencies cares about on campus." (Pittinsky, 2010).

"Portals are on everyone's mind right now, both within the university and externally," agrees Dirk Herr-Hoyman, project manager at the University of Wisconsin's Department of Information Technology, in Madison, Wisc. "In not too many years, every university will have a personalized portal system." - (Herr-Hoyman, 2017).

Herr-Hoyman continues to explain further: "The benefits will be in keeping and attracting world-class students and faculty," he says. "That's important to us as a university. We're not cutting staff or expenses; we're trying to do more with the budgets we already have by providing better services." (Herr-Hoyman, 2017).

2.4 A REVIEW OF AOCOED SCHOOL PORTAL

AOCOED (Adeniran Ogunsanya College of Education Otto, Ijanikin, Lagos) student portal was created for students only, used by the ICT department and students only.

Students will visit the ICT center to meet the portal officers to do what they want. The student portal provided basic information related to their academic records, result, course registration, profile updating. The portal supports mostly payment, via remita.

Software technologies used

i. HTML5
ii. Javascript – jquery, bootstrap, icheck
iii. Css – bootstrap, font awesome, AdminLTE
iv. ASPX

Strength

Support for payment gateways like remita. Students can just pay online and get a reference.

Weakesses

It's not smart

i. courses are registered one at a time. For students that register at least 14 courses, the submit button will be clicked 14 times and page reload 14 times to register the courses one by one. The portal speed is affected by the traffic weight on their server. When there is too much traffic which always does, the portal becomes unreasonably slow and therefore service delivery becomes poor.
ii. It uses old technologies and methodologies
iii. To recover or reset a password, the ICT portal officers have to put a call through to the remote development company of the portal, supplying the relevant information of the student.

2.5 A REVIEW OF UNILAG PORTAL

UNILAG application portal integrates application for any of its offered degree in a dropdown box. Also UNILAG application integrates payment and by default it has course registration features and result processing, student profile management using sub-domains. For example, http://results.unilag.edu.ng and http://moodle.unilag.edu.ng.

UNILAG also has a parent's corner where a parent can view all information pertaining to a ward Just by inputing the child's matriculation number.

Weakness

Old and mediocre interface

2.6 A REVIEW OF CRAWFORD UNIVERSITY WEB PORTAL

The portal was made to assist Crawford staffs and students in providing basic information related to their academic records, registration, clearances, assessments, result processing and uploading, approval of course forms and bursary and other managements. Likewise, this will also give privilege to the students to submit their requests and transactions via Internet.

Enrolled students will be given accounts, usernames as the string "CRW" prefixed or suffixed to their matriculation number and passwords – default "crawford", to access and login to their account Portal. Students can then access this portal via Internet wherein they can conveniently inquire information or do basic student transactions anywhere and anytime they want.

This system basically includes the following:

i. **My Profile** – This is where students could view their personal and contact information. Students can also update and print their bio data at any time.
ii. **Student Academic Information** – This is where students could view their academic records such as grades, class schedules and current assessments.
iii. **Staff section** – The portal covers from Deans to HODs, course advisers, level advisers, lecturers and bursary unit with their respective activities, privileges and dashboards.
iv. **Admin section** - The administrative section.

v. **Other Services** – Other services the portal offers are course registration by students and approval by various units, result processing, information management and dissemination, resumption clearance and examination docket/pass, account management.

Certainly, Crawford portal will be enhanced and additional services will be incorporated in the future. Nevertheless, the student Portal definitely raised the bar of the College in rendering quality services for its students.

Software technologies used

i. HTML5
ii. Javascript – jquery, velocity, velocity-ui, thickbox.js, bootstrap javascript
iii. Css – bootstrap, font awesome, thickbox
iv. PHP

Strength

Sufficient, proficient and successful in its course registration by students and approval by various units, result processing, information management and dissemination, resumption clearance and examination docket/pass, account management.

Weaknesses

Just as a program will have its flaws, also this has its own shortcomings that eventually makes it incompetent, such as performance, as it is very inefficient, because it is written in plain contemporary PHP language and/or very high traffic. It also has other numerous imperfections such as defective user interface and poor user experience, unimplemented functionalities, persistent and offensive dialog on page load, broken, duplicate and redundant links. The system being outsourced is one of the major challenge of the system as it doesn't afford availability at all times. User cannot get a new password online but have to go to the edu-portal office so they can manually set a new password to the account.

In conclusion, along with all proposed features, the portal in development will be usable by the security, bursary, registry units and parents including whistle blowing functionalities.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 INTRODUCTION

Understanding of what system analysis mean will be great importance before giving into detail discussion about analysis of existing system. In order to just computerize without solving the problem, information collected should be organized and analyzed because it is when the problems and potentials are clearly identified that the new system could be used to solve the problem.

3.2 APPLICATION ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

3.2.1 DATABASE SCHEMA

The database schema or database structure of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). Below is the overall schema of the database used for this project. The database is required to be named “crawfordportal” by the framework, and has tables named as follows - admin, colleges, courses, departments, fresh_clearance, lga(local government area), my_course_form, olevel, portal_pass, programmes, state and student_details.

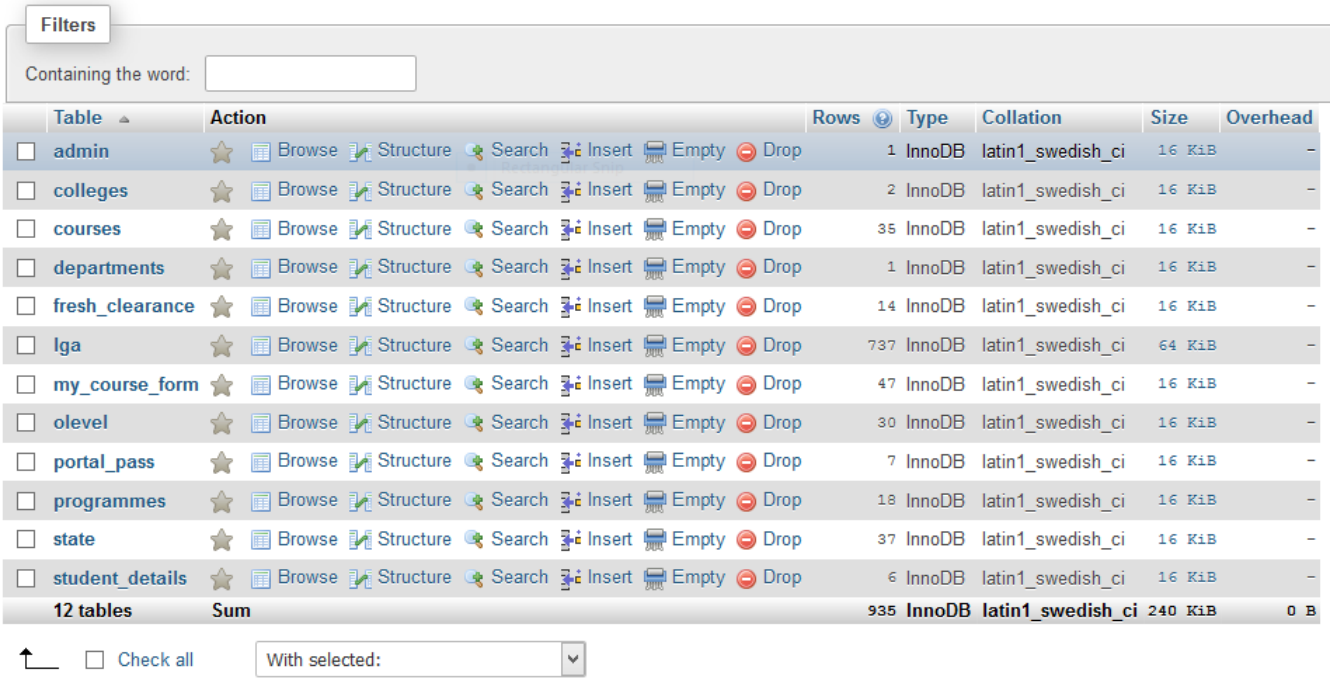

Figure 3.1 Database tables

Portal pass table: This table contains the authentication credentials for student portal login. It is joined with the student_details table at login to check for authentication data.

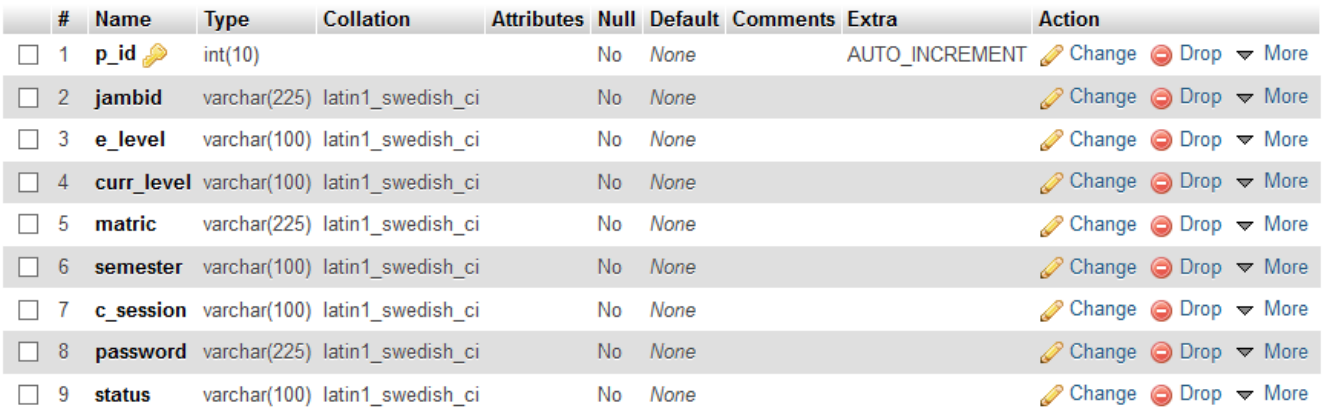

Figure 3.2 portal_pass table structure

Student details table: The student details table (Figure 3.3) below contains mostly student bio data.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	jamb_id	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	status_id	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	switch	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
5	lname	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
6	oname	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
7	gender	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	telephone	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
9	address	text	latin1_swedish_ci		No	None			Change Drop More
10	email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
11	religion	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
12	marital	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
13	nationality	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
14	state	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
15	lga	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
16	department	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
17	college	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
18	parent	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
19	ptelephone	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
20	paddress	text	latin1_swedish_ci		No	None			Change Drop More
21	nok	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
22	nok_tele	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
23	image	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
24	user	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Figure 3.3 student_details table structure

Programmes table: The programmes table contains information about Crawford university programmes. It is fetched when necessary for example when user is prompted to select the programme in a dropdown list.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	prog_id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	programme_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	college_id	varchar(10)	latin1_swedish_ci		No	None			Change Drop More

Figure 3.4 programmes table structure

Course form table: It contains data about course forms. The table structure below shows the fields of the course form table such as course level, course code, course title, course units, course status and so on.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	matric_no	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
3	course_level	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
4	c_semester	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
5	course_code	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
6	course_title	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
7	course_units	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
8	course_status	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
9	c_programme	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
10	c_department	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
11	c_college	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
12	total_unit_attempt	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
13	c_status	int(10)			No	None			Change Drop More

Figure 3.5 course form table structure

3.3 DESIGN PATTERNS

In software engineering, a software design pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is not a finished design that can be transformed directly into source or machine code. It is a description or template for how to solve a problem that can be used in many different situations. Design patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.

Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved. Patterns that imply mutable state may be unsuited for functional programming languages, some patterns can be rendered unnecessary in languages that have built-in support for solving the problem they are trying to solve, and object-oriented patterns are not necessarily suitable for non-object-oriented languages.

Design patterns may be viewed as a structured approach to computer programming intermediate between the levels of a programming paradigm and a concrete algorithm.

3.3.1 USES OF DESIGN PATTERNS

Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.

Often, people only understand how to apply certain software design techniques to certain problems. These techniques are difficult to apply to a broader range of problems. Design patterns provide general solutions, documented in a format that doesn't require specifics tied to a particular problem.

In addition, patterns allow developers to communicate using well-known, well understood names for software interactions. Common design patterns can be improved over time, making them more robust than ad-hoc designs.

3.3.2 TYPES OF DESIGN PATTERNS

i. **Creational patterns:** These design patterns provide a way to create objects while hiding the creation logic, rather than instantiating objects directly using new operator. This gives program more flexibility in deciding which objects need to be created for a given use case.

ii. **Structural patterns:** These design patterns concern class and object composition. Concept of inheritance is used to compose interfaces and define ways to compose objects to obtain new functionalities.

iii. **Behavioural patterns:** These design patterns are specifically concerned with communication between objects.

iv. **J2EE patterns:** These design patterns are specifically concerned with the presentation tier. These patterns are identified by Sun Java Center.

The design pattern applied for this project is MVC (Model View Controller). This pattern is used to separate application's concerns.

3.3.3 MVC – MODEL VIEW CONTROLLER

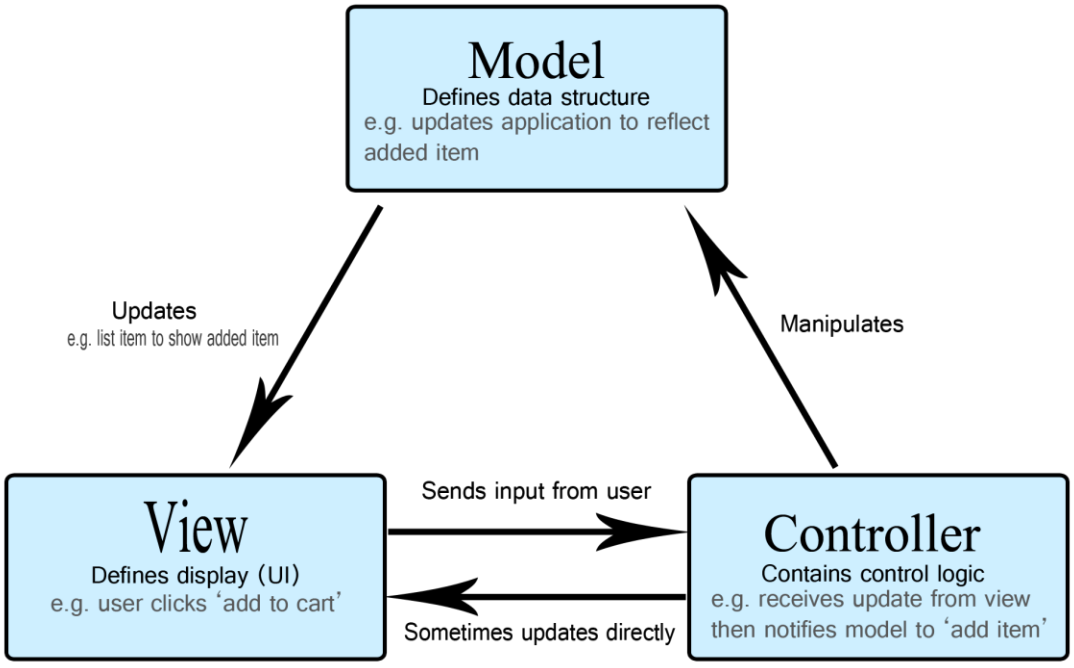

Figure 3.6 Model view controller

Model–View–Controller (usually known as MVC) as shown in figure 3.6 is an architectural design pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for code reuse and parallel development.

Traditionally used for desktop graphical user interfaces (GUIs), this architecture has become popular for designing web applications. Popular programming languages like JavaScript, Python, Ruby, PHP, Java, and C# have MVC frameworks that are used in web application development straight out of the box.

It is an application design model comprised of three interconnected parts. They include the model (data), the view (user interface), and the controller (processes that handle input).

The MVC model or "pattern" is commonly used for developing modern user interfaces. It provides the fundamental pieces for designing a programs for desktop or mobile, as well as web applications. It works well with object-oriented programming, since the different models, views, and controllers can be treated as objects and reused within an application.

Components

i. Model

The central component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic and rules of the application. A model is data used by a program. This may be a database, file, or a simple object, such as an icon or a character in a video game.

ii. View

Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. A view

is the means of displaying objects within an application. Examples include displaying a window or buttons or text within a window. It includes anything that the user can see.

iii. Controller

Accepts input and converts it to commands for the model or view. A controller updates both models and views. It accepts input and performs the corresponding update. For example, a controller can update a model by changing the attributes of a character in a video game. It may modify the view by displaying the updated character in the game.

In addition to dividing the application into these components, the model–view–controller design defines the interactions between them.

i. The model is responsible for managing the data of the application. It receives user input from the controller.
ii. The view means presentation of the model in a particular format.
iii. The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

As with other software patterns, MVC expresses the "core of the solution" to a problem while allowing it to be adapted for each system. Particular MVC architectures can vary significantly from the traditional description here. The MVC framework used for this project as introduced in the topic is CODEIGNITER. Other examples of MVC frameworks are Laravel, kendo, angular.js, backbone.js, react.js and many more.

3.3.4 MODEL CLASSES

The model classes defined in this application are

i. **Admin_Model:** Manages data control for administrator. It contains methods relevant to the administrator activities.

ii. **Courses_Model:** Data related to Courses such as course title, course unit, semester, course level course id and many more.

iii. **Get_Model:** The get model is responsible for generally getting data from any table. For example, it was used to get local government area based on selected state. It was also used to fetch states, colleges and department.

iv. **Portal_Model:** Contains methods relevant to the portal login and dashboard activities. Such as methods that gets course form, validates login credentials, hashes passwords with encryption keys using sha1 algorithm and logout.

v. **Student_Model:** It manages data pertaining to student. It fetches from student_details table.

3.3.5 VIEWS

The views are just simple php and/or html files that are included by the codeigniter framework at specified location. Every separate entity on a page can be separated in different view files. View inclusion is mostly done in the controller files using statement

```
$this->load->view (view_file, data_array);
```

Where parameter view_file is a file relative to the views folder

3.3.6 CONTROLLER CLASSES

The controllers used for this application are

i. **Academic_Login:** Controls the academic login section.

ii. **Clearance:** Controls the resumption clearance section.

iii. **Home:** This is the controller that loads the default view, that the portal welcome page.

iv. **My_Portal:** This controller controls the student portal dashboard activities like password recovery, fetching course form, matric number update and log out.

v. **Student_Portal:** This controller loads the student portal dashboard default view, that is the dashboard home or dashboard root view.

vi. **Undergraduate_pg:** This controls the undergraduate or post graduate account activation and bio data processing.

And the admin controllers are

i. Courses

ii. **Dashboard**: The admin dashboard.

iii. **Login**: The admin login.

iv. **Update_Semester**: This controller toggles the current semester between harmathan and rain semester.

3.4 CODEIGNITER FRAMEWORK

According to its documentation at https://www.codeigniter.com/, *"CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications."*

For building a web application you spend a lot of time in writing the same code again and again. Frameworks provide you a starting block and minimize the amount of code needed to build a website.

CodeIgniter is PHP driven framework but it's not a PHP substitute. Diving into CodeIgniter doesn't mean you are leaving PHP behind. PHP is a server-side scripting language for building dynamic web-based applications.

CodeIgniter contains libraries, simple interface and logical structure to access these libraries, plug-ins, helpers and some other resources which solve the complex functions of PHP more easily maintaining a high performance. It simplifies the PHP code and brings out a fully interactive, dynamic website at a much shorter time. It supports PHP version of 5.2.6 or newer and MySQL version 4.1 or newer. It makes your web more robust and your code easier to read and maintain. It is a free toolkit, light weight and easier to install.

A person using CodeIgniter must be familiar with PHP. You need to have a good knowledge about PHP like its basic syntax and how it interacts with database and HTML.

3.4.1 REASONS FOR THE USAGE OF CODEIGNITER

i. If you need a framework with small footprint - CodeIgniter 3 has a 2MB download, including the user guide.

ii. You need a high and exceptional performance - consistently outperforms most of its competitors.

iii. Need a framework which requires nearly zero configurations - Much of the CodeIgniter configuration is done by convention, for instance putting models in a "models" folder. There are still a number of configuration options available, through scripts in the "config" folder.

iv. Need a framework which don't use command line.

v. Need a framework which doesn't require adhering to restrictive coding rules.

vi. To get a simplified code structure.

vii. Simple solutions over complexity - CodeIgniter encourages MVC, but does not force it on you.

viii. Strong Security – It takes security seriously, with built-in protection against CSRF and XSS attacks.

ix. Clear documentation - The CodeIgniter User Guide comes with the download. It contains an introduction, tutorial, a number of "how to" guides, and then reference documentation for the components that make up the framework.

3.5 PROGRAMMING TOOLS

A programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs, that can be combined together to accomplish a task, much as one might use multiple hand tools to fix a physical object.

The most basic tools are a source code editor and a compiler or interpreter, which are used ubiquitously and continuously. Other tools are used more or less depending on the language, development methodology, and individual engineer, and are often used for a discrete task, like a debugger or profiler. Tools may be discrete programs, executed separately – often from the command line – or may be parts of a single large program, called an integrated development environment (IDE). In many cases, particularly for simpler use, simple ad hoc techniques are used instead of a tool, such as print debugging instead of using a debugger, manual timing (of overall program or section of code) instead of a profiler, or tracking bugs in a text file or spreadsheet instead of a bug tracking system.

Programming tools were initially designed to support or complement programming languages by providing the functionality and features these languages did not have. Typically, they are standalone utilities that provide or support a particular task within any phase of the development/programming cycle. For example, a debugger is a programming tool that helps programmers identify and resolve bugs within a program's source code. Compilers, linkers, assemblers, disassemblers, load testers, performance analysts, GUI development tools and code editors are also all forms of programming tools.

The following are programming tools used for this project

i. Server-side scripting language: PHP
ii. Markup language: HTML
iii. Stylesheet: CSS

iv. Programming Framework: codeigniter

v. Database: mysql

vi. Web server: Apache

vii. Code Editor: sublime text

viii. Browser(s): Mozilla Firefox

ix. Other development tools are LAMP (Linux Apache Mysql PHP) stack, browser developer tools.

