|  |  |  |
| --- | --- | --- |
| Image | |  |
| REPORTINTERNSHIP TRAINING AT KMC | | |
| KIIRA MORTORS CORPORATIONThursday, 3 September, 2020AUTHOR: STEPHEN TIPA AUGUSTINE |  | |

|  |
| --- |
|  |
| **EXECUTIVE SUMMARY** The training at KMC for the past weeks has been splendid.  The author was able to acquire so many skills from different aspects in the course.  The training started back in the last five weeks and this being the sixth, the author was able to get intuition on;-   1. Web development   An abstract definition of web development is the creation of a web application generally known as a web-site for the Internet (The World Wide Web) or an intranet within an organization [1].   1. Mobile application development   Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Hence, the mobile development process involves creating installable software bundles (code, binaries, assets, etc.), implementing backend services such as data access with an API, and testing the application on target devices [3].   1. Artificial Intelligence.   Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals.  Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving [4].  In each of the above three fields mentioned above the author was able to venture into background information or theories concerning them, and also familiarized himself with the concepts in these fields by practical programming codes, and examples some of which have been mentioned in the Internship Training Details section. |

### **TABLE OF CONTENTS**

Contents

[REPORT 2](#_Toc50017709)

[INTERNSHIP TRAINING AT KMC 2](#_Toc50017710)

[KIIRA MORTORS CORPORATION 2](#_Toc50017711)

[Thursday, 3 September, 2020 2](#_Toc50017712)

[AUTHOR: 2](#_Toc50017713)

[EXECUTIVE SUMMARY 3](#_Toc50017714)

[TABLE OF CONTENTS 4](#_Toc50017715)

[INTRODUCTION 4](#_Toc50017716)

[INTERNSHIP TRAINING DETAILS 5](#_Toc50017717)

[CONCLUSIONS 17](#_Toc50017718)

[REFERENCES 18](#_Toc50017719)

### **INTRODUCTION**

Following the internship placement offer granted on the letter issued to the author from the CEO KMC on 5th August, 2020, the author intends to address this report giving the details of the work he performed from the date of the start of his training to when this report was written. Since the training was a requirement of the bachelors program in Makerere University, the Corporation drew a plan of how the training was supposed to run for the eight planned weeks. Figure 1 below shows the plan.

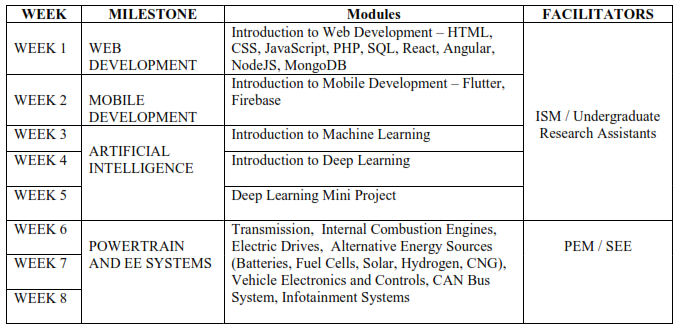


Figure 1: Program plan

The author organized this paper in four sections;-

The paper starts by the Executive summary section, which gives the general overview of the paper.

The Introduction which is the current section: This section will give brief information on the aim of this report.

The Internship training Details section, which will give detailed information on the work performed.

Then lastly but not least the Conclusion, where the author gave general overview of the report as well as achievements and challenges encountered during the period of the training.

The paper ends by giving a list of references to citations.

Note: The referencing style used here is the IEEE referencing style.

### **INTERNSHIP TRAINING DETAILS**

As stated in the preceding section of this report, in this particular section of the report the author describes briefly the details of what he did during this period of the training.

##### WEB DEVELOPMENT

This part of the program was designed to give the author an intuition on introduction to web development. An abstract definition of web development is the creation of a web application generally known as a web-site for the Internet (The World Wide Web) or an intranet within an organization [1].

The web development process involve two main aspects, that’s the frontend as well as the backend development.

