



UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

FINAL YEAR PROJECT REPORT

PROJECT INDEX: PRJ002

FTTX NETWORK PLANNING AND DOCUMENTATION PLATFORM

STUDENT NAME: ALLAN OOKO

REGISTRATION NUMBER: F17/81920/2017

SUPERVISOR: ENG. VICTOR KYALO

EXAMINER(S): PROF. HEYWOOD OUMA/ MR. KINYWA WACHIRA

This project report is submitted in partial fulfillment of the requirement for the award of Bachelor of Science Degree in Electrical and Electronics Engineering from the University of Nairobi.

Submitted on 26th May 2022

Declaration of Originality

I hereby declare that this project report entitled “FTTX NETWORK PLANNING AND DOCUMENTATION PLATFORM” submitted to the University of Nairobi, Department of Electrical and Electronics Engineering is a record of my unique work under the supervision of Eng. Victor Kyalo.

The project work is submitted in partial fulfilment of the requirement of the award of the degree of Bachelor of Science in Electrical and Electronic Engineering.

This work has not been submitted previously for any other degree award or examination in another institution.

ALLAN OOKO

Signature

Date

CERTIFICATION

This report was submitted to the Department of Electrical and Information Engineering, University of Nairobi for examination with my approval as supervisor:

Eng. Victor Kyalo

Signature

Date

Dedication

This project work is dedicated to all the engineers and technology enthusiasts who have toiled over the years to research on the Fiber Optic technology, and made innovations that have greatly changed the way our world is connected for the better.

Acknowledgment

I would like to sincerely thank Eng. Victor Kyalo, my project supervisor for his support and guidance through this project work.

I register my sincerest appreciations to Victor Wekesa, Planning and Design Engineer at Egypro East Africa Ltd. for his guidance on current FTTx industry trends. Our interactions helped me develop a keen interest in the operation of FTTx networks, and have been a great source of inspiration through this journey.

I recognize and appreciate the encouragement and support of my friends and classmates through this work, and over the five-year college journey.

I acknowledge openrouteservice and OpenStreetMap contributors for the permission to use their routing data in the design.

Most of all, I thank The Lord God Almighty for His enabling through this work, and indeed through all my life and academic journey. I do acknowledge alongside the old poets of Athens that in Him we live and move and have our being.

Table of Contents

Declaration of Originality	2
Dedication	4
Acknowledgment	5
List of Figures	iii
ABSTRACT	iv
Abbreviations and Acronyms	iv
CHAPTER 1: INTRODUCTION	1
1.1 BACKGROUND	1
1.2 Problem Statement	3
1.3 Objectives	3
Main Objective:	3
Specific Objectives	3
1.4 Scope of Work	3
1.5 Project Organization	3
CHAPTER 2: LITERATURE REVIEW	4
2.1 Introduction	4
2.2 Technical Options	4
2.3 Gigabit PON	6
2.3.1 Comparison of GPON with other Technologies	6
2.3.2 Components of GPON	7
2.4 Using GPON	9
2.4.1. Triple play	9
2.4.2. GPON QoS (Quality of Service)	10
2.4.3. Dynamic bandwidth assignment (DBA)	10
2.4.4. GPON Encapsulation Method (GEM)	10
2.4.5. Multi-user ONT	10
2.4.6. Different user's profile	11
2.4.7. GPON as Metro network and ONT port as Uplink	11
2.4.8. Upgrade	11
2.4.9. Phone services	11
2.4.10. GPON in Greenfields and Brownfields	12
2.5. Network Modelling	12

2.6 Network Deployment Cost	13
2.7 Related Works	13
CHAPTER 3: DESIGN AND IMPLEMENTATION	14
3.1 Introduction.....	14
3.1.1. Design Tools.....	14
3.1.2 The Design.....	15
3.2 Picking User Inputs	17
3.3 Establishing Connections	21
3.3.1 Network Starting Point	21
3.3.2 Scratching the APIs.....	21
3.4 Network Plotting	23
3.5 Cost of Network Deployment	23
3.6 Process Flowchart	25
CHAPTER 4: RESULTS	26
4.1 Distance Calculations.....	26
4.1.1 Openrouteservice.....	26
4.1.2 Google Maps	27
4.2 Network Plot.....	28
CHAPTER 5: CONCLUSION AND RECOMMENDATIONS.....	32
5.1 Conclusion.....	32
5.2 Recommendations for Future Work.....	32
REFERENCES.....	33
APPENDIX	35
Appendix A: Project Code	35
APPENDIX B: Item Price List	78

List of Figures

Figure 1 Active Optical Network.....	2
Figure 2 Passive Optical Network.....	2
Figure 3 General Structure FTTx network.....	4
Figure 4 PON Evolution.....	5
Figure 5 Travelling of Light Energy in PON.....	7
Figure 6 Components of PON.....	8
Figure 7 An Optical Line Terminal.....	8
Figure 8 An Optical Splitter.....	8
Figure 9 An Optical Network Terminal.....	9
Figure 10 Macro view of a Typical City FTTx Network.....	12
Figure 11 Website Landing Page a.....	15
Figure 12 Website Landing page b.....	15
Figure 13 Website Landing page c.....	16
Figure 14 Website Home Page Footer.....	16
Figure 15 Log in page.....	17
Figure 16 Setting a Plot Name.....	18
Figure 17 Existing Plot Screen.....	18
Figure 18 Inputting Plot Coordinates for a Planning Session.....	19
Figure 19 Defining optional Network Starting point and point of integration with existing Network.....	19
Figure 20 Interface for adding Inventory Items.....	20
Figure 21 Inventory Page.....	20
Figure 22 Network Plan Result Page.....	28
Figure 23 Sample Designed Network.....	28
Figure 24 Plot with a Set OLT point.....	29
Figure 25 Changing the Plot Layout.....	29
Figure 26 Google Maps Street View.....	30
Figure 27 Starting Street View.....	30
Figure 28 Network Deployment Cost.....	31

ABSTRACT

The advancement of communication systems over the years has greatly enhanced human communication. With the advent of the internet in 1983, there was need for even greater connectivity and faster communication channels with greater bandwidths and throughput. Today, the demand for internet bandwidth is doubling every year. As a result, fiber continues to be preferred for telecommunication systems as it provides exponential bandwidth capabilities.

FTTx (Fiber to the X) provides a reliable future proof voice, data and video traffic transmission method. Reduced consumption of electrical energy and higher bandwidth, which translates to higher data rates are its main advantages.

The latest construction, connection and techniques of transmission can be fully leveraged by bringing fiber connections closer to the end user. This effectively eliminates the bottlenecks of the traditional copper cable.

Skillful planning and thorough execution are what it takes to make this possible.

In this project, a platform for planning and documenting FTTx network is developed. The platform develops a quick network design that provides shortest connection from the OLT to the customer premises.

Abbreviations and Acronyms

FTTx	Fiber to the X
AON	Active Optical Network
PON	Passive Optical Network
APON	ATM PON
EPON	Ethernet PON
GPON	Gigabit PON
BPON	Broadband PON
CAPEX	Capital Expenditure
OPEX	Operational Expenditure
ORS	Openrouteservice
ATM	Asynchronous Transfer Mode
ROI	Return on Investment
OLT	Optical Line Terminal
ONU	Optical Network Unit
ODU	Optical Distribution Network
QoS	Quality of Service
PoE	Power over Ethernet
DSL	Digital Subscriber Line
CO	Central Office

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

FTTX (Fiber to the X) represents the technology aspirations and ambitions of today's businesses that seek reliable technology investment guidance and reasonably assured Return on Investment (ROI). It is a means of providing Next Generation Access (NGA) to corporates that rely on a robust IT communications mechanism to get more done, faster. Homes are equally taking up fiber for fast and reliable internet connection.

FTTX refers to any of the following possibilities:

- FTTN - Fiber to the Node
- FTTB - Fiber to the building
- FTTH - Fiber to the Home

High bandwidth requirements including enterprise cloud computing applications, virtualization, audio and video conferencing, high definition 4K streaming, Intelligent devices and IoT/AI, SaaS applications, remote and work from home approaches, surveillance applications, mobile traffic drive businesses and individuals to invest more in bandwidth.

For this, fiber is strongly trending and future proof. Compared to copper, it stands out in the following ways:

- High operational safety
- Lighter, faster and longer-life
- Excellent Capex ROI
- Easier to install
- Ideal long-haul characteristics
- Highly reliable and secure
- Lower energy consumption
- High electrical, RF noise immunity

FTTX is implemented as either Active Optical Network (AON) or Passive Optical Network (PON).

In **Active Optical Network**, the point-to-point (PTP) architecture is used. Every client gets a dedicated fiber line. Switching equipment like routers and switches and active devices are deployed from the central office to the user distribution unit when transmitting signals.

Electricity is used to drive the switching equipment to manage signal distribution and direction to and from the customers. Light sources (lasers) and optical transceiver modules are examples of active equipment used.

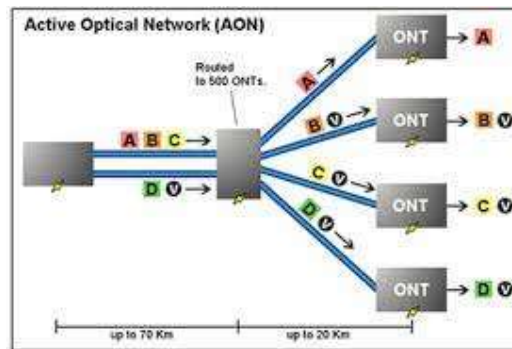


Figure 1 Active Optical Network

Passive Optical Network uses the point to multipoint architecture, and is more widely deployed. Optical fibers and passive components are used, while active equipment is only used at the signal source and receiving end. The optical splitter is the core and separates and collects signals passed through the network.

The splitters for PON are bidirectional. More than one service, for instance IP data, voice, and video are distributed by the Optical Line Terminal in the downstream direction through broadcast. They are split through the 1: N passive optical distributor in the Optical Distribution Network to every other Optical Network Units (ONU) on the PON. Upstream, service information from each ONU is coupled to the same fiber through the 1: N passive optical combiner in the ODN. This happens without any interference. The signals are then sent to the Optical Line Terminal.

In PON an optical line terminal is installed at the central control office, while the receiving end has a set of supporting ONUs (optical network units). The optical distribution network (ODN) between the OLT and the ONU contains optical fibers and passive splitters or couplers. There are three technical PON standards.

These are Asynchronous Transfer Mode PON (APON), Ethernet PON (EPON) and Gigabit PON (GPON). GPON applies General Frame Protocol.

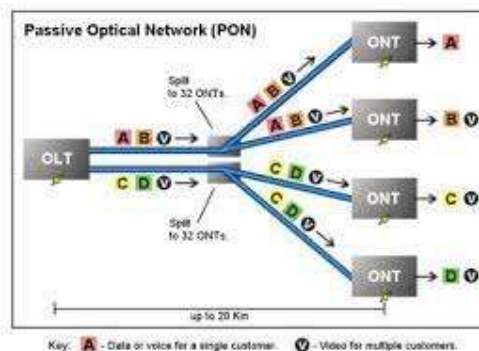


Figure 2 Passive Optical Network

In AON, each client has a dedicated fiber line. Consequently, network upgrade, maintenance and capacity expansion can be done with more ease. It has a wider coverage of one hundred kilometers. PON, on the other hand goes for up to 20 km. In Active PON, optical signals are guided using active equipment, while in passive networks passive devices are used. As such, the costs of deploying AONs is much higher than that of a PON.

In this project, a PON FTTX network planning and documentation platform that ensures good CAPEX returns and meets user communication needs is designed.