3.5.1 PHP

PHP is a server side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php"(Morris, 2018).

3.5.2 SCRIPTING LANGUAGE

A script is a set of programming instructions that is interpreted at runtime. A scripting language is a language that interprets scripts at runtime. Scripts are usually embedded into other software environments.

The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application. Server side scripts are interpreted on the server while client side scripts are interpreted by the client application.

PHP is a server side script that is interpreted on the server while JavaScript is an example of a client side script that is interpreted by the client browser. Both PHP and JavaScript can be embedded into HTML pages (Techopedia, 2019).

3.5.3 PROGRAMMING LANGUAGE VS SCRIPTING LANGUAGE

Table 3.1 Differences between programming language and scripting language

PROGRAMMING LANGUAGE	SCRIPTING LANGUAGE
Has all the features needed to develop complete applications.	Mostly used for routine tasks
The code has to be compiled before it can be executed	The code is usually executed without compiling
Does not need to be embedded into other languages	Is usually embedded into other software environments.

3.5.4 HTML

First developed by Tim Berners-Lee in 1990, HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance) (Shannon, 2019).

3.5.5 CSS

Short for cascading style sheets, CSS is a language used to describe reusable styles for presenting documents written in a markup language. Its concept was originated by Håkon Wium Lie in 1994. In December 1996, CSS was made a specification by the W3C and today allows web developers to alter the layout and appearance of their web pages. For example, CSS may be used to change the font used in certain HTML element, as well as its size and color. A single CSS file may be linked to multiple pages, which allows a developer to change the appearance of all the pages at the same time (Computer Hope, 2019).

3.5.6 MYSQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.(Margarete Rouse, 2018).

3.5.7 MYSQL WORKING PRINCIPLE

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results (Margarete Rouse, 2018).

3.5.8 APACHE

Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all webservers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules. Most WordPress hosting providers use Apache as their web server software. However, WordPress can run on other web server software as well. (WpBeginner, 2019).

3.5.9 WEB SERVER

A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well. The web server used for this project is LAMP (LINUX APACHE MYSQL/MongoDB PHP/Python/Perl).

Web servers often come as part of a larger package of Internet- and intranet-related programs for serving email, downloading requests for File Transfer Protocol (FTP) files, and building and publishing Web pages. Considerations in choosing a Web server include how well it works with the operating system and other servers, its ability to handle server-side programming, security characteristics, and the particular publishing, search engine and site building tools that come with it (Allan, 2018).

3.6 MODULES OF THE DEVELOPED SYSTEM

The solution proffered was to develop an online portal that can perform the below:

i. Registration of students in the faculty of science
ii. Online course registration and approval processes
iii. Provision of payment details

3.6.1 VARIABLES USED AND ITS RELATIONSHIP

i. Variable used for registration

The variables needed for the processing of registration are: name, matric no, password, department, programme and school.

ii. Variable used for login page

The variables are matric no and password.

iii. Variable used for course registration page

The variables are matric no, password, level, course of study, course title, course code, unit and amount.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND DOCUMENTATION

4.1 MODULES DESCRIPTION AND IMPLEMENTATION

Below are the description and implementation of each frame (modules) as contained in the program:

Homepage: A homepage is the main or first page of a web site or set of hyperlinked documents. This page is set to open by default when the application starts up or when the home function is invoked. Also this page contains all links to other sections/pages of the application.

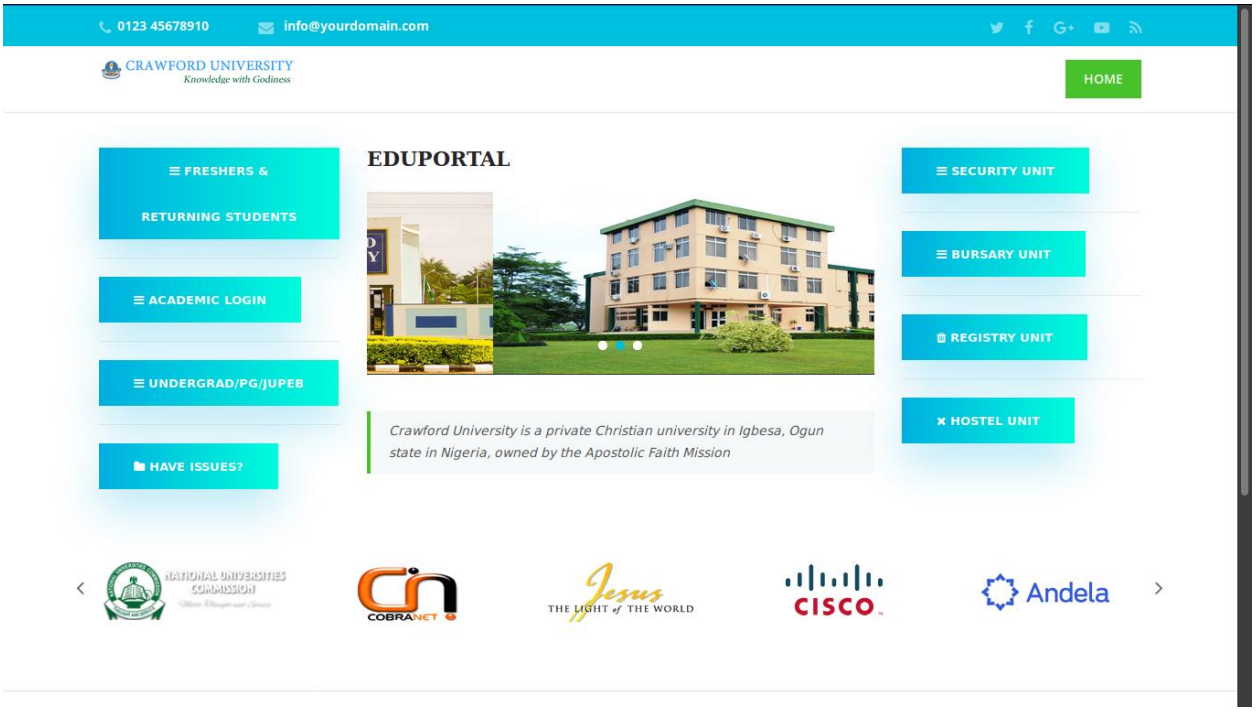

Figure 4.1 Portal home page view

Academic Login: The academic login is where all academic candidates including staff and student logs in to their respective accounts with their UID and password. The Unique Identification (UID) identifies what type of account is attempting to login.

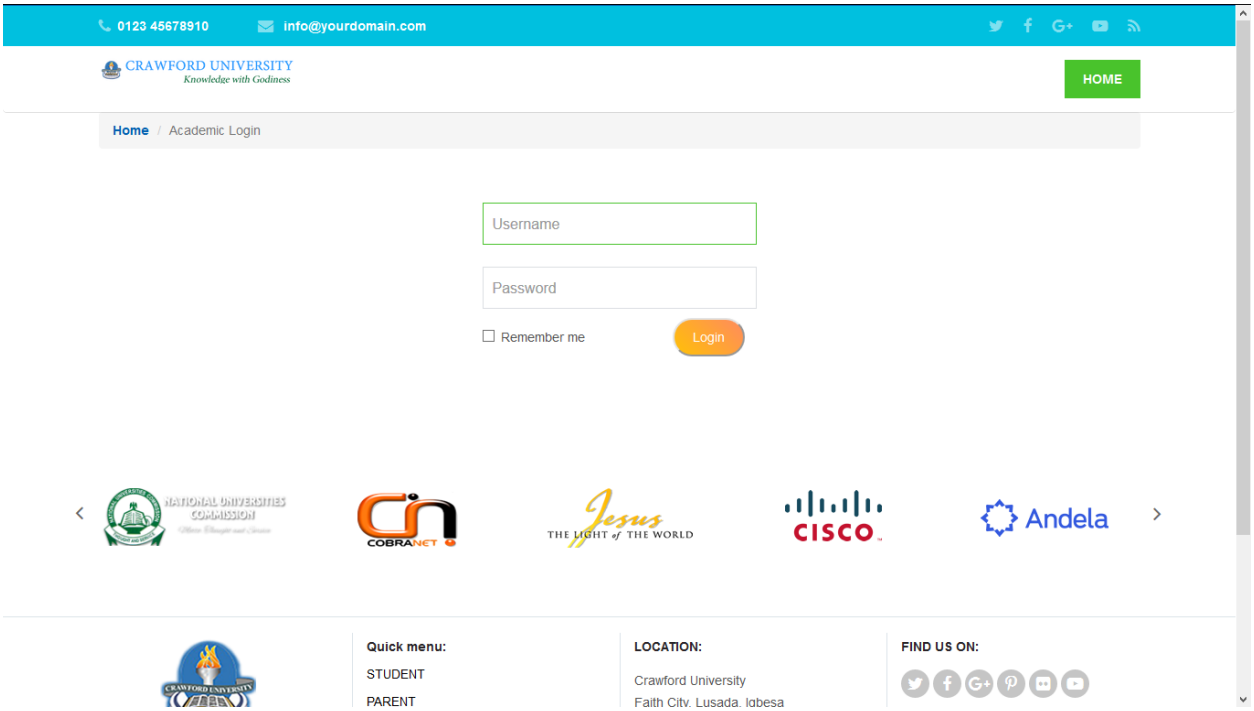

Figure 4.2 Academic login view

Resumption clearance: This implements how students process resumption clearance.