The former is the part of the website that the user directly interacts with, it is a times referred to as the client side [2].  The common languages used for Front End development based on the data from [2] include;-

* HTML: HTML stands for Hyper Text Markup Language. It is used to design the front end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.
* CSS: Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
* JavaScript: JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

Famous frameworks designed for the frontend development include the following according to [2];-

* AngularJS: AngularJs is a JavaScript open source front-end framework that is mainly used to develop single page web applications (SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open source project which can be freely used and changed by anyone. It extends HTML attributes with Directives, and data is bound with HTML.
* React.js: React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.
* Bootstrap: Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.
* jQuery: jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.
* SASS: It is the most reliable, mature and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.
* Some other libraries and frameworks are: Semantic-UI, Foundation, Materialize, Backbone.js, Express.js, Ember.js etc.

The following figures obtained from Chart created by GitHub user Kamranahmedse: <https://github.com/kamranahmedse>, shows the frontend and backend developer roadmaps respectively.

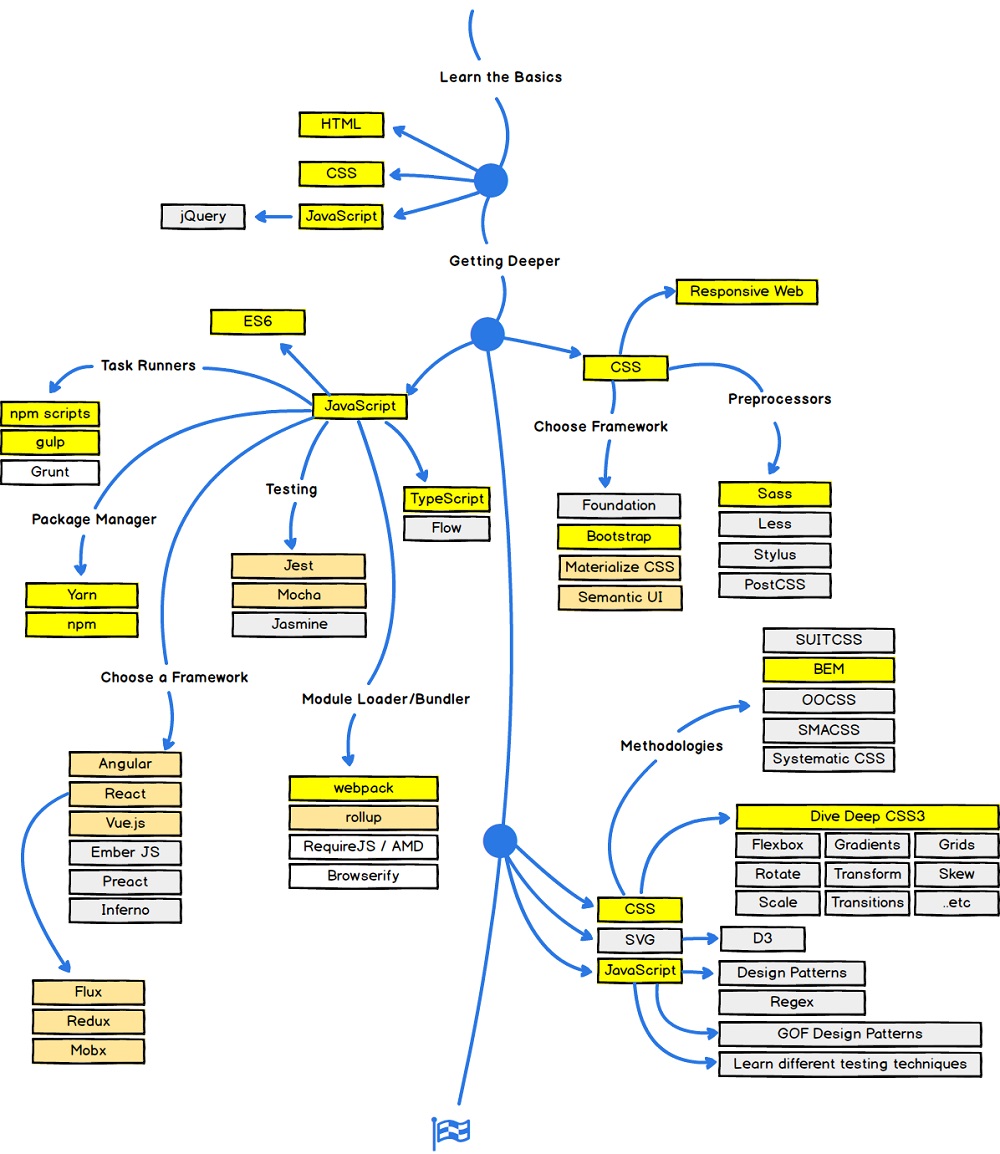


Figure 2: Frontend road map

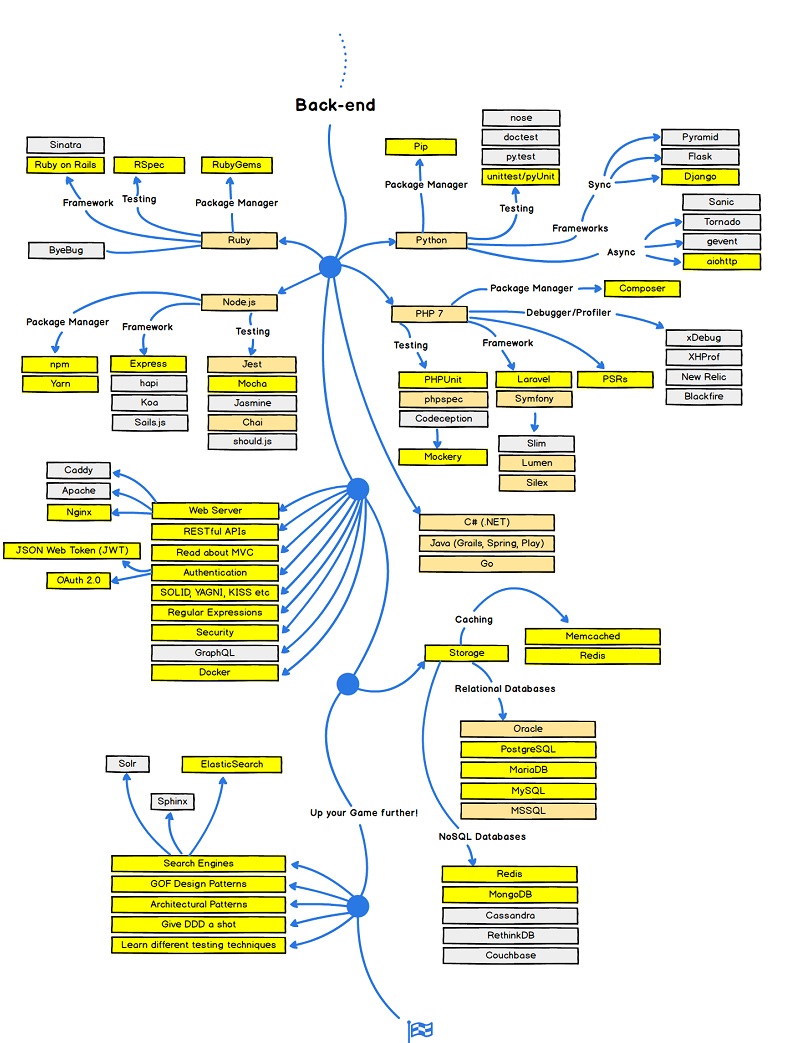


Figure 3: Backend road map

On the other hand, the latter is server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. The languages used in the backend according to [2] include;-

* PHP: PHP is a server-side scripting language designed specifically for web development. Since PHP code executed on the server side so it is called server-side scripting language.
* C++: It is a general purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
* Java: Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available.
* Python:Python is a programming language that lets you work quickly and integrate systems more efficiently.
* JavaScript: Javascript can be used as both (front end and back end) programming languages.
* Node.js: Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it’s not a programming language. Most of the people are confused and understand it’s a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It’s used in production by large companies such as Paypal, Uber, Netflix, Wallmart and so on.