1.2 Problem Statement

In FTTX network, the success and consequently the profitability of network technology heavily depend on the topology and its implementation. As such, theoretical knowledge has to be employed in coming up with highly optimized networks. This ensures the cost of deployment is low, and that the network topology can be implemented.

This project focuses on designing a platform for the planning and documentation of PON FTTx, paying regard to deployment cost minimization while considering operational issues.

1.3 Objectives

Main Objective:

To design a PON FTTx planning and documentation platform

Specific Objectives

- To reduce time and effort spent in FTTX network design
- To plan a practically implementable PON FTTX that ensures optimum ROI, and is easy to maintain.
- To generate accurate network implementation cost.
- To provide reliable network documentation.

1.4 Scope of Work

In this project, the geographical data or existing access network topologies are taken as inputs and the optimum network plan formulated and presented as a graph. The optimum costs are also determined and presented.

1.5 Project Organization

The subsequent sections of this project are organized as follows:

- Chapter 2 Literature Review and Related Work,
- Chapter 3 Design and Implementation of the project,
- Chapter 4 Sample Results and discussion,
- Chapter 5 Conclusion and Recommendation for Future Work.
- Finally, the References and Appendix are presented.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

FTTx is an essential component of NGA (Next Generation Access) networks. The evolution of broadband network infrastructure towards enhanced speed and quality of service (QoS) is the key focus of NGA [1].

With FTTx, we get higher bandwidths and consequently faster connections accompanied with reduced energy consumption. The latest transmission techniques and network construction can be fully taken advantage of by moving fiber closer to the user.

In PON, a high bandwidth connectivity is provided. End users are grouped together based on location and sales level agreements and connected via splitters to form a network. Such a network is served by an OLT at the operator's Central Office.

The splitter merges and splits optical light. It serves as a distribution point from which end users are connected to the Optical line terminal.

An optical network unit is placed at the end user premises. It provides an interface between the user's equipment for data, audio, video and TV traffic and the optical fiber network [2].

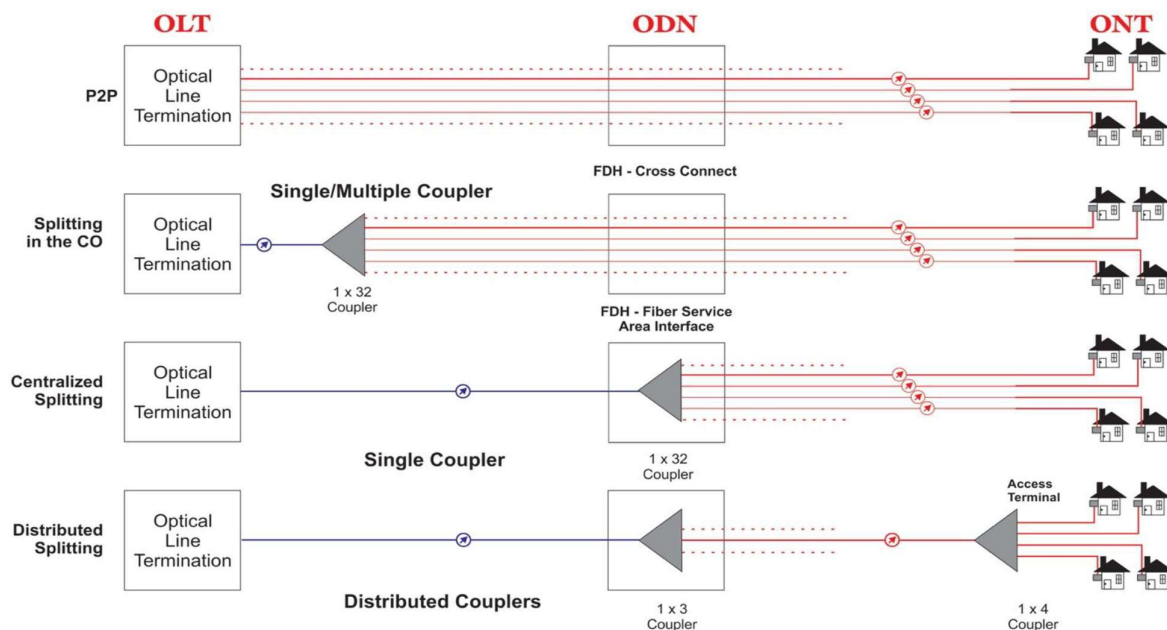


Figure 3 General Structure FTTx network

The low attenuation (0.2–0.6 dB/km) of single mode fiber cables and the accompanying bandwidth, greater than 30,000 GHz are taken advantage of by FTTx. This effectively puts it ahead of the competition compared with other broadband technologies.

2.2 Technical Options

The Optical distribution network employs passive equipment between the OLT and the end user ONU.

PON technology has developed from ATM-PON, Broadband PON, to later EPON and GPON. The modes of transfer employed vary. These are based on standards that were developed in different periods. It is however noteworthy that in essence, the theoretical capacity at the optical level remains the same. The electrical overlay, which is the protocol used in the allocation and management of capacity in the connection limits the bandwidth capability. [5].

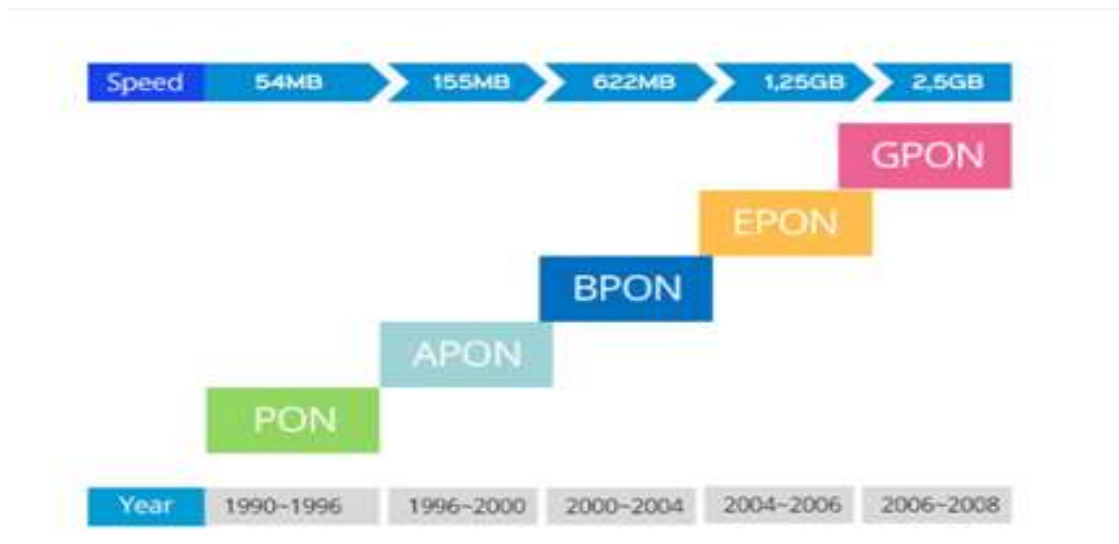


Figure 4 PON Evolution

APON was the first Passive Optical Network system that was developed and deployed commercially. It relied on cell switching protocols (ATM) for packet communication. APON was first proposed by the ITU (International Telecommunication Union) in the 1990s. Centralized and statistical multiplexing ATM abilities were exploited alongside the sharing effect of passive splitters. Effectively, its cost was about 20 to 40% lower than circuit switching based networks.

As Ethernet technology rapidly developed, ATM PON was phased out. BPON, an enhancement of the APON was proposed. Broadband Passive Optical Network is based on ATM and provides an uplink of 155 Mbps and 622 Mbps upstream. Additionally, it brought along dynamic bandwidth allocation and network protection.

PONS based on Ethernet technology are the successors of the two Asynchronous Transfer Mode Technologies. EPON technologies provide higher bandwidth and give better ROI. They are based on Ethernet PON technology, bringing along the good sides of both PON and Ethernet technologies. The point to multipoint network structure is adopted and multiple services are available on Ethernet.

After Ethernet PON came GPON. GPON comes with wide speed offers, starting from 622 Mbps symmetrical. This implies the upstream and downstream capacities are similar. There is also the asymmetric 2.5 Gbs download and 1.25 Gbs upload speeds. Gigabit PON is hybrid and employs ATM for voice transport, while data transport uses Ethernet. It is widely used in FTTx networks [6].

Table 1 summarizes Time Division Multiplexing TDM PON technologies standards with their important parameters [3].

Parameter	Broadband PON	Ethernet PON	Gigabit PON	XGPON	10G-EPON
Standard	ITU-T G.983	IEEE 802.3ah	ITU-T G.984	ITU-T G.987	IEEE 802.3av
Download Rate	622 Mbps	1.25 Gbp/s	2.5 Gbp/s	10 Gbp/s	10 Gbp/s

Upload Rate	155 Mbps	1.25 Gbp/s	1.25 Gbp/s	2.5 Gbp/s	10 Gbp/s/Symmetric 1 Gbp/s/Asymmetric
-------------	----------	------------	------------	-----------	--

For TDM PON, a major downside is that operators cannot share the same fiber cable. As such, multi fiber deployment has to be used to make it possible to share the access network [4].

In the access networks development, Wavelength Division Multiplexing PONs are the next big step. Study Group 15 (SG15) of the International Telecommunication Union–Telecommunication Standardization Sector (ITU–T) is studying two flavors of WDM-PON.

TWDM (Time Division and Wavelength Division Multiplexing) PON is the first. It allows 4-16 wavelengths on the one fiber cable, effectively supporting a higher number of users and greater data speeds. It also allows several operators to share the same cable, each using a different wavelength.

AWG (arrayed waveguide grating) is the second. The goal here is to provide each customer with a dedicated wavelength, just as it is in point to point networks. The data capacity is 1.25 Gbp/s symmetrical [4].

2.3 Gigabit PON

2.3.1 Comparison of GPON with other Technologies

Compared to EPON, GPON has the advantages of higher bandwidth and support for multiple services. On the flipside, it is a more complicated technology hence a higher cost. The technology choice depends on the cost of optical fiber access and service requirements. For high bandwidth requirements, GPON is more viable. Future GPON technologies are set to have even higher data rates. An example is the 10 G EPON/10 G GPON based on EPON/GPON technology, and presents an even higher bandwidth [6].

GPON can be compared with other similar technologies from two perspectives, economic and technical.

From an *economic perspective*, a large number of papers and presentations present the economic dynamics of GPON in relation to other broadband solutions. It is generally agreed that DSL technology is less expensive compared to GPON (the price of ADSL2+ modem is, for instance up to ten times less expensive than a GPON modem). This though, is not the only factor in economic analysis; there are also price of port at DSLAM and OLT price and price of outside cable network [7].

DSLAM and OLT port prices do not vary significantly. In some areas, DSL technology has the upper hand since copper cables are already widely deployed. Fiber networks will need to be installed anew, which is more expensive. As such, GPON is seen to be especially sound for green field application. This notwithstanding, today fiber cables are getting deployed more, and gaining a wider coverage.

From a *technological perspective*, the only copper-based technology in last mile connection is VDSL2. The main competitions to GPON from other technical perspectives are other FTTx technologies.

When the GPON is compared with GEAPON, it has the upper hand as up to 64 customers can be served with a maximum downlink of 2 GB per second per port. GEAPON can only serve 32 customers with maximum downstream of 1 GB per second. As for the data rates, the true

competitor for Gigabit PON is Active Optical Network technology (100M and 1G). A major characteristic of GPON is GPON Encapsulation Mechanism (GEM). This gives it an edge over the other access technologies. Because of GEM mechanism, the efficiency of GPON compared to ATM is 30% larger. Useful traffic level for GPON is around 93% and just about 73% for ATM.