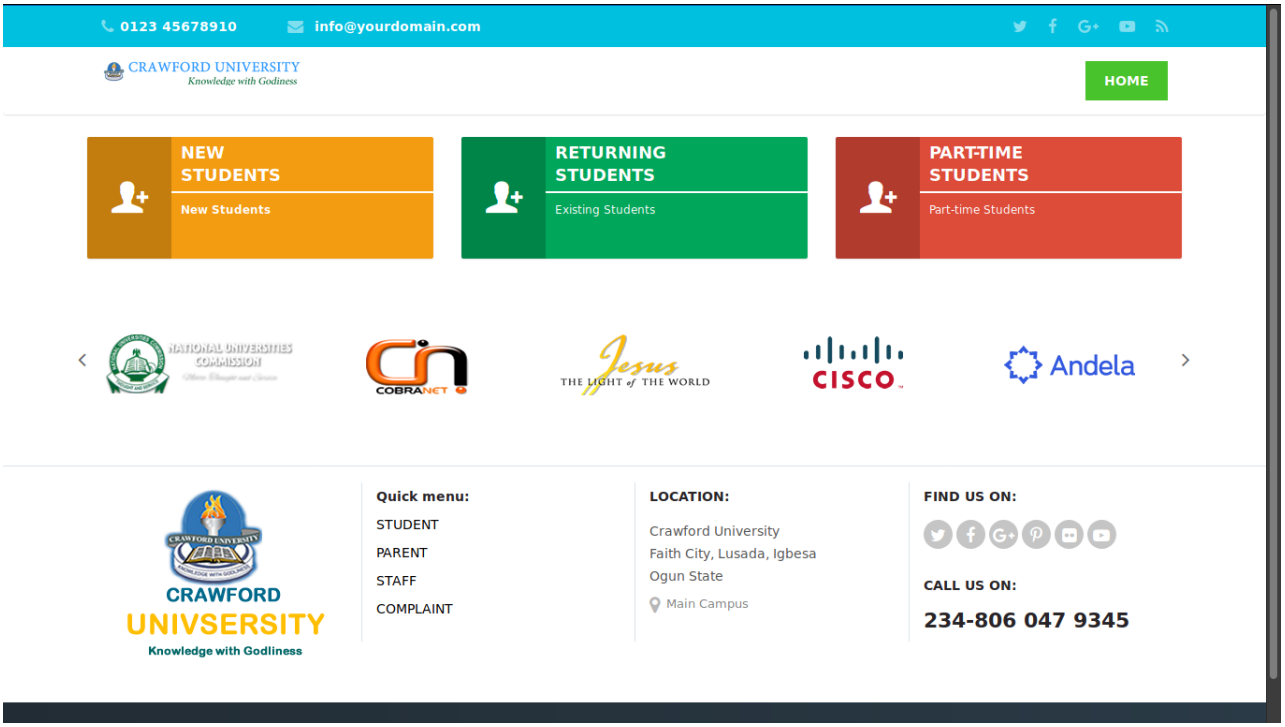

Figure 4.3 Resumption clearance view

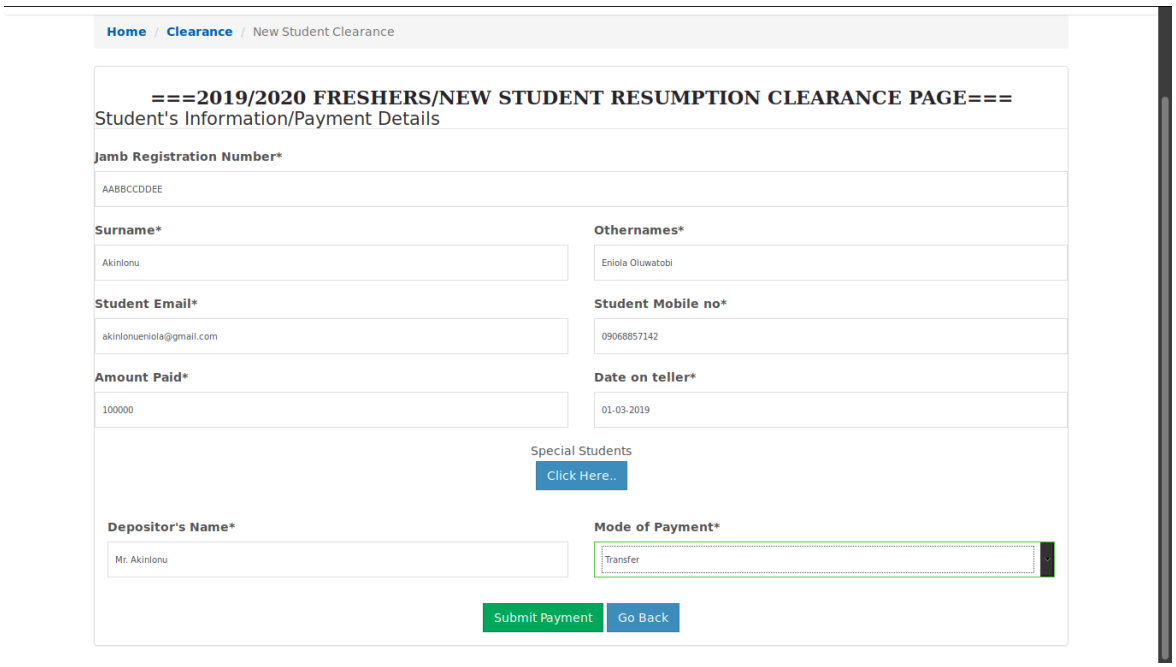

Figure 4.3.1 Fresher's resumption clearance view

Undergraduate account activation: After freshers have submitted their clearance and portal administrator has approved of the payment, they can now proceed to activate their portal account by inputing their JAMB registration number and fill in their bio data.

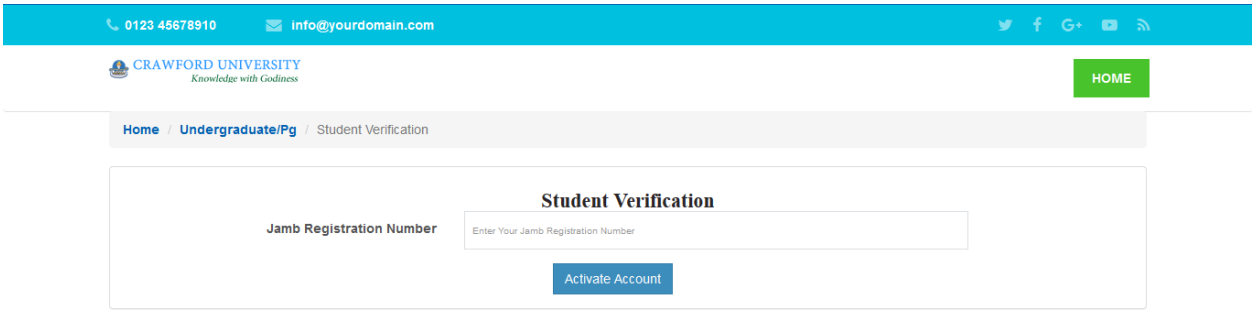

Figure 4.4 Undergraduate account activation view

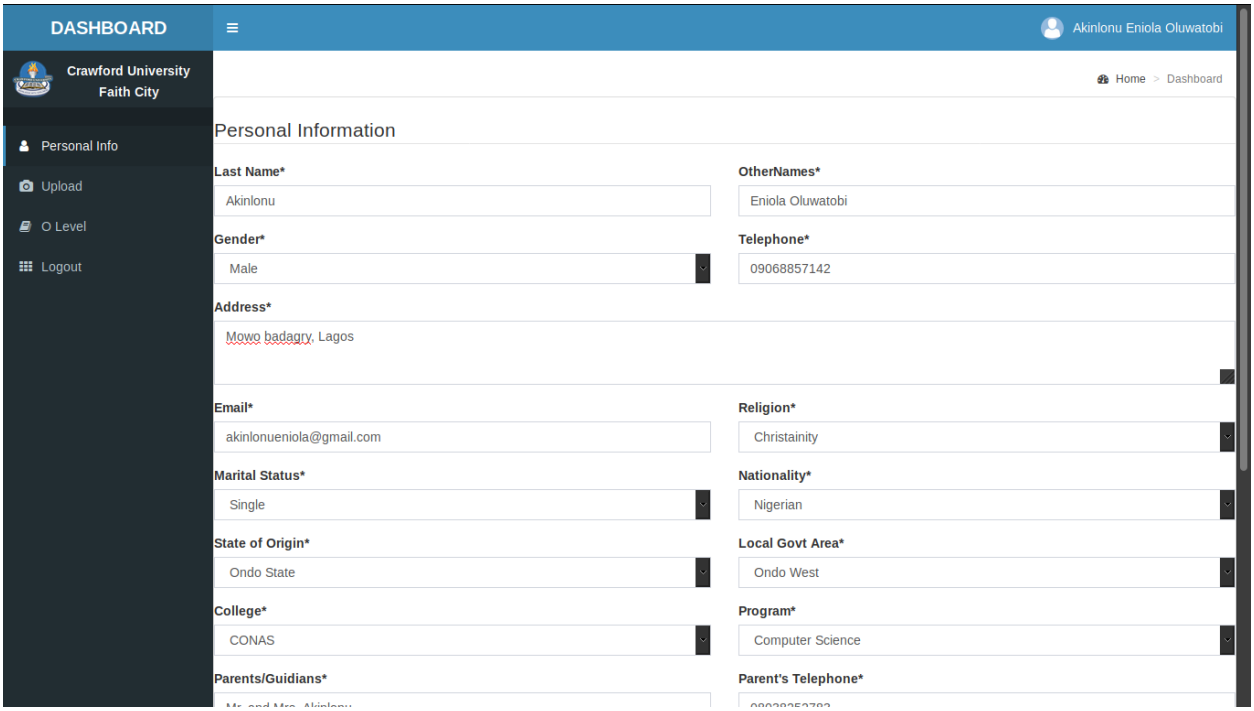

Figure 4.4.1 Bio-data registration view

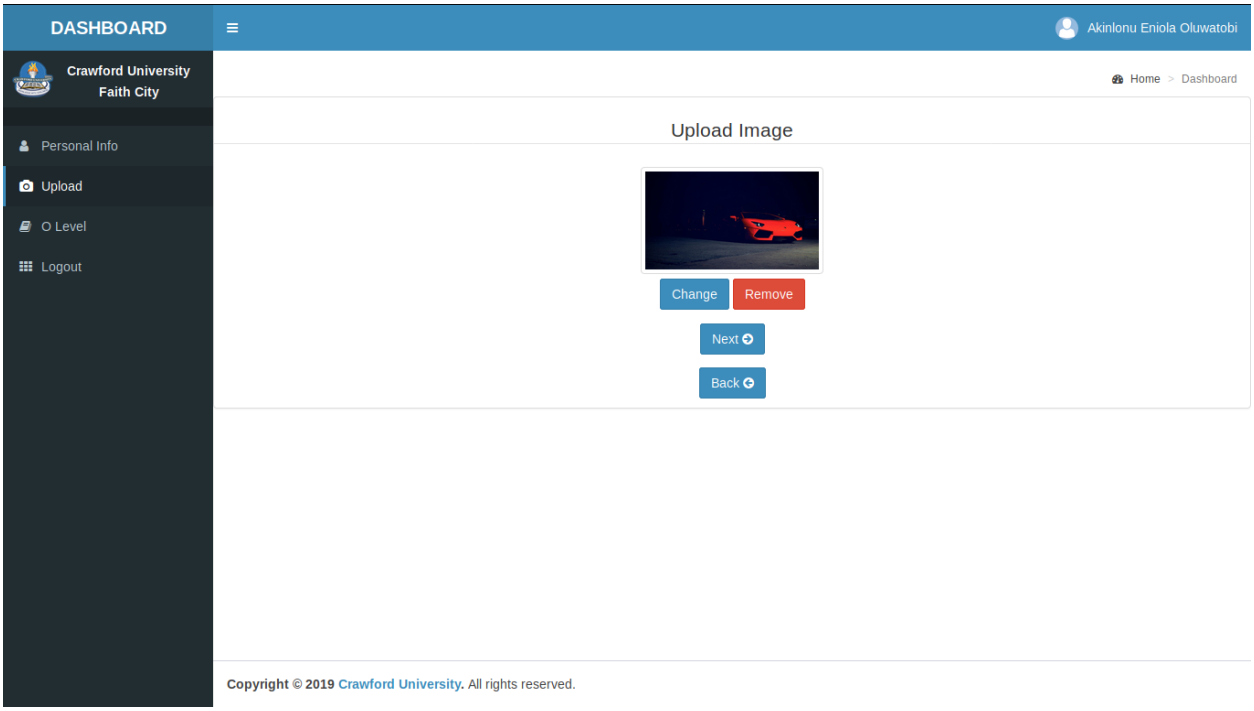

Figure 4.4.2 Profile picture upload view

Student dashboard: A dashboard is a type of graphical user interface which often provides at-a-glance views of key performance indicators (KPIs) relevant to a particular objective or business process. In other usage, "dashboard" is another name for progress report. Below is the student dashboard only accessible to logged in users.