The famous frameworks used in the backend development include;-

* The list of back end frameworks are: Express, Django, Rails, Laravel, spring, etc.
* The other back end program/scripting languages are: C#, Ruby, REST, GO etc.

**The Internship Training**

During the training in this field, the author was able to gain greater intuitions on both aspects that is the frontend and backend developments.

1. The author started by looking at the basic HTML,CSS, and JavaScript, he when further exploring the purpose of each, when to use each, and the best practices of using each.
2. The author then looked at the jQuery JavaScript based framework into depth.
3. Following the above the author looked at the angular frontend development framework for the frontend that was developed by Google.
4. Lastly in the frontend development, the author learned about the famous React Js JavaScript based frontend development framework developed by Facebook. He went further exploring the different features in this framework and what makes it different from Google’s Angular framework.
5. Then following the roadmaps in the frontend, the author started by looking at the databases, both relational and non-relational databases. The strengths of each as well as their weaknesses. Here the author interfaced with MySQL, PostgreSQL, Mongo databases
6. The author also looked at Python programming language and its famous Django framework for the backend development.
7. The author also learned a few basics of PHP, Java and Node Js for the backend development.

As evidence of the work done, the author has created the following GitHub repositories with their links listed to demonstrate the practice he underwent during the training.

1. <https://github.com/Stephen-Tipa-Augustine/RecruitmentFirm.git>
2. <https://github.com/Stephen-Tipa-Augustine/AngularProject.git>
3. <https://github.com/Stephen-Tipa-Augustine/Day1_Html_CSS_Js.git>

##### MOBILE DEVELOPMENT

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Hence, the mobile development process involves creating installable software bundles (code, binaries, assets, etc.) , implementing backend services such as data access with an API, and testing the application on target devices [3].

**Mobile Applications and Device Platforms**

There are two dominant platforms in the modern smartphone market. One is the iOS platform from Apple Inc. The iOS platform is the operating system that powers Apple's popular line of iPhone smartphones. The second is Android from Google. The Android operating system is used not only by Google devices but also by many other OEMs to build their own smartphones and other smart devices [3].

Although there are some similarities between these two platforms when building applications, developing for iOS vs. developing for Android involves using different software development kits (SDKs) and different development toolchain. While Apple uses iOS exclusively for its own devices, Google makes Android available to other companies provided they meet specific requirements such as including certain Google applications on the devices they ship. Developers can build apps for hundreds of millions of devices by targeting both of these platforms [3].

**Alternatives for Building Mobile Apps**

According to [3], there are four major development approaches when building mobile applications.

Each of these approaches for developing mobile applications has its own set of advantages and disadvantages. When choosing the right development approach for their projects, developers consider the desired user experience, the computing resources and native features required by the app, the development budget, time targets, and resources available to maintain the app.

1. Native Mobile Applications

Native mobile applications are written in the programming language and frameworks provided by the platform owner and running directly on the operating system of the device such as iOS and Android.

1. Cross-Platform Native Mobile Applications

Cross-platform native mobile applications can be written in variety of different programming languages and frameworks, but they are compiled into a native application running directly on the operating system of the device.

1. Hybrid Mobile Applications

Hybrid mobile applications are built with standard web technologies - such as JavaScript, CSS, and HTML5 - and they are bundled as app installation packages. Contrary to the native apps, hybrid apps work on a 'web container' which provides a browser runtime and a bridge for native device APIs via Apache Cordova.

1. Progressive Web Applications

PWAs offer an alternative approach to traditional mobile app development by skipping app store delivery and app installations. PWAs are web applications that utilize a set of browser capabilities - such as working offline, running a background process, and adding a link to the device home screen - to provide an 'app like' user experience.

**The Mobile Application Development Lifecycle**

There are two interlinked core components of a mobile application: 1) the mobile application “Front-End” that resides on the mobile device, and 2) the services “Back-End” that supports the mobile front-end [3].