GPON thus has the upper hand when compared with other broadband technologies in terms of Quality of Service and capability of differentiating services offered to each customer [7].

2.3.2 Components of GPON

GPON, like the other passive optical networks uses the point to multipoint architecture. Passive optical splitters are used to split a signal from one fiber line to multiple premises, up to 64. WDM (wavelength division multiplexing), is leveraged, and one wavelength is used for downstream traffic while another is used for upstream traffic on a single Non-zero dispersion-shifted fiber (ITU-T G.652).

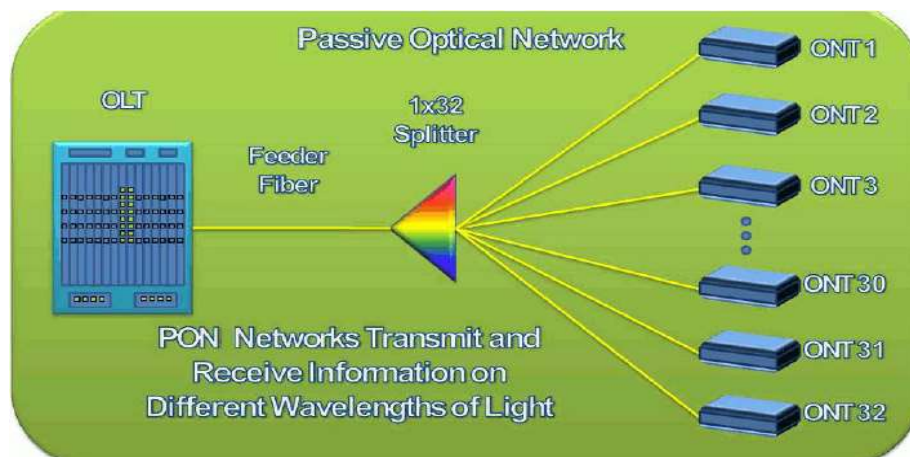


Figure 5 Travelling of Light Energy in PON

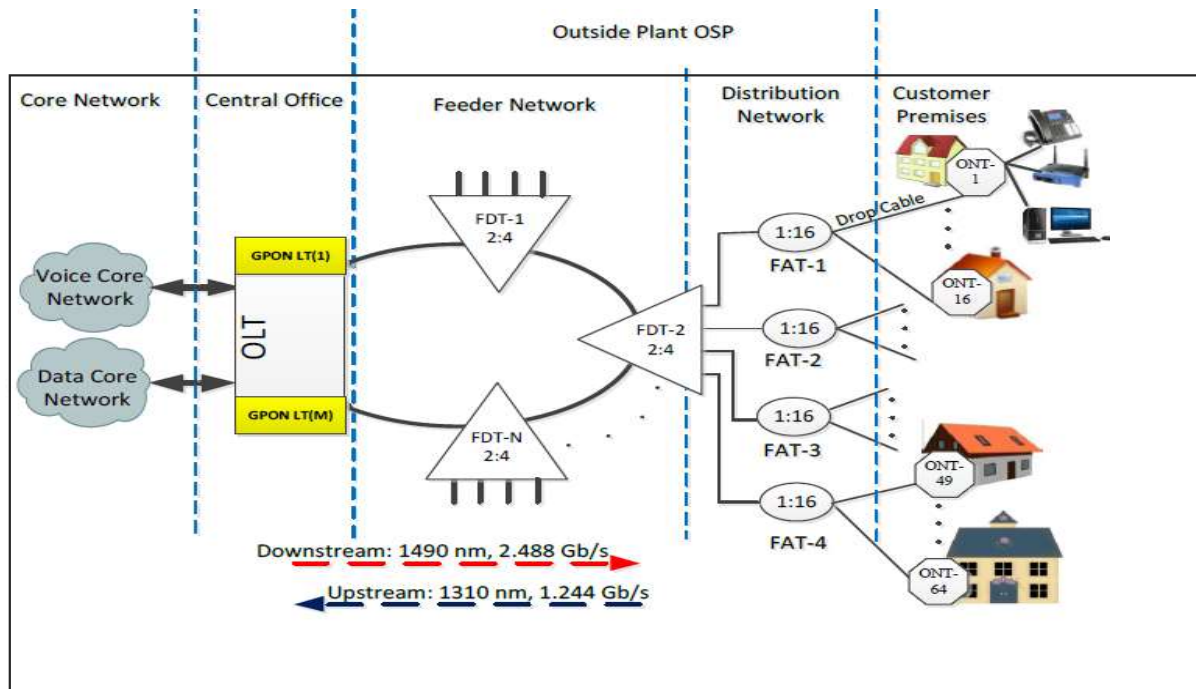


Figure 6 Components of PON

i. Optical Line Terminal OLT



Figure 7 An Optical Line Terminal

The Optical Line Terminal (OLT) is the main component of the network and is placed in the Central Office of the operator. It is the engine driving the FTTx network. Bandwidth allocation, scheduling of traffic and buffer control are the most significant functions of the OLT. They usually run on redundant DC power (-48VDC) and have at least one Line card for incoming traffic, a System Card used for on board configuration, and one to many GPON cards. Each GPON card has several GPON ports. [8]

ii. Optical Splitters



Figure 8 An Optical Splitter

The optical splitter splits the power of the signal. Each link (fiber) getting into the splitter may be split into a given number of fibers that leave it. Typically, there are 3 or more levels of fibers corresponding to 2 or more levels of splitters. This makes it possible for each fiber to be shared by

many users. As a result of power splitting, the signal gets attenuated but retains the initial structure and properties. A passive optical splitter should possess the following properties [9]:

- ✓ broad operating range of wavelength
- ✓ low insertion losses which should be uniform
- ✓ have minimal dimensions
- ✓ highly reliable
- ✓ support network survival

iii. **Optical Network Terminal ONT**



Figure 9 An Optical Network Terminal

Optical Network Terminals (ONTs) are deployed on the end user side. They connect to the OLT through fiber cables without any active elements in the link. The ONT transceiver is the physical connection between the end user premise and the central office OLT. WDM triplexer module separates the 3 wavelengths 1310nm, 1490nm and 1550nm (for CATV service). ONT receives data at 1490nm and sends burst traffic at 1310nm. Analogue video is received at 1550nm. Media Access Controller (MAC) controls upstream burst mode traffic and ensures no collisions occur as a result of upstream data transmission from the different users [9].

There are fiber to copper media converters offering RJ11, RJ45, and F-Series device connectors. The devices are available in different configurations and port densities ranging through to 24 ports.

There are ONTs for outdoor and those indoor installation, providing POE or no POE, 10/100/1000, AES encryption. They may come with batteries to survive in case of power outage.

GPON employs DBA (Dynamic Bandwidth Allocation). Bandwidth is dynamically allocated depending on the number of packets available. The OLT reads the number of packets waiting in T-CONT and performs bandwidth assignment. In case no packets are waiting in T-CONT, another T-CONT is assigned the bandwidth. For an ONT with a long queue, the OLT allocates multiple T-CONTS [10].

2.4 Using GPON

2.4.1. Triple play

Triple play service stands for 3 services, that is:

- ✓ voice,
- ✓ internet and
- ✓ video

As pointed out in [11], triple play service transfer are of different types. All of them can, however be classified into 2 based on the type of triple play service offered: standard, without HDTV (High Definition TV) and with HDTV. HDTV users can be satisfied with up to 10 Mbps, while more than 15 Mbps is needed by the HDTV users. GPON and its PON tree structure (class B+, 64 users, 2.5 Gbit/s upstream) makes it possible for each customer to get 37 Mbps when all the users take up all the available bandwidth. Usually, not all the users use all the available bandwidth. As such, a factor of 2x is used to create required bandwidth plans [12]. This makes GPON to effectively have more than enough bandwidth for triple play.

2.4.2. GPON QoS (Quality of Service)

2 mechanisms are employed for Quality of Service [13], DBA (Dynamic Bandwidth Allocation) algorithm and GEM (GPON Encapsulation Mode) with eight levels of priority. DBA helps in achieving control of upstream speed. Buffer control, another mechanism is employed for downstream. It is guaranteed that bandwidth with adequate priority can be achieved in both downstream and upstream. QoS can be achieved over each port on GPON ONT.

2.4.3. Dynamic bandwidth assignment (DBA)

DBA provides bandwidth management and sets different levels of priority. Through DBA, Gigabit PON is able to provide flexibly accommodate a wide range of services with without compromising the Quality of Service. The bandwidth capacity can be split to four bandwidth classes, with range starting from best effort to fixed. Each category has its unique level of priority depending on SLA (Service Level Agreement) between the network provider and the customer.

Bandwidth allocation depends on its category, with fixed bandwidth, which suited to static bandwidth management allocated cyclically. The other bandwidth categories are dynamically allocated by the DBA (Dynamic Bandwidth Allocation) function and distributed among different ONTs and among the different traffic flows from each ONT. As such, DBA effectively improves Quality of Service and makes it possible for GPON to use bandwidth effectively.

2.4.4. GPON Encapsulation Method (GEM)

GEM is the unique mechanism for transport in GPON that allows service differentiation. It has a similarity with ATM, as it adds a 5 bytes header, while the maximum size of payload that can vary up 4095 bytes (12 bits). According to [14] its downstream efficiency is 94% while the upstream efficiency is 93%. It also allows for very efficient Ethernet transport over GPON networks, as well as transport of other services like TDM.

Using specific GEM mechanism, OLT can be associated with different traffic flows. This is to say that GEM can be seen as a virtual port identifier which is highly significant for any Gigabit PON service. Traffic coming from each physical port can be placed into up to eight queues after traffic flow classification.

GPON owes its great flexibility and great bandwidth capabilities to GEM.

2.4.5. Multi-user ONT

Each GPON ONT port can get bandwidth with separate Services Level Agreement; and so each port can effectively serve deferent customers. Those that require multiple ports use a mini router.

In a residential set up, for instance, the ONU can be connected with optical fiber to GPON OLT, and the residential households each connected via mini routers with UTP cables to the ONU.

2.4.6. Different user's profile

The DBA, Buffer control and GEM makes of creation of different user's profiles possible over every physical port or for each service. As such, GPON has the ability of creating user profiles that are essentially unique for each customer based on the required services. Therefore, bandwidth can be allocated between several users in a PON network tree depending on what they pay or need.

The remaining bandwidth can be redirected for use in other network services.

2.4.7. GPON as Metro network and ONT port as Uplink

As seen earlier for class B+ for ODN, each of 64 users can use around 35 Mbps. For triple play service, 15 Mbps is needed. Since not all users use full triple play services at the same time, it can safely be assumed around 2 Gbit/s of bandwidth is almost always free. This redirected and employed for uplink for mini DSLMs, or for Business operators, or another entity with great bandwidth requirement. Factoring this in combined with the default properties of a GPON's protection mechanisms in case of malfunction [15] and possibility of extension up to 60 km, the conclusion that GPON can cover a region very efficiently can be made.

2.4.8. Upgrade

Considering the development of services, and ever-increasing demand for higher data rates and more reliable connection, there are reasons for concerns that GPON 2.5Gbit/s of bandwidth will be overwhelmed by future needs. Presently the speed of 2.5Gbs is adequate, but to be ready for the future demands, there are debates about NG GPON (Next Generation GPON) that will have speed of up to 10Gbs both in upstream and downstream. Currently there is a solution called 10-Gigabit-capable passive optical network (XG-PON) that was standardized in 2010 by ITU-T with Recommendation G.987.1, 2 and 3.