Also this dashboard contains links to other activities of a logged in student user such as password update, matric number update, clearance, bio-data update, log out and so on.

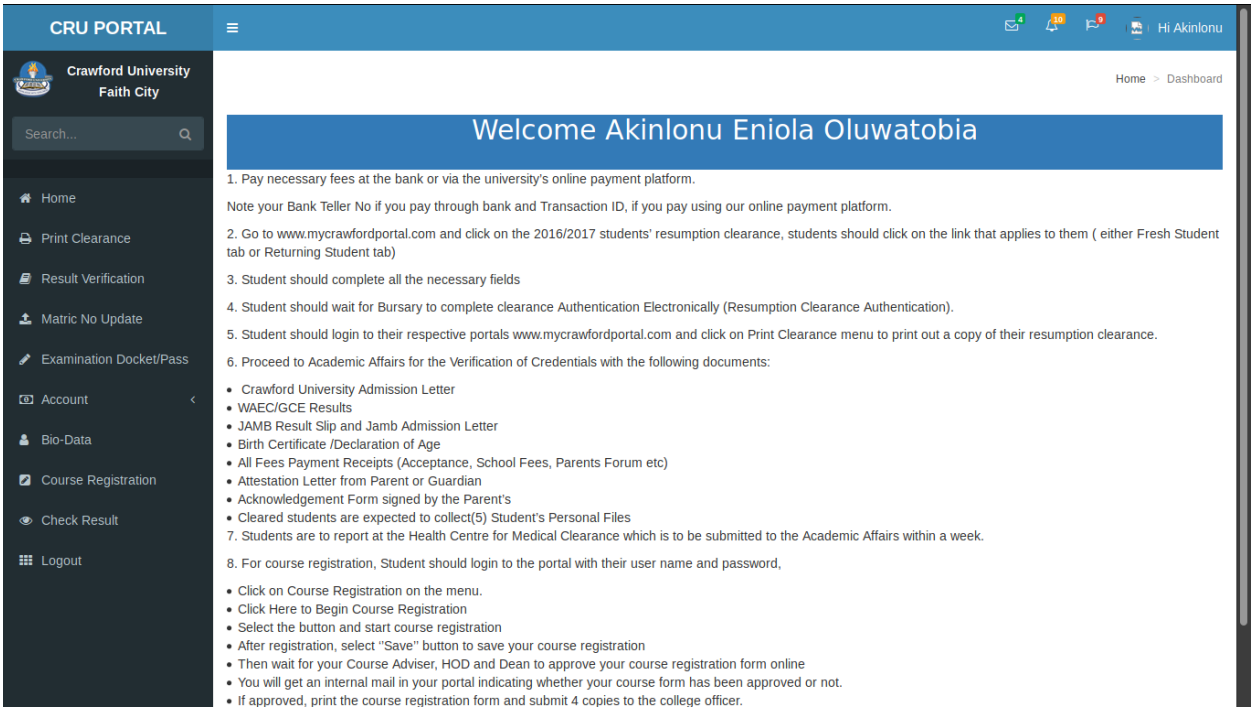

Figure 4.5 Logged in user dashboard view

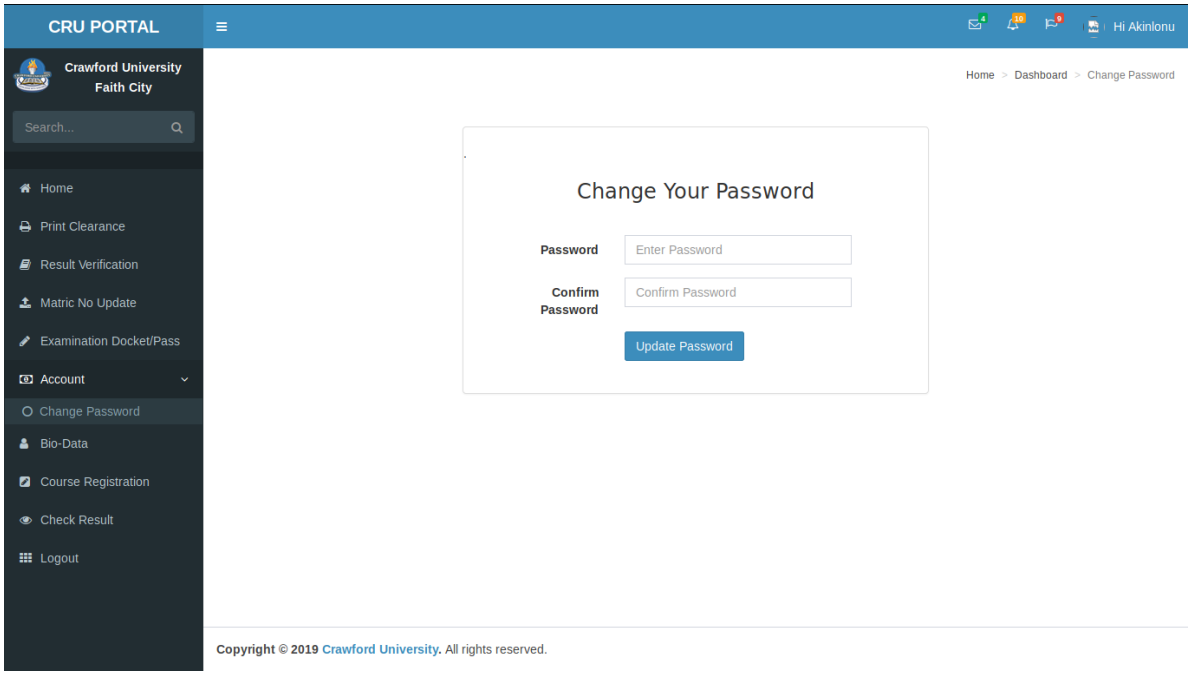

Figure 4.5.1 Password update view

ADMINISTRATOR VIEWS

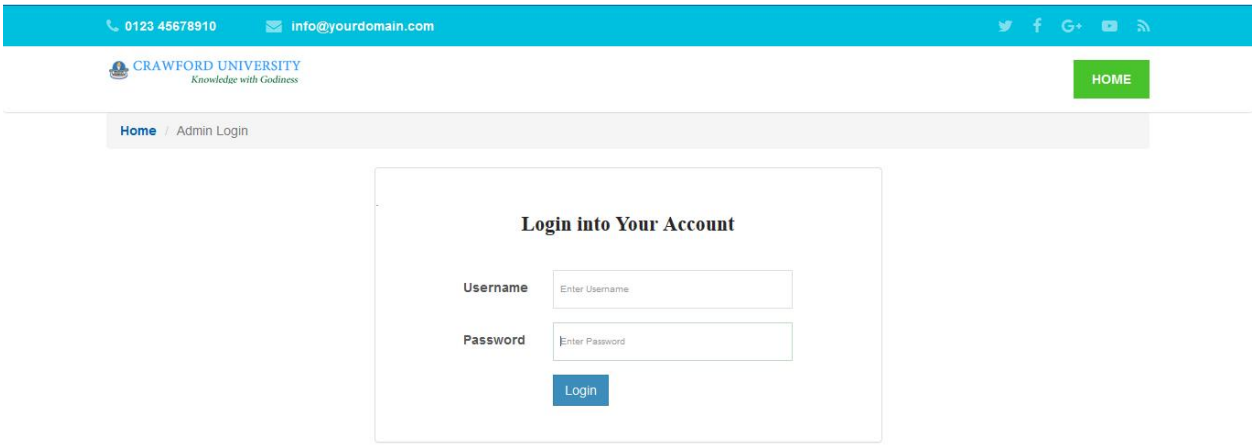

Figure 4.6 Administrator login view

Administrator dashboard: The administrator dashboard links to other activities of the administrator such as fresh student's approval, adding courses relative to departments and levels, adding programmes, updating semester, and matric number update.