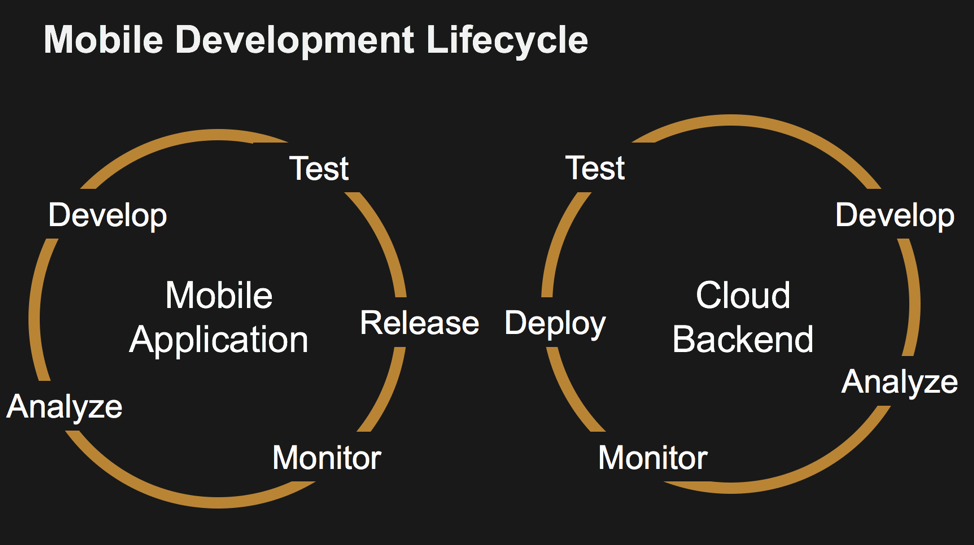


Figure 4: The mobile development life cycle

**The Mobile Application Front-End**

The mobile front-end is the visual and interactive part of the application the user experiences. It usually resides on the device, or there is at least an icon representing the app that is visible on the home screen or is pinned in the application catalog of the device. The application can be downloaded from the platform app store, side-loaded directly onto the device, or can be reached through the device’s browser, as in the case for PWAs [3].

**The Mobile Application Back-End**

Regardless of what front-end platform or development methodology is being used, delivering high-quality mobile applications that delight and retain users requires reliable back-end services.

Given the critical importance of back-end services for the success of the mobile application, the developers have several important architectural decisions that they must consider. These decisions include which services should they build themselves and which third party services should they leverage, and then should they run and maintain their own services or should they take advantage of 3rd party services.

The answer is increasingly clear; to improve developer productivity and efficiency, mobile app programmers should only build their own services if they are highly specific to the domain of the application and embody unique intellectual property. Also, even for the services they build themselves, they should almost always leverage cloud-based services to build and maintain their backend infrastructure.

**How Mobile Aps Integrate with the Backend?**

Regardless of the size of the team, a critical element of the development effort is building the app logic that is responsible for making network calls to the back-end services, retrieve data and update the data in the back-end systems with new information generated from the app.

These back-end services are typically accessed through a variety of application programming interfaces, most commonly known as APIs. There are different types of APIs, such as REST and GraphQL, and there are also a wide variety of means and styles of accessing them. While some back-end service APIs are available directly to the application through calls in the platform itself, many of the specialized services have to be integrated into the app via a software development kit, commonly known as an SDK. Once the SDK has been added to the app via the development environment, then the application can make use of the APIs defined in the SDK.

**Key Mobile Application Services**

There are hundreds of cloud and 3rd party services that mobile application developers can leverage to speed up the development and delivery of their applications. However, it’s unlikely that a developer is going to be able to become an expert in each of these individual services.

Instead, the mobile developers should look for a development environment that makes it easier for them to integrate, use, and consume the most commonly required capabilities into their application quickly and easily, while still preserving the freedom to take advantage of the many individual services available.