XG-PON is a translational solution between the GPON and Next Generation GPON. Its speed capability is 10Gbit/s and 2.5Gbit/s in the downstream and upstream respectively. Development of next generation network is projected to follow the trend GPON 2.5Gbit/s downstream and 1.2Gbit/s upstream, followed by XGPON 10Gbit/s and 2.5Gbit/s downstream and upstream respectively. Eventually, we hope to get to NG-PON offering 10Gbit/s for both downstream and upstream traffic. A PON upgrade advantage is that it can be performed as per the desire of each user. As such, those requiring higher bandwidth can be upgraded with the rest of the network users left intact.

2.4.9. Phone services

GPON gives telecommunication firms an opportunity to provide new services and improve existing ones. A common telecommunication services supported by GPON is phone service. 3 methods of phone service are supported by Gigabit PON:

- ✓ V.5.2
- ✓ SIP and
- ✓ H.248

Almost all the equipment used can easily switch between the options. As such, GPON is very reliable and has a great level of flexibility as seen in literature [16].

2.4.10. GPON in Greenfields and Brownfields

For a long time, FTTx was mainly used in Greenfield projects. Here, existing infrastructure is a non-issue, and communication networks are being laid for the first time.

Brownfield projects are, on the other hand restricted by existing network infrastructure. Typically, FTTx is installed where the services over present infrastructure is of very low quality.

Today this is changing and Fiber networks are gaining a wider coverage.

2.5. Network Modelling

Graphs are used as a formal representation for network design and topology planning. In the network construction, road systems and other routing information like railway lines and rivers determine potential network paths. A suitable location of the central office, places that can be used for network splitters, road crossing and user premises are also identified.

The resulting links and nodes form a graph representing the network map. The access network topology is an overlaying graph connecting customer premises to the central office, and meeting set specific restrictions. Splitter nodes, each of them connecting up to 64 customers exist in the network. The splitters are then linked to the Central Office. The network linking the Central Office to the splitters is known as the feeder network, while the outside region between splitters and customers is the distributive network [19].

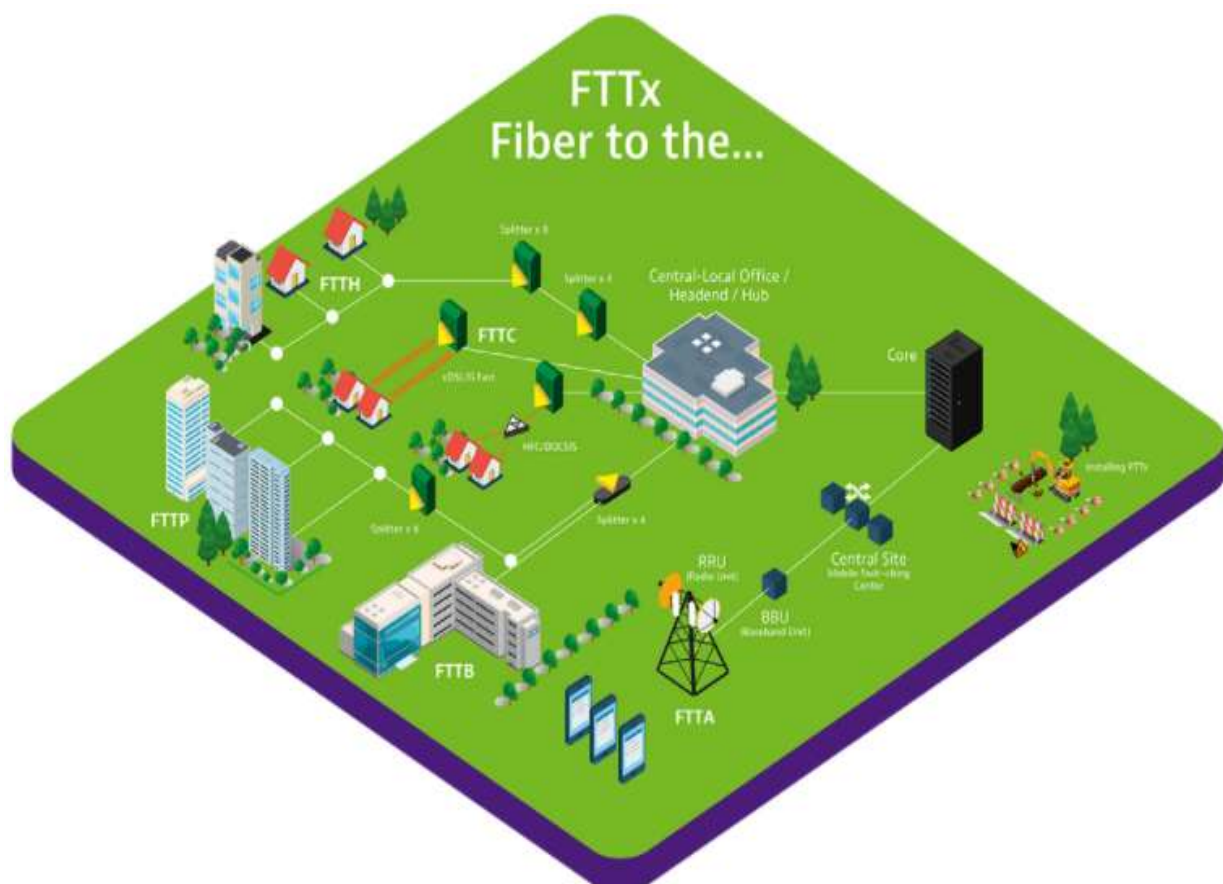


Figure 10 Macro view of a Typical City FTTx Network

2.6 Network Deployment Cost

The costs involved in setting up an FTTx network are the cost of the optic fiber equipment, cable plant and the amount spent fiber length:

$$\text{Network Cost} = C_{\text{Equipment}} + C_{\text{Cableplant}} .$$

To determine the equipment cost, the number of each type of equipment used, and the unit price of each should be known.

The price of the OLT and the splitter are handled together. Both mean a fixed cost for any network.

Cable plant cost is a little more complicated, since typically a bunch of connections are set in parallel, and are joined within the same cable. Such a cable may have 8, 12, 24, and so forth fibers.

As such, the cost model for a used network link consists of two parts, C_0 that has to be paid once, regardless of the amount of connections over the specified link, and C_v that depends on the connection cost.

The “feeder” and “distributive” part of the network can also be handled separately. Different type of cables can be used, and equally different techniques are used for cable deployment. As an example, for the last segment to the customer, aerial cables may be used. As such, the total cable plant cost can be described as [19]:

$$C_{\text{Cableplant}} = C_{\text{feed}} + C_{\text{dist}}$$

$$C_{\text{feed}} = \sum_{e \text{ Feeder}} (C_0^{\text{feed}} + \sum C_v^{\text{feed}})$$

$$C_{\text{dist}} = \sum_{e \text{ Dist}} (C_0^{\text{dist}} + \sum C_v^{\text{dist}})$$

2.7 Related Works

In [7], the authors present Gigabit Passive Optical Networks in a relation to New Generation Networks. It is seen that GPON is not only economically sound, but also an effective, and very reliable solution for triple play service. GPON is thus very viable technically and economically for NG Networks. The mechanisms employed to implement GPON applications such as GEM, DBA algorithm and network protection schemes are well pointed out. The paper also presents economical and technical aspects of GPON as compared to other broadband options.

A stepwise design and field deployment of a Gigabit PON FTTx network connecting up to 1000 customers is presented in [18]. Basic network elements are highlighted and the contribution of each and its role in the network architecture described. Class B protection is factored in to provide a redundancy that effectively provides an alternative data path when the main link fails.

The efficient planning for PON networks and the part it plays in ensuring the network is deployed economically is presented in [20]. An approach for PON network deployment, which can be scaled, is developed. Compared to benchmark models, it can reduce the deployment cost by up to 70% for designs with hundreds of ONUs.

The authors in [19] address broadband optical access network design with the aim of reducing the cost of setting up the network. The operational factors and other user considerations and cost parameters are also taken into account. A heuristic solution that works fast even for big networks is developed. The results from this solution vary with those from ILP computation by about 20%.

CHAPTER 3: DESIGN AND IMPLEMENTATION

3.1 Introduction

In this section, the focus will be on the methodological approach used to design the GPON Fiber network planning platform.

3.1.1. Design Tools

The work was implemented using Python programming language and presented as a web app developed using the Django framework. Flask was picked initially but dropped as Django was deemed better. Compared to Flask, Django provides a full featured Model-View-Controller framework, offers dynamic HTML pages, supports Object Relational Mapping and provides a useful admin interface.

Additionally, HTML and CSS were used to enhance the presentability of the webpages.

SQLite, the default Django database is used in the design.

The network design is determined and presented as a plot on a map. The following mapping APIs were considered:

1. OpenStreetMap

OSM (OpenStreetMap) is a project powered by the community and is supplying map data to many websites and apps. Since it is open-source, no charges accompany its usage. A high level of accuracy of the map data is maintained as a result of combined efforts of community map enthusiasts who add data to it and support it. In Mapbox OpenStreetMap data is used for a backbone. Google Maps also verifies and incorporates data from OSM.

OSM has the advantage of being open source and free.

On the flipside, its API has a very limited functionality. This is because primarily, its purpose was just to update map data. The maps are also not very appealing to look at.

2. Mapbox

It was founded in the year 2010 with an aim of giving non-profit based, environmental and humanitarian organizations mapping data. Over the years, it has become a fully-fledged giant in the mapping industry. Today, it provides mapping data to large organizations like Facebook while still keeping the needs of local entrepreneurs in mind.

A very elaborate features-set and integrating tools are provided which makes it easy to be used by developers. It also offers a great range of functionalities.

Mapbox offer a great deal of customization and flexibility, have a fast load time accompanied with great performance and have maintained the open-source approach. It releases its code which the community is encouraged to check out and make better.

Mapbox however has inferior coverage in certain regions and has very strict standardized data handling rules which may take developers some time to get used to.

3. Google Maps

Google Maps has well about 99% of the world map data. For some time, it has been the mapping pace-setter. Uber and Bolt are some of the large corporates that use its data.

Google Maps takes the advantage of its satellites, Street View cars, and Android devices, as well as feedback from local contributors, to provide large amounts of accurate map data with real time updates. Its product can meet the demands of any industry that needs to consume map data, but comes at a remarkable price.

Google Maps have an excellent global data quality and are inherently recognizable, making it easy to create a sense of trust with an audience in a web or app design. It also provides street view, a unique offering not provided by the rivals.

A major setback for Google Maps is it provides few options for customization. It is also not an open-source API.

In this project, the network design will be presented on a Google Map, while the routing information, including the network routes and distances will be fetched from openrouteservice which collects data from OpenStreetMap. The openrouteservice customization options lacking in Google Maps will be leveraged to optimize the design, while the extensive coverage and quality data of Google Maps will be leveraged to present an appreciable design.