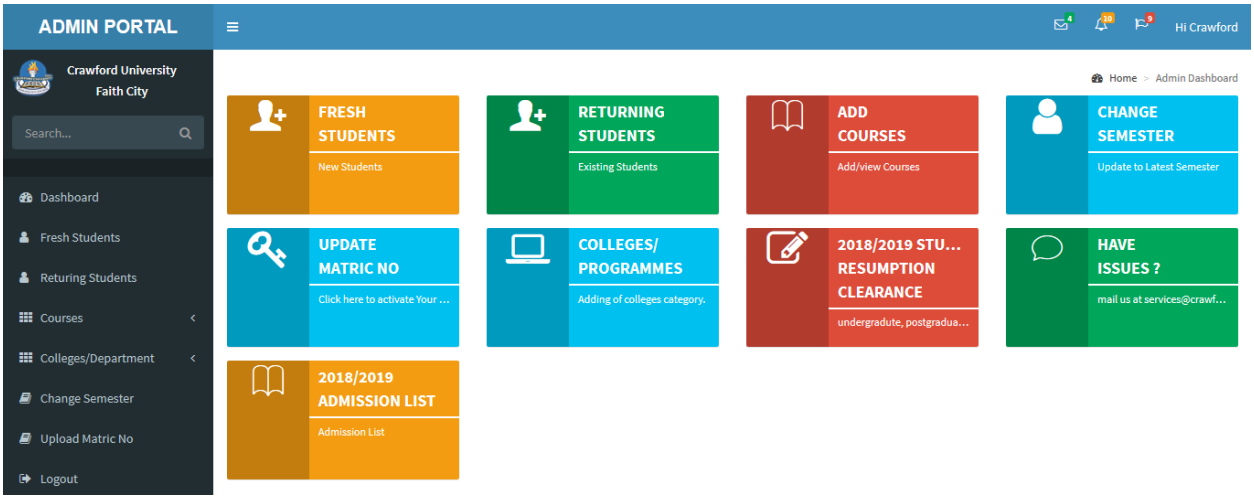

Figure 4.7 Administrator dashboard view

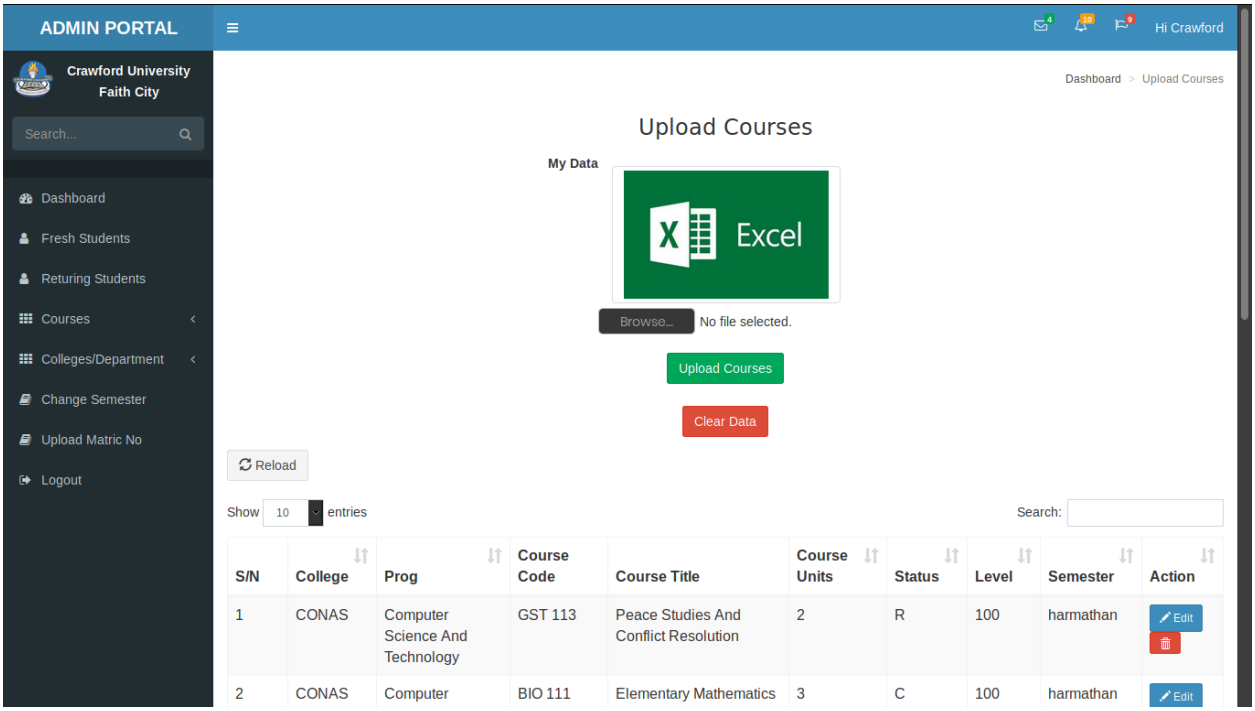

Figure 4.7.1 Admin upload courses view

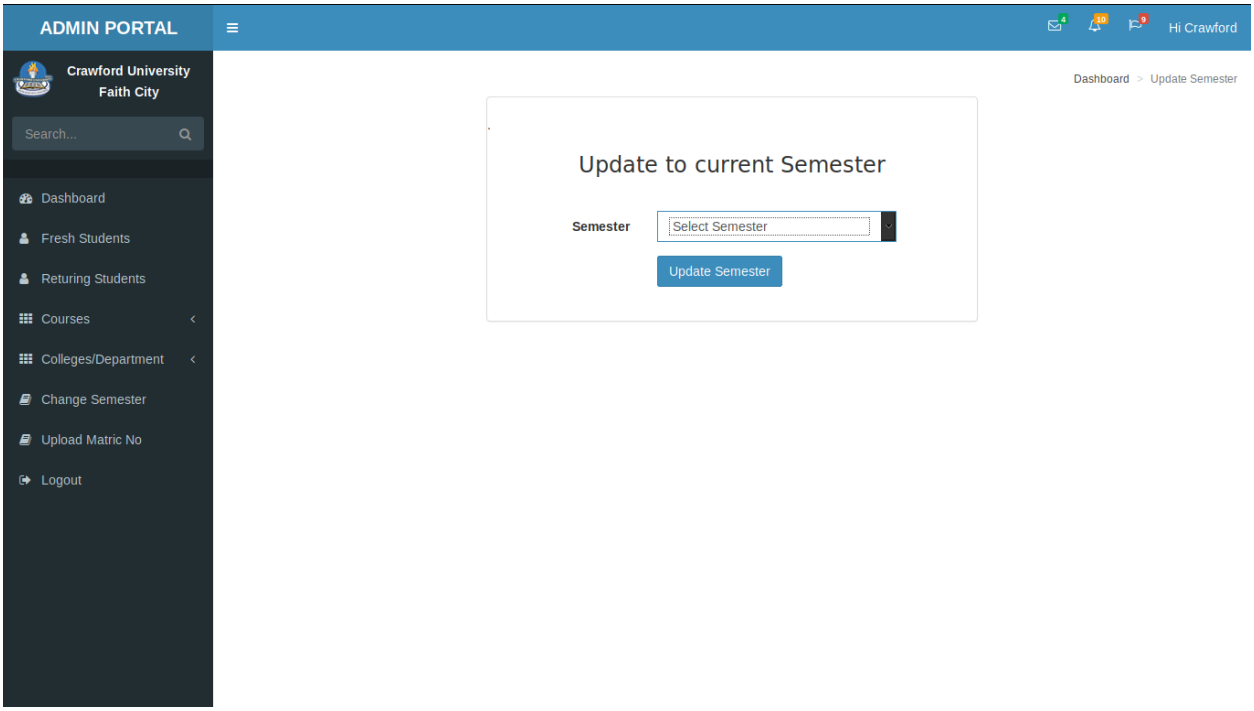

Figure 4.7.2 Admin update current semester view

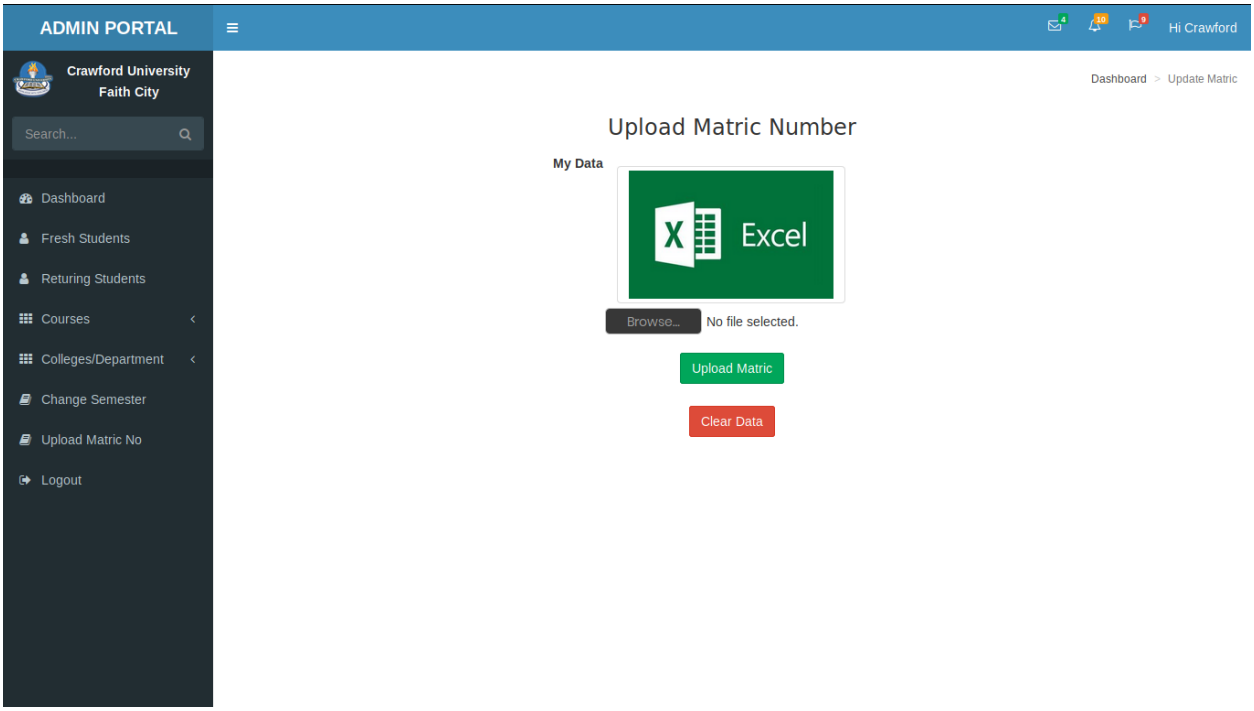

Figure 4.7.3 Admin upload matric number view

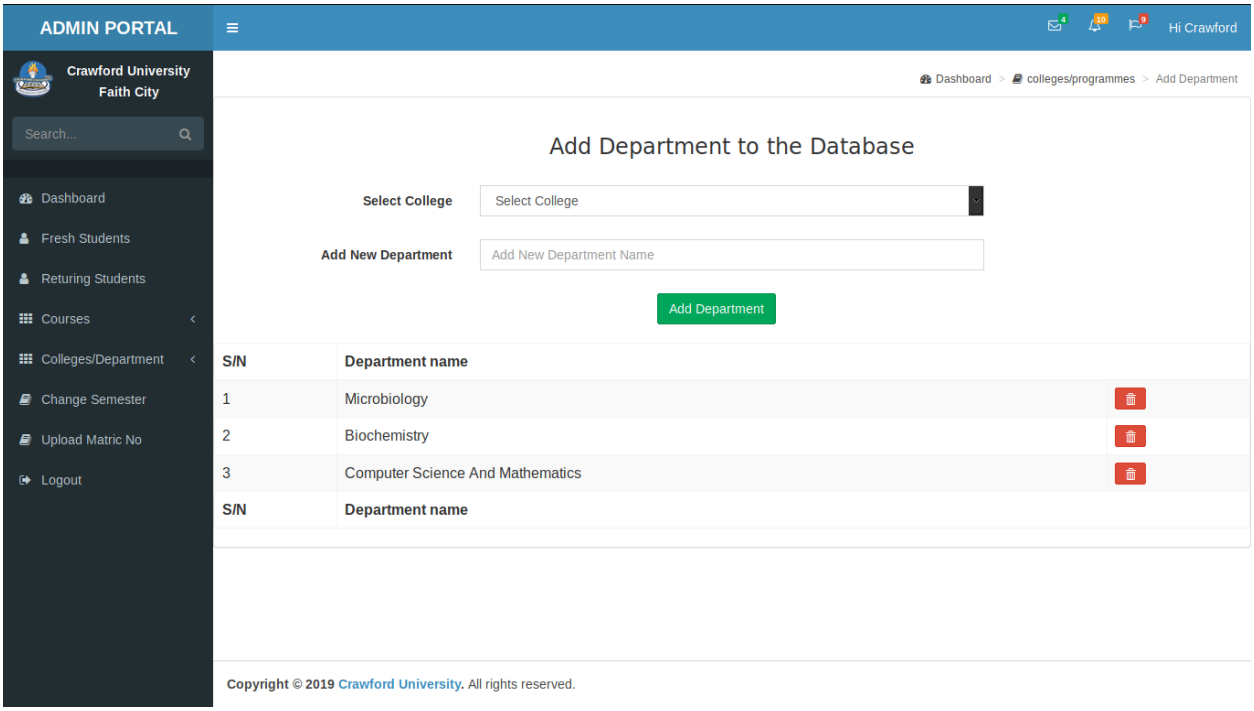

Figure 4.7.4 Admin add department view

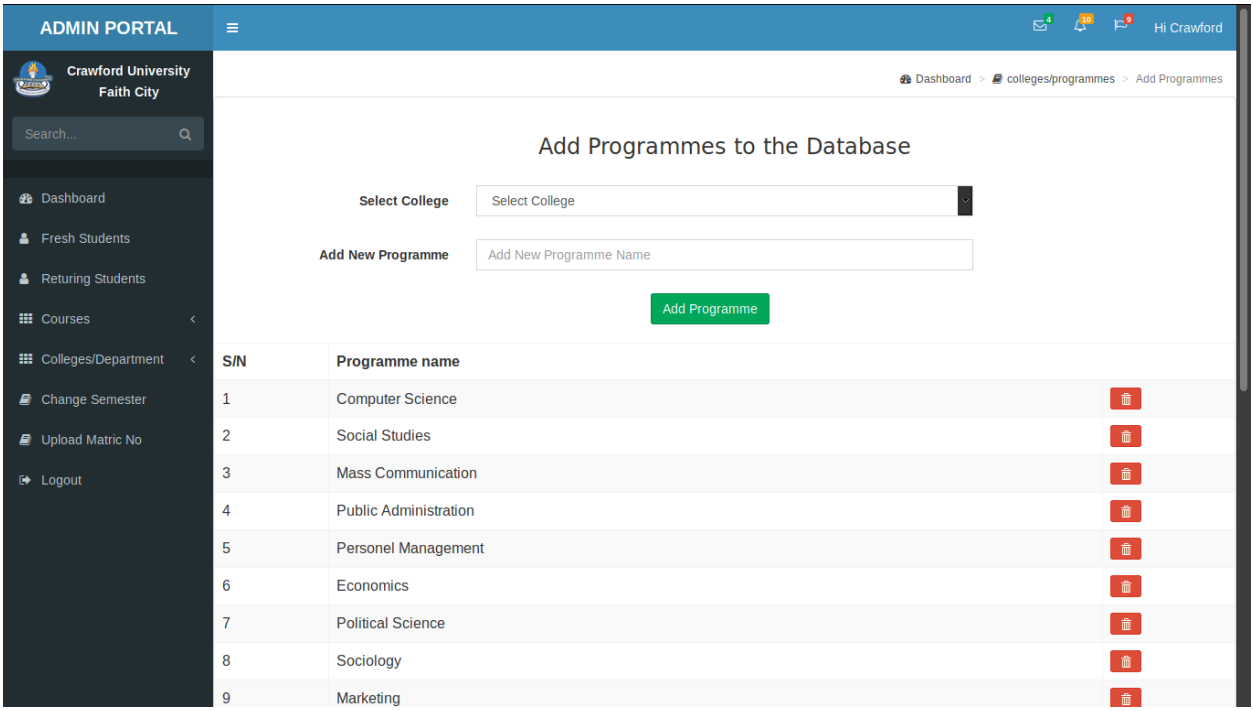

Figure 4.7.5 Admin add programmes view

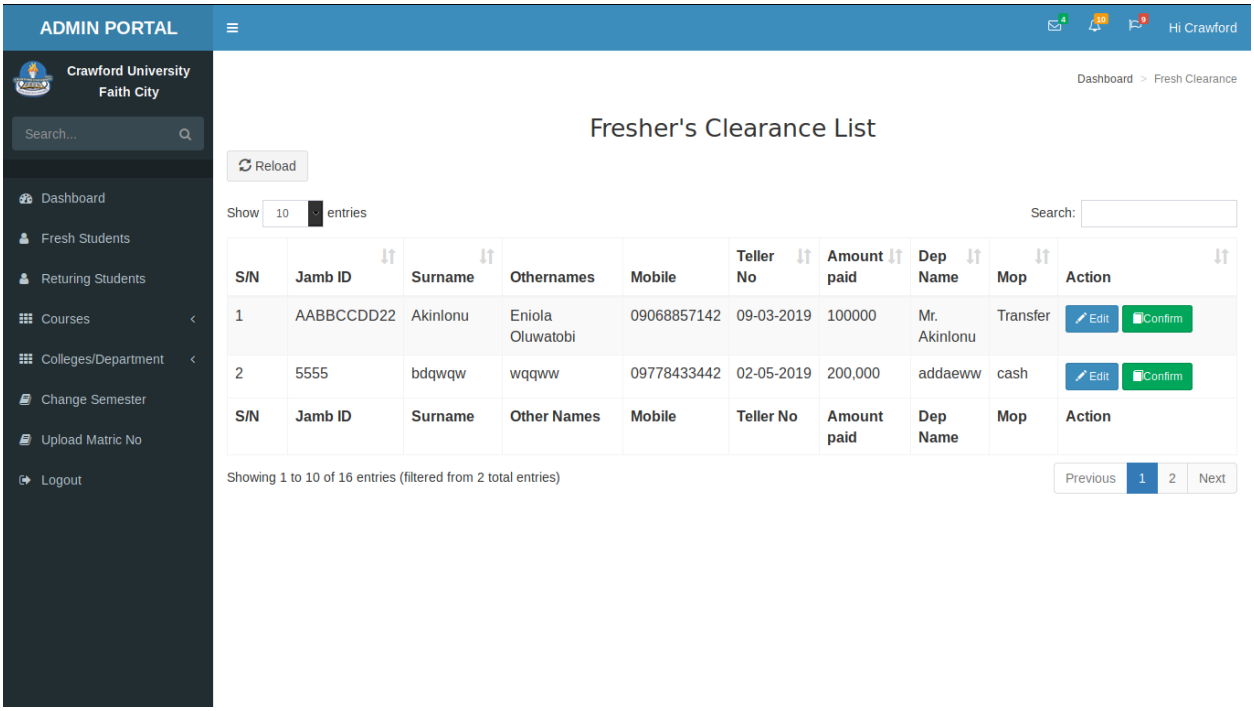

Figure 4.7.6 Admin fresher's clearance list view

4.2 SOFTWARE TESTING

This software system defines mainly three (3) stages of testing which were undergone in the development of the system:

i. Module testing

ii. System and Integration testing

iii. Acceptance testing

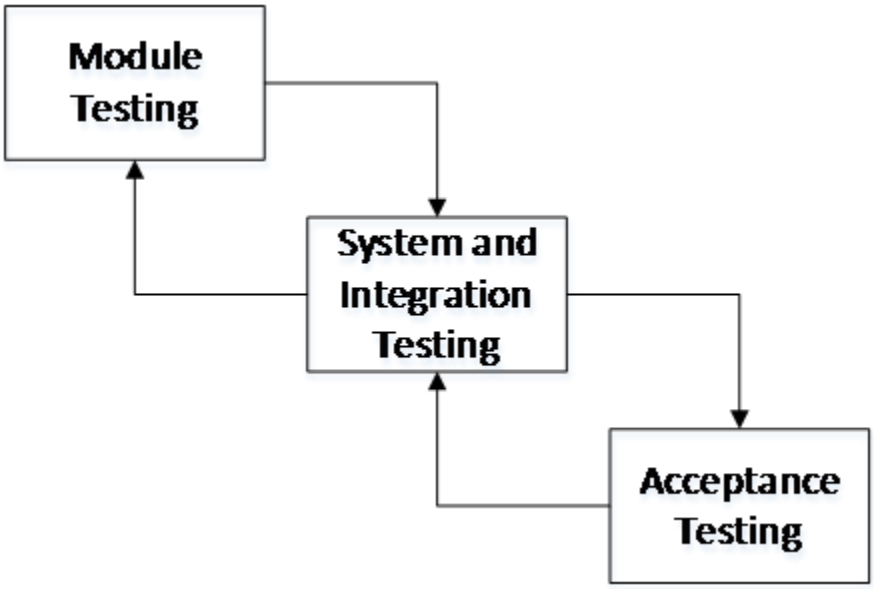

Figure 4.8 Types of software testing

4.2.1 MODULE TESTING

This stage of testing involved testing the individual modules independent of one another as they were developed. In some cases of module testing, errors were discovered which led to the codes for the module responsible being revised to uncover the cause of such errors.

4.2.2 SYSTEM AND INTEGRATION TESTING

During this stage, integration of the modules happened gradually, one module after the other, following a priority, until all the modules were integrated. As the integration happened gradually, system testing of the initial integrated modules also took place alongside. After the final integration of all the modules, system testing was performed to ensure no issues with the modules integration process.

4.2.3 ACCEPTANCE TESTING

This is the final stage of the testing process. This stage defines the testing of the whole software system after it had being packaged to further ensure that the software system will be acceptable, understandable, usable and convenient for use by users.

As seen in Figure 4.8, this testing process is two-way. Hence, one can always go back to check previous stage if there happens to be any errors encountered in any stage of the testing.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

The existing web portal has been faithfully successful in rendering the services it was built for. Students have registered courses on it for many years before I got admission into the institution till date, clearances have been issued, results have been processed and many more transactions it offers has been fulfilled. As it is sure that technology transitions over the years, the state and nature of modernized web technology has deprecated the current portal and a state-of-the-art upgrade is necessary. The institution is ungratified with the system. This project therefore will evolve a smart, light-weight and secure web application portal, thoroughly optimized in performance, speed and effectiveness. It provides the already existing services with modern approaches, with supplementary functionalities for bursary, security, registry, hostel and parent units. The new project guarantees data integrity through its stringent form validation rules, high security and error free execution.