**Essential**

* User Sign-up/Sign-in and Management
* Social login (Facebook sign-in, Twitter sign-in, etc.)
* Analytics and User Engagement
* Push Notifications
* Real Device Testing

**Data Services**

* Cloud Storage
* Real-time and Offline Data
* Application Logic/Cloud Functions

**Machine Learning**

* Conversational Bots
* Image and Video Recognition
* Speech Recognition

**The Training**

The training introduced the author to Dart programming language and the Flutter UI toolkit.

Dart is a programming language optimized for building user interfaces with features such as the spread operator for expanding collections, and collection if for customizing UI for each platform. On the other hand, Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.

1. During the training the author was able to learn Dart programming and explore many of its core concepts.
2. The author also learned build mobile applications for Android platform using the Flutter UI toolkit, Dart programming language and Android Studio SDK.
3. At the end of the training in this field, the author was able to demonstrate his skills in designing a mobile application for ordering items from a Hotel or Restaurant, due to the limited time frame of one week he was not able to get everything to the desired goals of the application. But the program has authentication system, then displays the available hotels in Kampala. When the user selects one of the hotel, he is redirected to the screen from where he can order and perform several other operations.

A full code base of this project is available in an open GitHub repository at: <https://github.com/Stephen-Tipa-Augustine/hotel_connect.git>

##### ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals.

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving [4].

Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals [5].

Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving" [6].

**WHAT ARE THE DIFFERENT TYPES OF AI?**

At a very high level artificial intelligence can be split into two broad types: narrow AI and general AI.

Narrow AI is what we see all around us in computers today: intelligent systems that have been taught or learned how to carry out specific tasks without being explicitly programmed how to do so.

This type of machine intelligence is evident in the speech and language recognition of the Siri virtual assistant on the Apple iPhone, in the vision-recognition systems on self-driving cars, in the recommendation engines that suggest products you might like based on what you bought in the past. Unlike humans, these systems can only learn or be taught how to do specific tasks, which is why they are called narrow AI.

Artificial general intelligence is very different, and is the type of adaptable intellect found in humans, a flexible form of intelligence capable of learning how to carry out vastly different tasks, anything from haircutting to building spreadsheets, or to reason about a wide variety of topics based on its accumulated experience. This is the sort of AI more commonly seen in movies, the likes of HAL in 2001 or Skynet in The Terminator, but which doesn't exist today and AI experts are fiercely divided over how soon it will become a reality.

**Machine Learning**

There is a broad body of research in AI, much of which feeds into and complements each other.

Currently enjoying something of a resurgence, machine learning is where a computer system is fed large amounts of data, which it then uses to learn how to carry out a specific task, such as understanding speech or captioning a photograph.

**Neural Networks and Deep Learning**

Key to the process of machine learning are neural networks. These are brain-inspired networks of interconnected layers of algorithms, called neurons, that feed data into each other, and which can be trained to carry out specific tasks by modifying the importance attributed to input data as it passes between the layers. During training of these neural networks, the weights attached to different inputs will continue to be varied until the output from the neural network is very close to what is desired, at which point the network will have 'learned' how to carry out a particular task.

A subset of machine learning is deep learning, where neural networks are expanded into sprawling networks with a huge number of layers that are trained using massive amounts of data. It is these deep neural networks that have fueled the current leap forward in the ability of computers to carry out task like speech recognition and computer vision.

There are various types of neural networks, with different strengths and weaknesses. Recurrent neural networks are a type of neural net particularly well suited to language processing and speech recognition, while convolutional neural networks are more commonly used in image recognition. The design of neural networks is also evolving, with researchers recently refining a more effective form of deep neural network called long short-term memory or LSTM, allowing it to operate fast enough to be used in on-demand systems like Google Translate.

The following figure shows the perceptron intuition of the neuron in Neuron networks, and thus deep learning.