3.1.2 The Design



Figure 11 Website Landing Page a

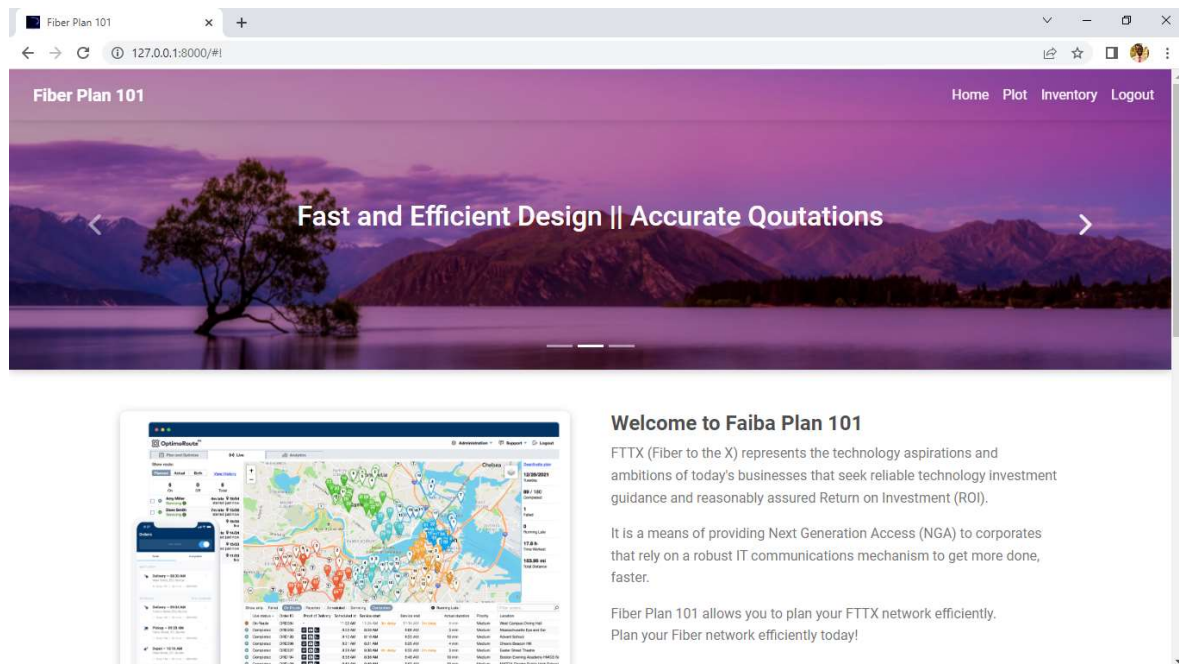


Figure 12 Website Landing page b

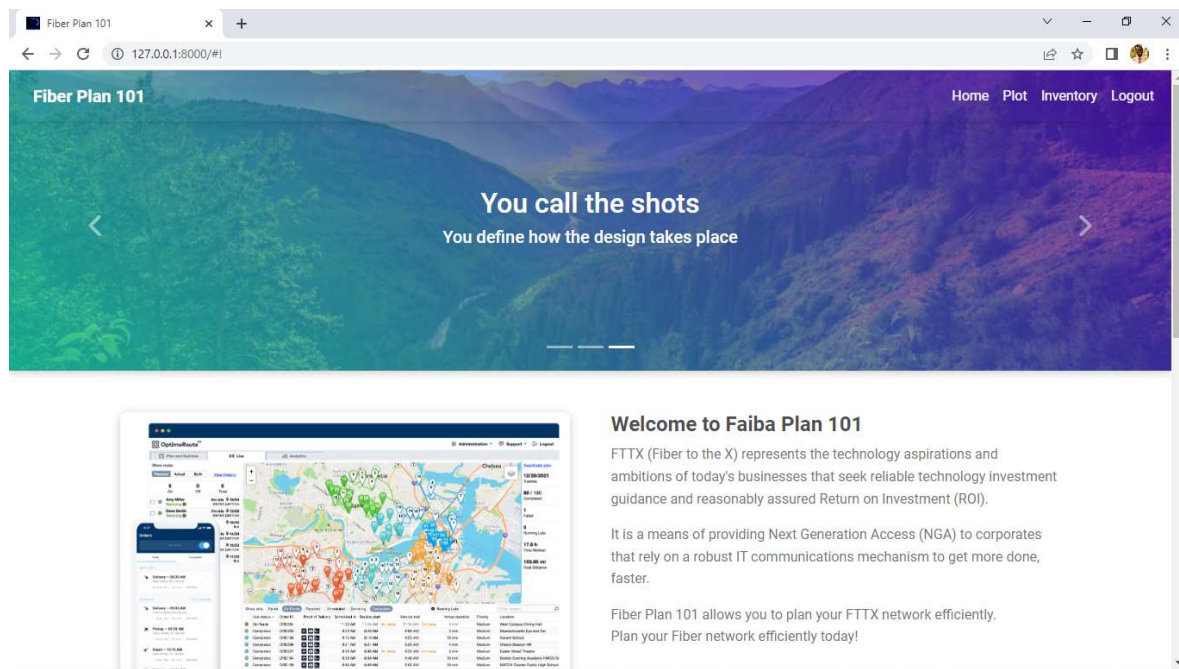


Figure 13 Website Landing page c

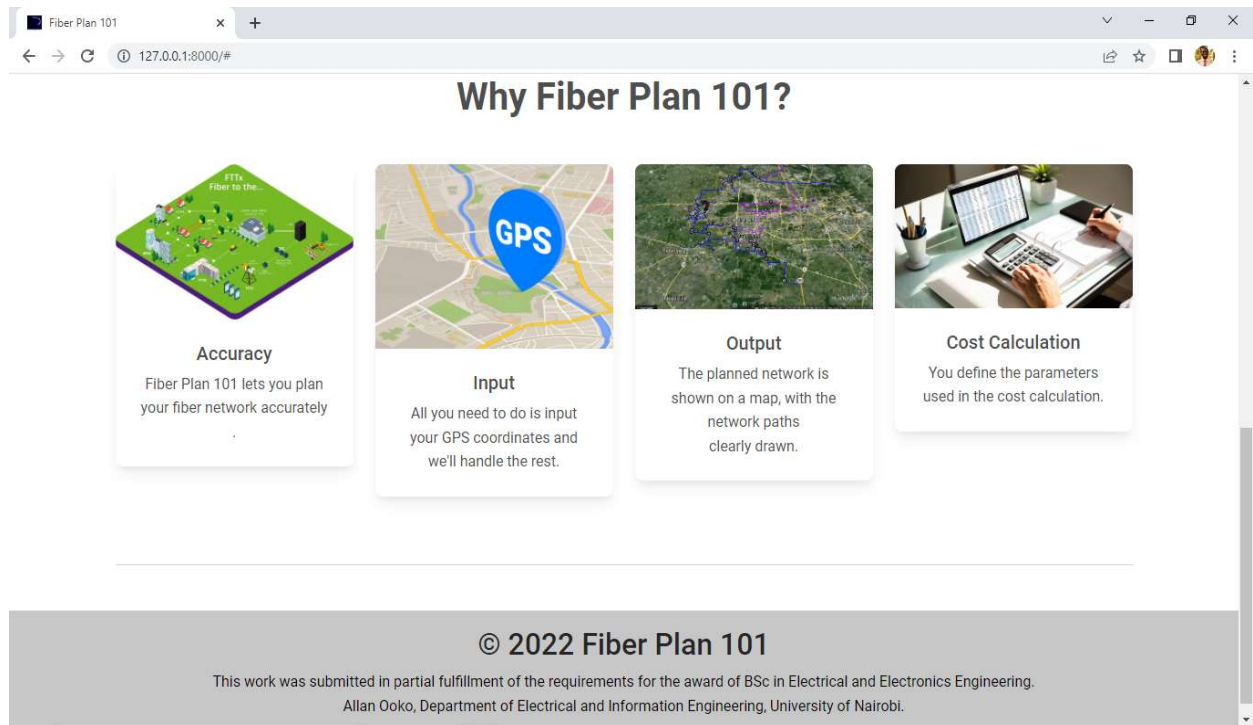


Figure 14 Website Home Page Footer

The network design process is divided into the following steps:

- i) Picking user inputs
- ii) Establishing Connections
- iii) Network Plotting
- iv) Cost Determination

For security, log in is required to start the planning process.

Figure 15 Log in page

3.2 Picking User Inputs

To start off the design process, a user is required to input the GPS Coordinates of all the points to be connected. Since GPON is used, the number of points entered should not be more than 64, and each point should not be more than 20 Km from the selected splitter location. The designer may choose to specify a point for the location of the OLT from which the rest of the points are connected to the network or use the point determined by the algorithm. A point to integrate the new network with the existing fiber network is also to be supplied, without which the location of the OLT is assumed to be the point of integration.

The system provides an inventory where details of items and equipment required to implement the network and their unit costs are specified.

Additionally, the following design parameters are also picked from the designer:

- Type of poles and distance between subsequent poles for aerial fiber deployment.
- Distance between manholes and handholes in underground networks.
- Distance after which extra fiber length is to be added, and the extra fiber length to be added.

For all these, default values picked from working standards are provided which the designer may choose to use, or change. This also applies to the costs of the items. Standard market prices are given, which the designer may choose to use, or provide alternative rates. This ensures the users of this platform have full control over the design process, while at the same time making the process not to be unnecessarily tasking.

These parameters are used to determine the cost of the network design.

To start off, a new plot name is input to start a new network plan, or an existing one selected to make changes.

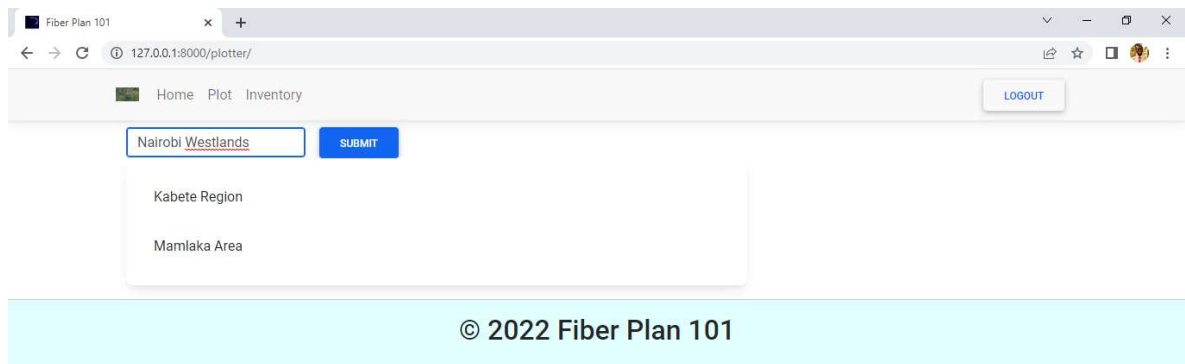


Figure 16 Setting a Plot Name

You may input a new plot name to start a new network planning session, or select an existing network plot to view its details or make adjustments.