The primary purpose of this project is to replace the existing system with error free, high speed and low cost and should improve the system capability, though in the long run, more features could still be added which will entail expanding the program to be more complex than this.

5.2 CONCLUSION

It is said that *"You know by doing".* This research work has familiarized me with contemporary web trends in web application. As an experienced developer, I still got to master new and crucial skill sets and ideas such as using the codeigniter framework, git, and SQL functions.

I learnt the following during this project

i. PRG – post/Redirect/Get - is a web development design pattern that prevents some duplicate form submissions, creating a more intuitive interface for user agents (users).

ii. How to place a submit button outside a <form> tag – This is done using the "form" attribute on the submit button, supplying the form name to the attribute.

```
<form id="myform" method="get" action="something.php">
        <input type="text" name="name" />
</form>
<input type="submit" form="myform" />
```

iii. How to insert multiple rows into database with one insert using mysql syntax

```
INSERT INTO table (c1, c2, …) VALUES (v11, v12, …),
        VALUES (V21, V22, …),
        VALUES (V31, V32, ...)
```

iv. Set initial AUTO_INCREMENT value for SQL column

v. How to use bootstrap

Therefore, this project may help other developers to experience even more knowledge than I encountered. Working on the project was good experience. I understand the importance of planning and designing as a part of software development.

5.3 RECOMMENDATION

I highly recommend that the system should be made an SPA – A **single-page application** (**SPA**) is a web application or web site that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server. This can be done with angular javascript.

A comprehensive training program should be organized by the institution to equip the staff and students with the basic principle of the new system.

The new system be maintained more often.

REFERENCES

Allan, R. (2018). Web Portal. Retrieved July 6, 2019, from ScienceDirect website: https://www.sciencedirect.com/topics/economics-econometrics-and-finance/web-portal

Aocoed. (2013). About Aocoed. Retrieved July 6, 2019, from http://aocoed.edu.ng/

Bashir, L. (2014). *DESIGN AND IMPLEMENTATION OF AN ONLINE PORTAL REGISTRATION: A CASE STUDY OF NATIONAL OPEN UNIVERSITY OF NIGERIA, DAMATURU STUDY CENTRE*.