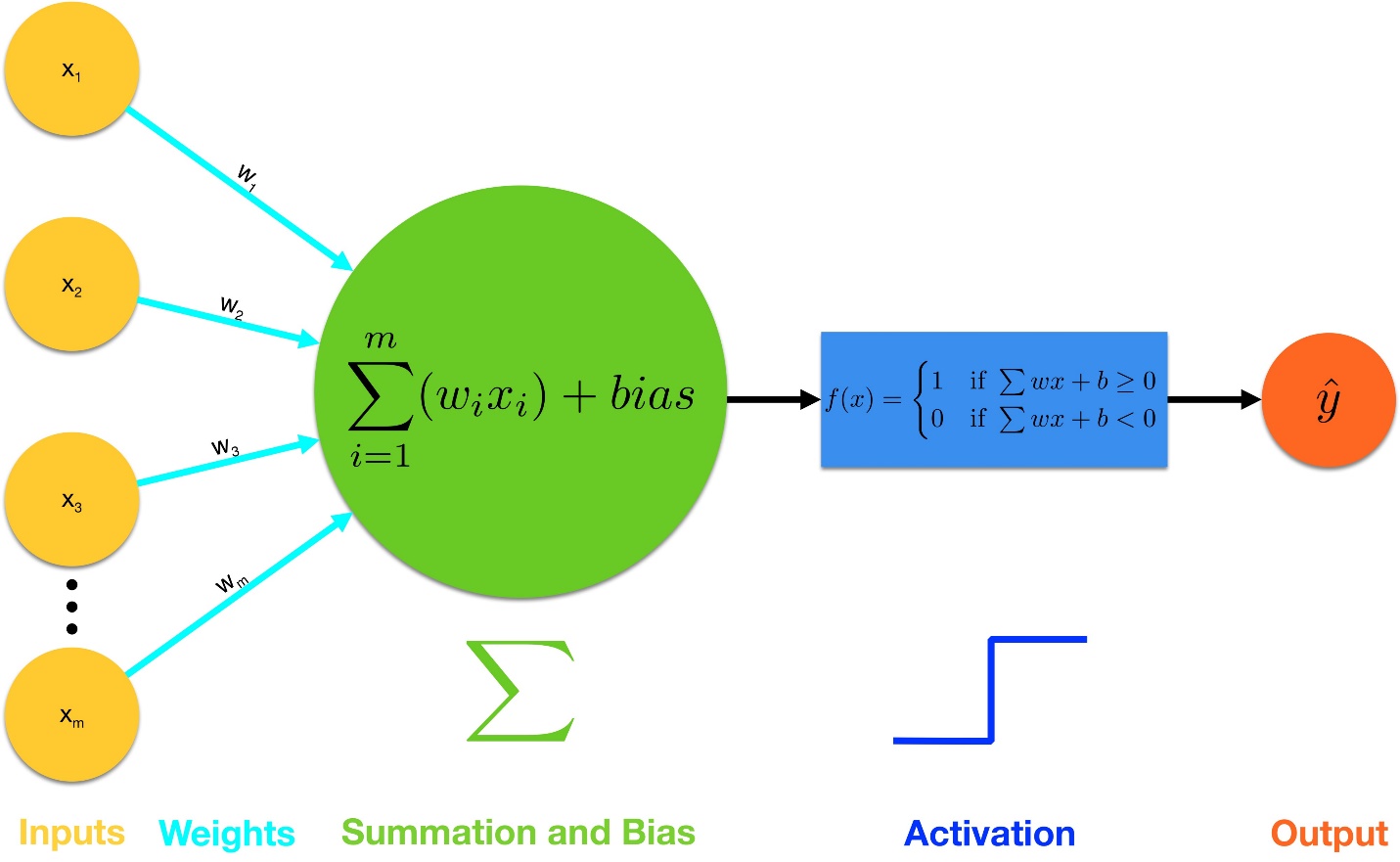


Figure 5: Perceptron based neuron

**Elements of Machine Learning**

1. Supervised Learning: A common technique for teaching AI systems is by training them using a very large number of labelled examples. These machine-learning systems are fed huge amounts of data, which has been annotated to highlight the features of interest. These might be photos labelled to indicate whether they contain a dog or written sentences that have footnotes to indicate whether the word 'bass' relates to music or a fish. Once trained, the system can then apply these labels can to new data, for example to a dog in a photo that's just been uploaded.
2. Unsupervised learning: In contrast, unsupervised learning uses a different approach, where algorithms try to identify patterns in data, looking for similarities that can be used to categorize that data.