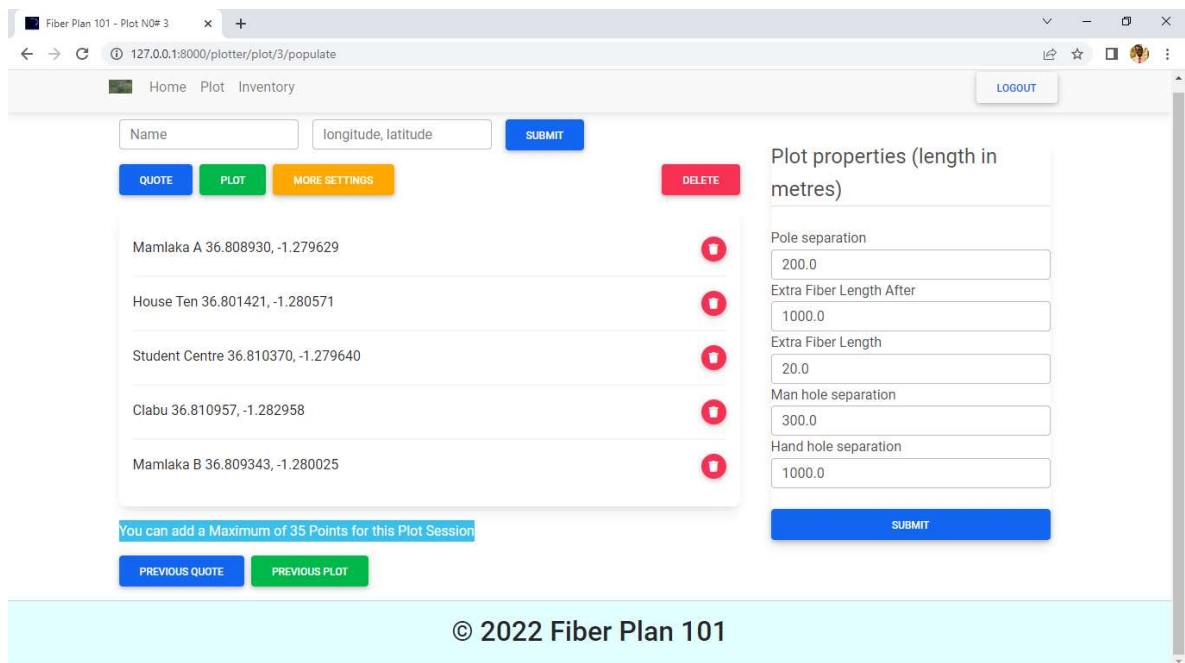


Figure 17 Existing Plot Screen

Figure 18 Inputting Plot Coordinates for a Planning Session

Since this project heavily runs on openrouteservice, which picks coordinates as longitude, latitude, the user is required to input the data in a similar fashion.

Figure 19 Defining optional Network Starting point and point of integration with existing Network

The set_start feature is optional. If selected from the existing coordinates, this will be used as the default starting point for the network design, and as such the system will not determine an OLT location.

The optional Integration Point is used to specify the point at which the network to be built is to be integrated into the existing fiber network.

Item

Sku*

Unit price*

Category*

SUBMIT

© 2022 Fiber Plan 101

Figure 20 Interface for adding Inventory Items

ADD ITEM

#	sku	category	unit price	Actions
1	Wooden Pole	PL	9000.0	
2	Concrete Pole	PL	10000.0	
3	8 Core Single Mode	FO	165.0	
4	Precast Handhole	HH	3000.0	
5	Masonry Manhole	HH	10000.0	
6	Precast Man hole	MH	4000.0	
7	15" Duct Diameter	DT	0.2	
8	40" Diameter HDPE	DT	0.63	

Figure 21 Inventory Page

3.3 Establishing Connections

3.3.1 *Network Starting Point*

The optimum network design is determined from the given inputs by running an algorithm to pick the point that gives the minimum distance to all the other points to be connected in the network. This is done by considering each point and calculating its road route distance to all the other points to be connected.

As indicated earlier, routing information used in this design is primarily fetched from *openrouteservice* which was developed by HeiGIT- Heidelberg Institute for Geoinformation and relies on OpenStreetMap for data. The servers however have an upper limit of 40 hits per minute. This greatly limits the number of points that can be looped through to determine the best connection.

As such, Google Maps directions API will be used to loop through the points to determine the best OLT location point. Google maps have no cap, but charge \$10 for every 1000 hits.

This is not likely to affect the design, as the distances between points do not vary significantly as will be demonstrated in the results section. Once the starting point is determined, it is plugged into *openrouteservice* and used for the rest of the computation.

This particular location is suggested as a primary starting point. The designer may use this, or set an alternative OLT location from which all the other nodes are to be connected to form the network map. The point of integration with existing network is also input, and the connection implemented.

Moreover, a redundant link to the point of integration is provided, which provides network security. In case of failure of the main link, the redundant connection provides an alternative path for flow of data.

3.3.2 *Scratching the APIs*

In both of the APIs used in this design, Google Maps and *openrouteservice*, the route information can be returned as json or geojson format. The *openrouteservice* geometry format for “json” is Google’s encodedpolyline.

a) Openrouteservice

For *ors*, the directions module will be used for both distance calculations and to fetch the route for plotting. An alternative module, distance matrix is available but will not be used since it allows less customization compared to the direction module. Notably, it does not take the parameter ‘continue_straight’ which avoids going around roundabouts, for this is not done in network design. Additionally, the directions API takes the parameter ‘preference’, allowing for the options ‘fastest’, ‘shortest’ and ‘recommended’. For navigation, we would be keen on taking the recommended or fastest path, but for the network design the shortest route will be used.

The following are some of the parameters the *ors* directions API takes.[22]

1. coordinates – The gps locations to be used in the route determination.
2. profile (string) – Here, the travel means is selected. An example is “foot-walking”
3. format (str) – States the format of the call response, eg ‘json’
4. preference (string) – States the routing preference to be used, eg shortest.
5. units (string) – Gives the units for distance values, eg ‘m’
6. geometry (boolean) – Specifies whether geometry is to be returned.
7. geometry_simplify (boolean) – Specifies whether to simplify the geometry data. Default False.
8. instructions (boolean) – Specifies whether to return turn-by-turn instructions. Default True.
9. alternative_routes (dict[int|float]) – Specifies whether alternative routes are computed, and parameters for the algorithm determining suitable alternatives. Expects 3 keys: share_factor (float), target_count (int), weight_factor (float).

10. `roundabout_exits` (boolean) – Provides bearings of the entrance and all passed roundabout exits. Adds the 'exit_bearings' array to the 'step' object in the response. Default False.

Below is a sample request to the openrouteservice directions API

```
coordinates=[(36.816156,-1.281715),(36.817268,-1.278819)]

Client = openrouteservice.Client(key='YOURAPIKEY')
route=Client.directions(coordinates=coordinates,
                        profile='foot-walking',
                        optimized=False,
                        preference='shortest',
                        continue_straight=True,
                        format='geojson')

print(route)
```

This return format is geojson.

The above call returns a Python dictionary type from which the distance between the two or more points can be picked. A demonstration of this will be done in the results section.

The route distance can be picked as follows:

```
route['features'][0]['properties']['segments'][0]['distance']
```

b) Google Maps

The Google Maps directions API takes the following parameters: [23]

1. origin: the gps coordinates to be used as a starting point.
2. destination: gps coordinate to be used as end point.
3. mode: the means of transport to be used, eg 'driving'.
4. waypoints: Gives points the route is to go through.
5. avoid: States features that are to be avoided in the rout
6. units: Specifies the unit system to be used, eg "metric"
7. param optimize_waypoints: Way points are reorganized in an optimised order or not.
8. transit_mode: Specifies one or more preferred modes of transit.This parameter may only be specified for requests where the mode is transit. Valid values are "bus", "subway", "train", "tram", "rail".
:type transit_mode: string or list of strings
9. param transit_routing_preference: Specifies preferences for transit requests. Valid values are "less_walking" or "fewer_transfers". type transit_routing_preference: string
10. traffic_model: Specifies the predictive travel time model to use. Valid values are "best_guess" or "optimistic" or "pessimistic". The traffic_model parameter may only be specified for requests where the travel mode is driving, and where the request includes a departure_time.

Below is a sample request to the Google Maps directions client from Python

```
gmaps=googlemaps.Client(key=YOUR_GMAPS_API_KEY)
route=gmaps.directions((36.816156,-1.281715),(36.817268,-1.278819),
mode="driving", units='metric')
print (route)
```

The return is equally a GeoJson.

The distance between the particular coordinate set can be picked form the disctionary as follows:

```
route[0]['legs'][0]['distance']['value']
```

This is then used in the computations.

3.4 Network Plotting

The network is plotted and presented as a map.

The determined OLT location, or the OLT location selected by the user is connected to all the other locations using the best map route, effectively completing network connection.

The routing information is picked from openrouteservice, and for the start point to each of the points, the resulting GeoJson file is assigned and converted to a google maps GeoJson layer and the layer added to the Google Map used to display the data.

The process is repeated till all the points are plotted, and the connection to the network integration point made. Equally, the redundant link connection is also made.

The code snippet that does this is shown below:

```
now=[(36.804896,-1.273139),(36.816253,-1.281417),(36.8159,-1.2795),(36.8219,-1.2921),(36.8259,-1.2850)]

Client = openrouteservice.Client (key=YOUR ORS API KEY)

route=Client.directions(coordinates=now,
                        profile='foot-walking',
                        optimized= False,
                        preference='shortest',
                        continue_straight=True,
                        format='geojson')

gmaps.configure(api_key=YOUR GOOGLE MAPS API KEY)
fig=gmaps.figure()

geojson_layer = gmaps.geojson_layer(route)
fig.add_layer(geojson_layer)
```

A Google Maps marker layer is then added. This layer indicates the locations of each of the clients, and the names assigned to the premises by the designer. The network integration point, redundant link and starting point are also rightly indicated.

The process is demonstrated below:

```
gmaps.configure(api_key='YOUR API KEY')
fig=gmaps.figure()
locations=((36.816156,-1.281715),(36.817268,-1.278819))
names='Ambank House', 'American Wing'
marker_layer = gmaps.marker_layer(locations,
                                   hover_text='Client',
                                   info_box_content=names)
fig.add_layer(marker_layer)
```

This is plotted, and the output presented as an interactive map. A layer control is also included, which allows the designer to choose the type of layout. Additionally, the powerful Street view feature of Google Maps allows one to see what the physical place looks like, and make decisions on the location suitability without having to go in person.

3.5 Cost of Network Deployment

The cost of greenfield deployment is calculated based on the size of the network and the specified design parameters.

The cost of fiber cable is determined by multiplying the required fiber length with the unit price of the selected cable. The same is done for the duct used in case of underground networks.