CodeIgniter. (2019). CodeIgniter Rocks. Retrieved July 6, 2019, from https://www.codeigniter.com/

Computer Hope. (2019). CSS. Retrieved July 6, 2019, from https://www.computerhope.com/jargon/c/css.htm

Gerd, W. (2015). *Good-Bye, SAP Design Guild*. Retrieved from https://experience.sap.com/news/good-bye-sap-design-guild/?b=/innovation/artikel/walo2.html&t=/innovation/innovation_topnav.html

Gibb, R. (2016). What is a web Application? Retrieved July 6, 2019, from STACKPATH website: https://blog.stackpath.com/web-application/

Gibb, R. (2016). How web applications work. Retrieved July 6, 2019, from STACKPATH website: https://blog.stackpath.com/web-application/

Guru99. (2019). What is CodeIgniter? How does it Work? Retrieved July 6, 2019, from https://www.guru99.com/what-is-codeigniter.html

Jashon, N. (2013). *Web in the modern world*.

Kohan, B. (2017). Guide to Web Application Development. Retrieved July 6, 2019, from comentum 360 website: https://www.comentum.com/guide-to-web-application-development.html

Marcus, P., & Dafydd, S. (2010). *The Evolution of Web Applications* (2nd ed.). Retrieved from https://www.oreilly.com/library/view/the-web-application/9781118026472/chap01-sec001.html

Margarete Rouse. (2018). MySQL. Retrieved July 6, 2019, from TechTarget website: https://searchoracle.techtarget.com/definition/MySQL

Margarete Rouse. (2019). web application development. Retrieved July 6, 2019, from TechTarget website: https://searchcloudcomputing.techtarget.com/definition/web-application-development

Margarete Rouse. (2019). Internet. Retrieved July 6, 2019, from TechTarget website: https://searchwindevelopment.techtarget.com/definition/Internet

Margarete Rouse. (2017). World Wide Web (WWW). Retrieved July 6, 2019, from TechTarget website: https://whatis.techtarget.com/definition/World-Wide-Web

Martin, J. (2017). How does a web application work? Retrieved July 6, 2019, from Quora website: https://www.quora.com/How-does-a-web-application-work

Morris, S. (2018). Everything You Need to Know About PHP. Retrieved July 6, 2019, from Skillcrush website: https://skillcrush.com/2012/04/11/php/

Nathan, J. (2016). *Developer Justice*.

Nations Daniel, E. M. (2019). What Exactly Is a Web Application? Retrieved July 6, 2019, from Lifewire website: https://www.lifewire.com/what-is-a-web-application-3486637

Remick, J. (2011). What Is a Web App? Here's Our Definition. Retrieved July 6, 2019, from AppStorm website: https://web.appstorm.net/general/opinion/what-is-a-web-app-heres-our-definition/

Rick, S. (n.d.). *A portal - thesis*. Retrieved from https://www.blacksmith.com/books/reports/portals/4432443/32/rick.s

Rouse, M. (2011). Web application (Web app). Retrieved July 6, 2019, from TechTarget website: https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app

Shannon, R. (2019). What is HTML? Retrieved July 6, 2019, from HTML Source website: https://www.yourhtmlsource.com/starthere/whatishtml.html

Techopedia. (2015). WORLD WIDE WEB (WWW). Retrieved July 10, 2019, from https://www.techopedia.com/definition/5217/world-wide-web-www

Techopedia. (2019). Scripting Language. Retrieved July 6, 2019, from https://www.techopedia.com/definition/3873/scripting-language

Techopedia. (2019). Programming Tool. Retrieved July 6, 2019, from https://www.techopedia.com/definition/8996/programming-tool

Techopedia. (2019). Client. Retrieved July 6, 2019, from https://www.techopedia.com/definition/437/client

Tutorialspoint. (2019). Design Patterns in Java Tutorial. Retrieved July 6, 2019, from https://www.tutorialspoint.com/design_pattern/index.htm

Tutorialspoint. (2019). Design Patterns - MVC Pattern. Retrieved July 6, 2019, from https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm

Winkler, R. (2016). *Information System Portal*. Retrieved from https://www.scribd.com/document/360901884/Information-System-Portal-2824-Pages-29

WpBeginner. (2019). What is: Apache. Retrieved July 6, 2019, from https://www.wpbeginner.com/glossary/apache/

APPENDIX

Academic_Login.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Academic_Login extends CI_Controller {

	public function __construct(){

		parent::__construct();
		$this->load->model('student_model','student');
		$this->load->model('admin_model','admin');
		$this->load->model('portal_model','portal');

	}
	public function index(){

		$rules = $this->portal->rules;
			$this->form_validation->set_rules($rules);
			if ($this->form_validation->run() == true){
			redirect('My_portal');
			}else{

		$data['title'] = "Login Page";
		$this->load->view('header', $data);
		$this->load->view('portal/portal_login_view', $data);
		$this->load->view('footer2', $data);
		}
	}

	public function Reset_Login(){

		$rules = $this->portal->rules;
			$this->form_validation->set_rules($rules);
			if ($this->form_validation->run() == true){
			redirect('My_portal');
			}else{
		$data['account_created'] = 'Your Password Has Been Updated Successfully';
		$data['title'] = "Login Page";
		$this->load->view('header', $data);
		$this->load->view('portal/portal_login_view', $data);
		$this->load->view('footer2', $data);
		}
	}



	public function validate_user_portal(){
```

```
		$user  = $this->input->post('username');
		//$pass  = $this->input->post('password');
		$pass  = $this->hash($this->input->post('password'));


		$result = $this->portal->validate_user_portal($user, $pass);

		if ($result)// if user's detail is valid..
		{
			$sess_array = array();
			foreach ($result as $row ){
			$sess_array = array('p_id' => $row->p_id,
			 'matric' => $row->matric,
			  'level' => $row->curr_level,
			  'department' =>$row->department,
			 'password' => $row->password);
			$this->session->set_userdata('logged_in', $sess_array);

			}
			redirect('My_portal');
			}else{
		$this->form_validation->set_message('validate_user_portal', 'Please, Check Your
Username and Password');
		return false;
			}

		}


		//===========hashing of password===========
		public function hash($string){
		return hash ('sha1', $string . config_item('encryption_key'));
					}


}
```

Clearance.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Clearance extends CI_Controller {

	public function __construct(){

		parent::__construct();
		$this->load->model('student_model','student');
		$this->load->model('admin_model','admin');

	}
	public function index(){

		$data['title'] = "Resumption Clearence";
		$this->load->view('header', $data);
		$this->load->view('clearance_view', $data);
		$this->load->view('footer2', $data);

	}

	public function new_students(){
		$data['title'] = "New Students";
		$this->load->view('header', $data);
		$this->load->view('new_students_view', $data);
	}

	public function save(){
		if(isset($_POST["submit"]) == "submit")
			{
		$query = $this->student->save_clearance();
		if($query){
		$data['account_created'] = 'Clearance Registration Successfull.';
		$data['title'] = "New Students";
		$this->load->view('header', $data);
		$this->load->view('new_students_view', $data);
		}else{
			echo "E no Enter...";
		}

		}else{
			//$this->student->save_clearance();
		$data['title'] = "New Students";
		$this->load->view('header', $data);
		$this->load->view('new_students_view', $data);
		}
	}
```

```
}
```

Home.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Home extends CI_Controller {

		public function index()
	{
		$data['title'] = "My Portal";

		$this->load->view('header', $data);
		$this->load->view('home_view', $data);

	}
}
```

Student_Portal.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Student_Portal extends CI_Controller {

	public function __construct(){

		parent::__construct();
		$this->load->model('student_model','student');
		$this->load->model('admin_model','admin');

	}

	Public function index(){

		$data['title'] = "My Dashboard";
		$this->load->view('header', $data);
		$this->load->view('portal/welcome_page_view', $data);
	}




}
```

Courses_Model.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Courses_Model extends CI_Model {
public $rules = array(
        'username' =>array(
                'field' => 'username',
                'label' => 'Username',
                'rules' => 'trim|required'
                ),

        'password' =>array(
                'field' => 'password',
                'label' => 'Password',
                'rules' => 'trim|required|callback_validate'
                ),

        );

var $table = 'courses';
        var $select_column =
array('course_id','course_level','c_semester','course_code','course_title','course_units','course_status','c_
college','c_programme','c_department'); //set column field database for datatable orderable
        var $order_column = array(null,'course_code','course_title','course_level',null); //set column
field database for datatable searchable just firstname , lastname , address are searchable


        public function __construct()
        {
                parent::__construct();

        }

        function make_query()
        {
                $this->db->select($this->select_column);
                $this->db->from($this->table);
                if(isset($_POST['search']['value'])){
                        $this->db->like('course_code', $_POST['search']['value']);
                        //$this->db->or_like('teller', $_POST['search']['value']);
                        $this->db->or_like('course_title', $_POST['search']['value']);

                        $this->db->or_like('course_level', $_POST['search']['value']);
                        }

                        if(isset($_POST['order'])){

                                $this->db->order_by($this-
>order_column[$_POST['order']['0']['column']], $_POST['order']['0']['dir']);
```

```
                }else{

                        $this->db->order_by('course_id', 'DESC');

                }

        }

        function make_datatables(){
                $this->make_query();
                if($_POST['length'] != -1)
                $this->db->limit($_POST['length'], $_POST['start']);
                $query = $this->db->get();
                return $query->result();
        }


        function get_filtered_data()
        {
                $this->make_query();
                $query = $this->db->get();
                return $query->num_rows();
        }


        public function get_all_data()
        {
                $this->db->select("*");
                $this->db->from($this->table);
                return $this->db->count_all_results();
        }

}
```

Get_Model.php

```
<?php if (! defined('BASEPATH')) exit('No direct script access allowed');?>

<?php
	class Get_Model extends CI_Model{



public function fetch_state(){

			$this->db->order_by('state_name', 'ASC');
			$query =$this->db->get('state');
			return $query->result();
		}


public function fetch_colleges(){

			$this->db->order_by('college_name', 'ASC');
			$query =$this->db->get('colleges');
			return $query->result();
		}



		public function fetch_lga($state_id){

			$this->db->where('state_id', $state_id);
			$this->db->order_by('lga_name', 'ASC');
			$query = $this->db->get('lga');
			$output = '<option value="">Select LGA</option>';
			foreach ($query->result() as $row) {
				$output .= '<option value="'.$row->lga_name.'">'.$row-
>lga_name.'</option>';
			}
				return $output;
		}

		public function fetch_stat($state_id){

			$this->db->where('state_id', $state_id);
			$query = $this->db->get('state');

			foreach ($query->result() as $row) {
				$output = $row->state_name;

			}
				return $output;
		}
```

```
	public function fetch_depart($college_id){

		$this->db->where('college_id', $college_id);
		$this->db->order_by('programme_name', 'ASC');
		$query = $this->db->get('programmes');
		$output = '<option value="">Select programme</option>';
		foreach ($query->result() as $row) {
			$output .= '<option value="'.$row->programme_name.'">'.$row->programme_name.'</option>';
		}
			return $output;
	}

	public function fetch_college($college_id){

		$this->db->where('college_id', $college_id);
		$query = $this->db->get('colleges');

		foreach ($query->result() as $row) {
			$output = $row->college_name;

		}
			return $output;
	}

}
```