An example might be clustering together fruits that weigh a similar amount or cars with a similar engine size. The algorithm isn't setup in advance to pick out specific types of data, it simply looks for data that can be grouped by its similarities, for example Google News grouping together stories on similar topics each day.

1. Reinforcement learning: A crude analogy for reinforcement learning is rewarding a pet with a treat when it performs a trick. In reinforcement learning, the system attempts to maximize a reward based on its input data, basically going through a process of trial and error until it arrives at the best possible outcome. An example of reinforcement learning is Google DeepMind's Deep Q-network, which has been used to best human performance in a variety of classic video games. The system is fed pixels from each game and determines various information, such as the distance between objects on screen.

**The Training**

1. The author started exploring this field by looking at some aspect of data science in Python programming language. He explored the numpy, pandas, scipy, matplotlib and sympy python modules to get insights of data science.
2. The author also looked at some aspects of automation of computer tasks in Python programming language. In which he ventured into string operations, regular expressions, and also working with files and file paths in python using built-in modules like os, and also third party modules.
3. The author also learned many concepts of computer vision using the third party library called opencv using Python. He looked at working with images as well as videos from a file or from and input device like a camera.
4. The author went as far as learning about machine learning in Python using the scikit learn library.
5. The author was also able to start a specialization course on Coursera on Deep Learning and Neural networks. Currently he has completed the first course in the specialization and proceeding with the rest. There are five courses in the specialization.

As an evidence to this, the author attaches the Certificate of merit obtained for the first course.

### **CONCLUSIONS**

In general the training has been splendid, it has allowed me to gain skills in many aspects of programming in several fields. Some of this fields include web development, mobile application development, and Artificial intelligence. This not being the first encounter of the author with computer programming, the training has enabled the author to venture in more frameworks, and APIs (Application Programming Interfaces) thus, giving him a bigger sample space to pick from in executing real world projects basing on the requirements and designs of the particular problem at hand.

One of the great achievements of the author during the training at KMC allowed him to interact many individuals more specially his fellow interns during team work and collaborations. This has enhanced his team working skills.

The other is getting acquainted with proficiency in programming.

The author also gained a lot of communication skills during the interactions with his fellow interns, the staff members as well as the CEO KMC.

The list of the achievements are almost endless, if I were to list them all, then I would require a dedicated section of this paper for this purpose. However the few mentioned above are some of the core ones to say.

I would like to end this section by giving some of the challenges encountered during the training. In general the training has been really great success with just a few stumbling blocks. Here I will mention only one, and it is the fact that I possess a dual core Intel Laptop. This machine is literally inferior if weighed on the scale of the things I use it for. Some of the development environment like Android Studio requires high memory and greater computer resources and thus, making it impossible for me to complete some tasks as quickly as I can. As this is a retarding agent. Other than that, most of the things has been explicit and fabulous.

Lastly I treasure the time I had in KMC till to date for the great achievements during the training. I will continue treasuring the times I have with KMC.

### **REFERENCES**

[1] “What is Web Development? - Definition from Techopedia,” *Techopedia.com*.

[2] “Frontend vs Backend - GeeksforGeeks.” [Online]. Available: https://www.geeksforgeeks.org/frontend-vs-backend/. [Accessed: 03-Sep-2020].

[3] “Web Development.” [Online]. Available: https://www.w3schools.com/whatis/. [Accessed: 03-Sep-2020].

[4] “Artificial Intelligence (AI) Definition.” [Online]. Available: https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp. [Accessed: 03-Sep-2020].

[5] G. Tecuci, “Artificial intelligence,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 4, no. 2, pp. 168–180, Mar. 2012.

[6] S. Legg and M. Hutter, “A Collection of Definitions of Intelligence,” Jun. 2007.