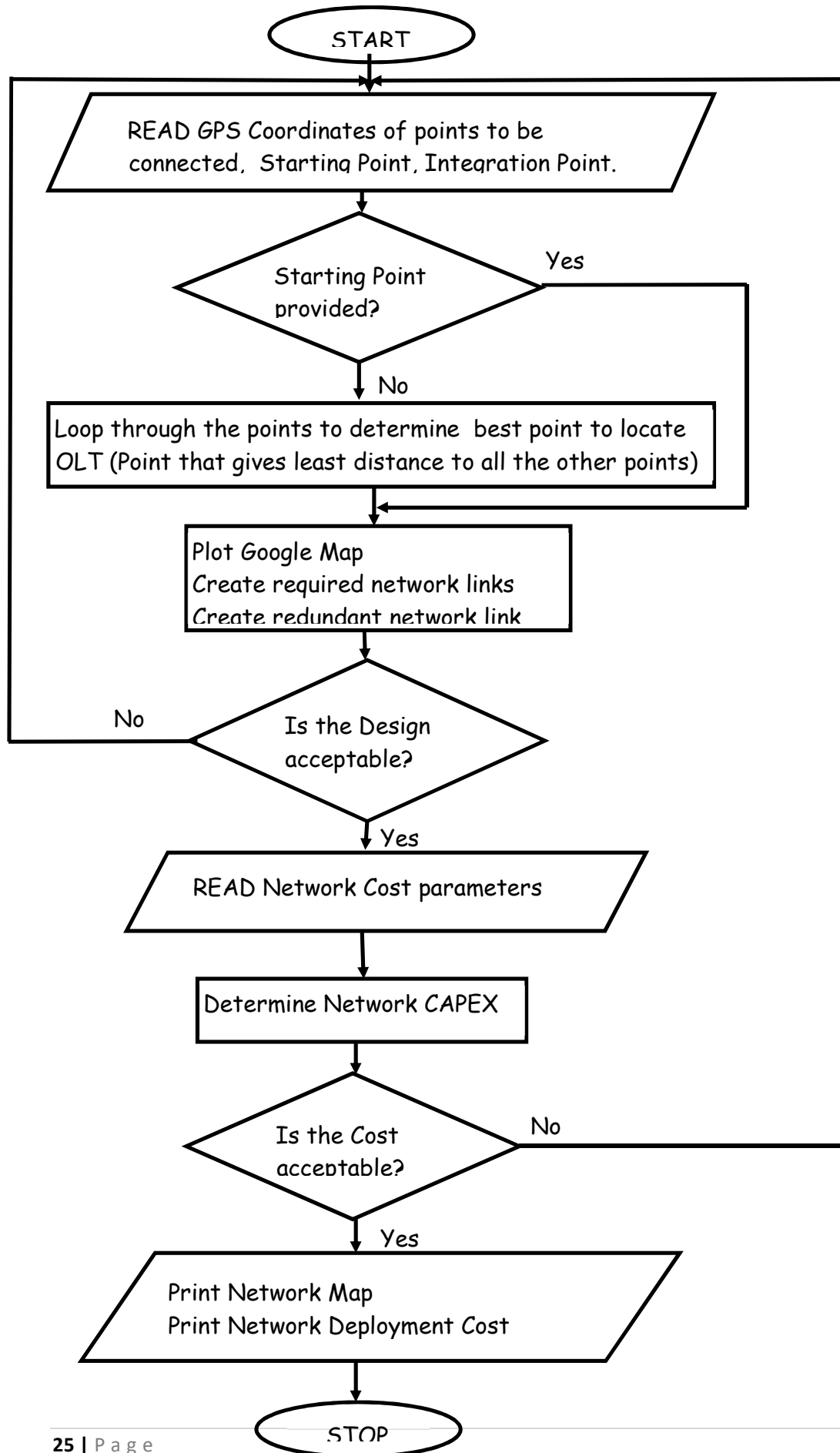
For the poles in aerial deployment, the unit price of the selected pole type is multiplied with the number of poles required, which is determined from the connection distance and the separation specified distance of separation between the poles. In a similar fashion, the cost of adding manholes and handholes to the network is calculated.

For ONUs, the total cost is calculated by multiplying the cost of each with the number of users to be connected. For aerial deployment, the capital required to add tension clamps is established by finding the product of the unit cost and the required number.

The costs are then summed up, to determine the network CAPEX.

Civil works and contractor service costs will not be incorporated, as these vary with the terms of contracts.

3.6 Process Flowchart



CHAPTER 4: RESULTS

In this chapter, the intermediate and final results of the network design process are presented.

4.1 Distance Calculations

4.1.1 Openrouteservice

The route distance between two points can be picked from the Python dictionary type returned from ors.

For the points (36.805763,-1.258764) and (36.802463, -1.263261)) i.e. the Nairobi Microsoft Office and KFC Westlands, using the ‘car-driving’ profile, the return is as follows:

```
{'type': 'FeatureCollection', 'features': [{'bbox': [36.802466, -1.263857, 36.806089, -1.258202], 'type': 'Feature', 'properties': {'segments': [{'distance': 892.9, 'duration': 116.2, 'steps': [{'distance': 19.2, 'duration': 6.9, 'type': 11, 'instruction': 'Head north', 'name': '-', 'way_points': [0, 1]}, {'distance': 16.3, 'duration': 5.9, 'type': 12, 'instruction': 'Keep left', 'name': '-', 'way_points': [1, 4]}, {'distance': 47.7, 'duration': 11.5, 'type': 0, 'instruction': 'Turn left onto Jalaram Road', 'name': 'Jalaram Road', 'way_points': [4, 5]}, {'distance': 452.5, 'duration': 38.6, 'type': 0, 'instruction': 'Turn left onto Ring Road Parklands', 'name': 'Ring Road Parklands', 'way_points': [5, 11]}, {'distance': 268.6, 'duration': 32.0, 'type': 7, 'instruction': 'Enter the roundabout and take the 2nd exit onto Ring Road Parklands', 'name': 'Ring Road Parklands', 'exit_number': 2, 'way_points': [11, 23]}, {'distance': 34.3, 'duration': 8.3, 'type': 6, 'instruction': 'Continue straight onto Ring Road Parklands', 'name': 'Ring Road Parklands', 'way_points': [23, 26]}, {'distance': 54.2, 'duration': 13.0, 'type': 0, 'instruction': 'Turn left onto Woodvale Lane', 'name': 'Woodvale Lane', 'way_points': [26, 29]}, {'distance': 0.0, 'duration': 0.0, 'type': 10, 'instruction': 'Arrive at Woodvale Lane, straight ahead', 'name': '-', 'way_points': [29, 29]}]}], 'summary': {'distance': 892.9, 'duration': 116.2, 'way_points': [0, 29]}, 'geometry': {'coordinates': [[36.805984, -1.258728], [36.80605, -1.258568], [36.806043, -1.258525], [36.806049, -1.258498], [36.806089, -1.258434], [36.805729, -1.258202], [36.805277, -1.258919], [36.805237, -1.258986], [36.805187, -1.259062], [36.805131, -1.25915], [36.804711, -1.259797], [36.803527, -1.261624], [36.803523, -1.261678], [36.803447, -1.262054], [36.803425, -1.262124], [36.803394, -1.262198], [36.803367, -1.262217], [36.803275, -1.262224], [36.803167, -1.262526], [36.803085, -1.26284], [36.80297, -1.263352], [36.80294, -1.263481], [36.802887, -1.263689], [36.802819, -1.263857], [36.802806, -1.263852], [36.802769, -1.26384], [36.802822, -1.263589], [36.802746, -1.26354], [36.802634, -1.263426], [36.802466, -1.263258]], 'type': 'LineString'}}], 'bbox': [36.802466, -1.263857, 36.806089, -1.258202], 'metadata': {'attribution': 'openrouteservice.org | OpenStreetMap contributors', 'service': 'routing', 'timestamp': 1653323573739, 'query': {'coordinates': [[36.805763, -1.258764], [36.802463, -1.263261]], 'profile': 'driving-car', 'format': 'geojson', 'engine': {'version': '6.7.0', 'build_date': '2022-02-18T19:37:41Z', 'graph_date': '2022-05-01T11:29:11Z'}}
```

From this, the distance can be extracted by running:

Route ['features'][0]['properties']['segments'][0]['distance']

returns the value 892.9, which is in meters.

The other components of interest in the dictionary may also be picked. For instance, route['features'][0]['geometry']['coordinates'] gives the coordinates of the places to be followed to walk between the two points. For these points, this returns:

```
[[36.805984, -1.258728], [36.80605, -1.258568], [36.806043, -1.258525], [36.806049, -1.258498], [36.806089, -1.258434], [36.805729, -1.258202], [36.805277, -1.258919], [36.805237, -1.258986], [36.805187, -1.259062], [36.805131, -1.25915], [36.804711, -1.259797], [36.803527, -1.261624], [36.803523, -1.261678], [36.803447, -1.262054], [36.803425, -1.262124], [36.803394, -1.262198], [36.803367, -1.262217], [36.803275, -1.262224], [36.803167, -1.262526], [36.803085, -1.26284], [36.80297, -1.263352], [36.80294, -1.263481], [36.802887, -1.263689], [36.802819, -1.263857], [36.802806, -1.263852], [36.802769, -1.26384], [36.802822, -1.263589], [36.802746, -1.26354], [36.802634, -1.263426], [36.802466, -1.263258]]
```

4.1.2 Google Maps

For the Google Maps directions API, feeding the same set of coordinates (Nairobi Microsoft Office and the Westlands KFC coordinates), the following dictionary is returned:

```
[{'bounds': {'northeast': {'lat': -1.2581891, 'lng': 36.8061055}, 'southwest': {'lat': -1.263391, 'lng': 36.8024664}}, 'copyrights': 'Map data ©2022', 'legs': [{'distance': {'text': '0.9 km', 'value': 924}, 'duration': {'text': '4 mins', 'value': 268}, 'end_address': 'Woodvale Lane, Nairobi, Kenya', 'end_location': {'lat': -1.2632577, 'lng': 36.8024664}, 'start_address': 'The Oval, Nairobi, Kenya', 'start_location': {'lat': -1.2588344, 'lng': 36.8058795}, 'steps': [{'distance': {'text': '51 m', 'value': 51}, 'duration': {'text': '1 min', 'value': 46}, 'end_location': {'lat': -1.2584332, 'lng': 36.8061055}, 'html_instructions': 'Head <b>northeast</b> toward <b>Jalaram Rd</b>', 'polyline': {'points': 'tztFwss_FIESKOIEA[0]', 'start_location': {'lat': -1.2588344, 'lng': 36.8058795}, 'travel_mode': 'DRIVING'}, {'distance': {'text': '51 m', 'value': 51}, 'duration': {'text': '1 min', 'value': 52}, 'end_location': {'lat': -1.2581891, 'lng': 36.8057161}, 'html_instructions': 'Turn <b>left</b> at Novel Technologies E A Ltd onto <b>Jalaram Rd</b>', 'maneuver': 'turn-left', 'polyline': {'points': 'dxtFeus_Fe@[IN]', 'start_location': {'lat': -1.2584332, 'lng': 36.8061055}, 'travel_mode': 'DRIVING'}, {'distance': {'text': '0.4 km', 'value': 447}, 'duration': {'text': '1 min', 'value': 56}, 'end_location': {'lat': -1.2615348, 'lng': 36.803494}, 'html_instructions': 'Turn <b>left</b> onto <b>Ring Rd Parklands</b><div style="font-size:0.9em">Pass by Fashion For Sight (on the left in 350&nbsp;m)</div>', 'maneuver': 'turn-left', 'polyline': {'points': 'tvtFwrs_FzAz@d@XLHXPt@`@VN~@h@b@Xj@Z\\RFBNHFB?@\\P|h@JHHDbBdA'}, 'start_location': {'lat': -1.2581891, 'lng': 36.8057161}, 'travel_mode': 'DRIVING'}, {'distance': {'text': '0.1 km', 'value': 123}, 'duration': {'text': '1 min', 'value': 28}, 'end_location': {'lat': -1.2625866, 'lng': 36.8032146}, 'html_instructions': 'Turn <b>left</b> at Dr Gohil Rajoo onto <b>Sarit Center Roundabout</b><div style="font-size:0.9em">Pass by Cop Bank ATM - Sarit Branch (on the left)</div>', 'maneuver': 'turn-left', 'polyline': {'points': 'pkuFyds_FBARCTBn@JNDJBDBFD`@Lh@J'}, 'start_location': {'lat': -1.2615348, 'lng': 36.803494}, 'travel_mode': 'DRIVING'}, {'distance': {'text': '0.2 km', 'value': 154}, 'duration': {'text': '1 min', 'value': 33}, 'end_location': {'lat': -1.26391, 'lng': 36.802833}, 'html_instructions': 'At Tawis Interior Designs, continue onto <b>Ring Rd Parklands</b><div style="font-size:0.9em">Pass by Kismayu Shop (on the left)</div>', 'polyline': {'points': 'druFacs_Fp@N|B\\j@FTHPHB@'}, 'start_location': {'lat': -1.2625866, 'lng': 36.8032146}, 'travel_mode': 'DRIVING'}, {'distance': {'text': '35 m', 'value': 35}, 'duration': {'text': '1 min', 'value': 20}, 'end_location': {'lat': -1.2636525, 'lng': 36.8028658}, 'html_instructions': 'Make a <b>U-turn</b>', 'maneuver': 'uturn-right', 'polyline': {'points': 'lzuFu`s_FIJI@S'}, 'start_location': {'lat': -1.26391, 'lng': 36.802833}, 'travel_mode': 'DRIVING'}, {'distance': {'text': '63 m', 'value': 63}, 'duration': {'text': '1 min', 'value': 33}, 'end_location': {'lat': -1.2632577, 'lng': 36.8024664}, 'html_instructions': 'Turn <b>left</b> at Taifa Butchery<div style="font-size:0.9em">Pass by the pharmacy (on the right)</div>', 'maneuver': 'turn-left', 'polyline': {'points': 'xxuF`s_FOVCDMHMJ[Z]', 'start_location': {'lat': -1.2636525, 'lng': 36.8028658}, 'travel_mode': 'DRIVING'}], 'traffic_speed_entry': [], 'via_waypoint': []}], 'overview_polyline': {'points': 'tztFwss_Fs@[Oe@|@INzAz@r@b@nAr@vAx@nAt@|@d@fBfApBhARCTB~@PPFh@RzAZhDd@f@RB@IJI@SOVQNk@f@'}, 'summary': 'Ring Rd Parklands', 'warnings': [], 'waypoint_order': []}]
```

We can pick the distance between the two routes by running:

```
Distance = route[0]['legs'][0]['distance']['value']
```

This gives 924, which is in meters.

As indicated earlier, the variation in the values from openrouteservice and Google maps is insignificant.

4.2 Network Plot

Once the network design has run successfully, the summary page appears as shown below:

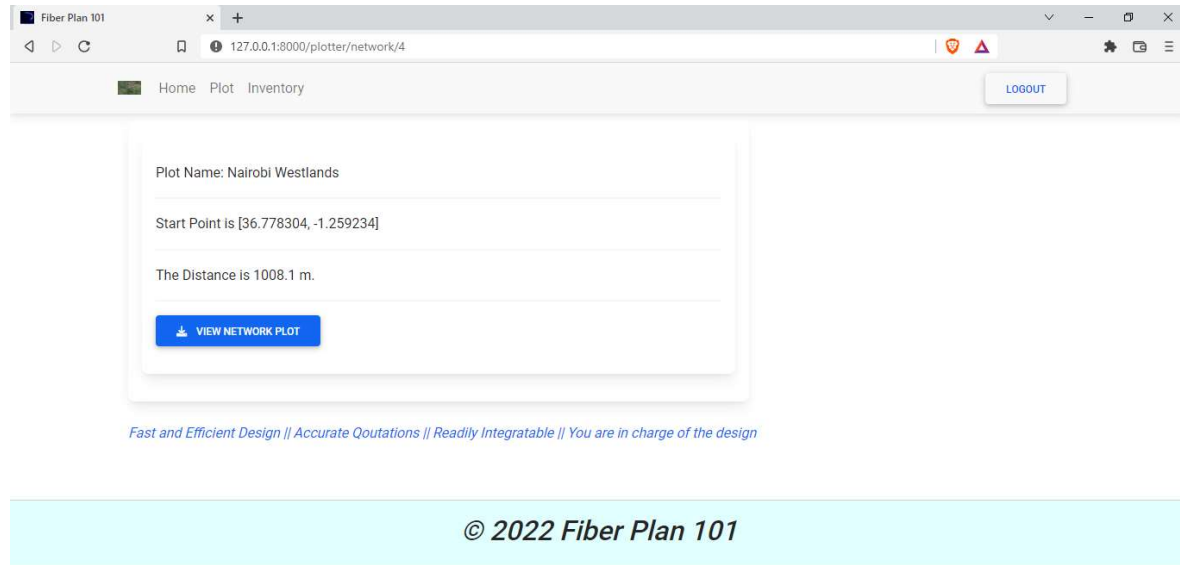


Figure 22 Network Plan Result Page

The network plan is presented as a graph showing the interconnection between the points. The location of each point to be connected is shown using a marker. Clicking a marker shows the name of the customer premise.

The recommended OLT location/ OLT location set by designer and the point from which the redundant network connection starts are indicated by OLT, and RP respectively. The point of integration with existing network is labelled IP.

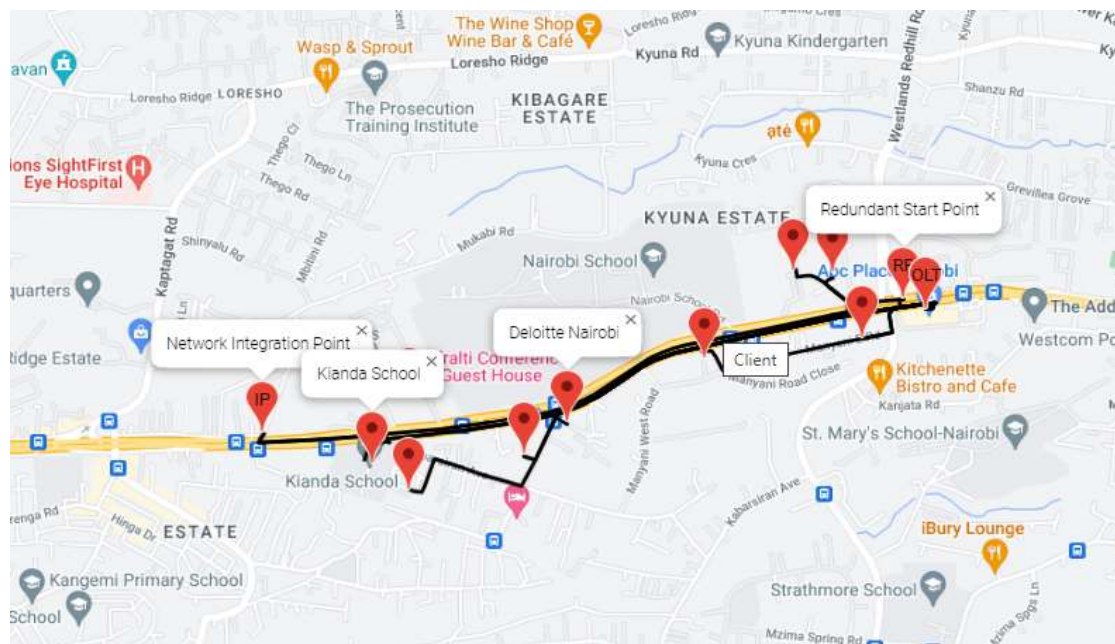


Figure 23 Sample Designed Network

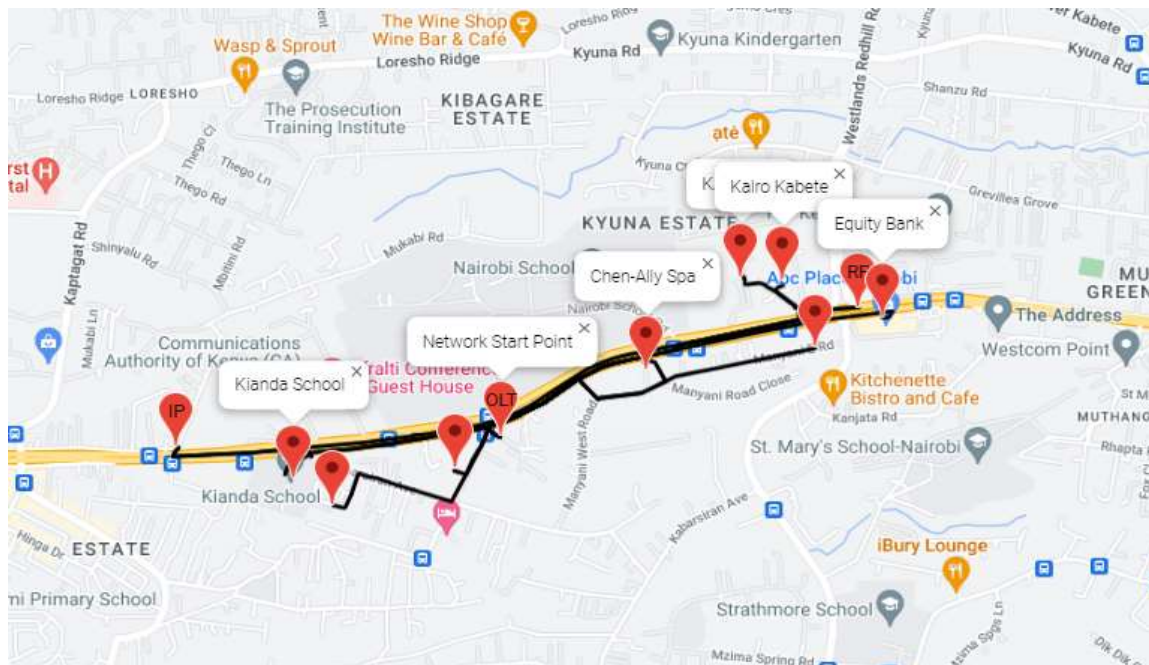


Figure 24 Plot with a Set OLT point

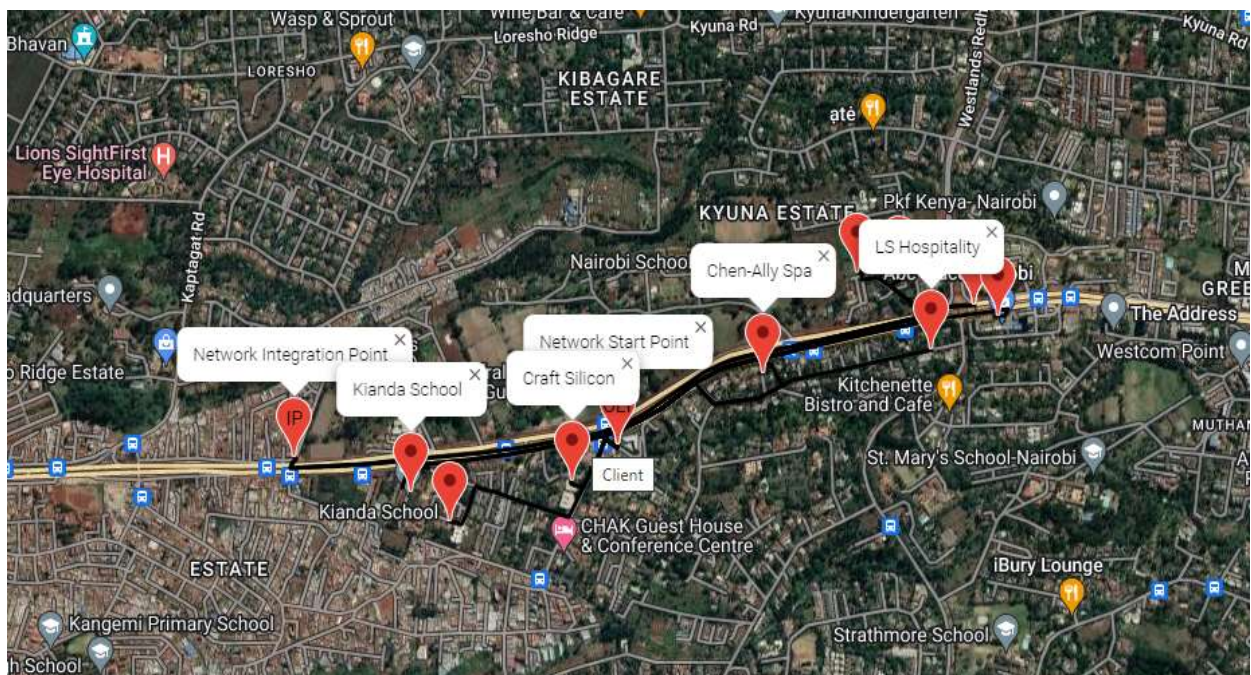


Figure 25 Changing the Plot Layout

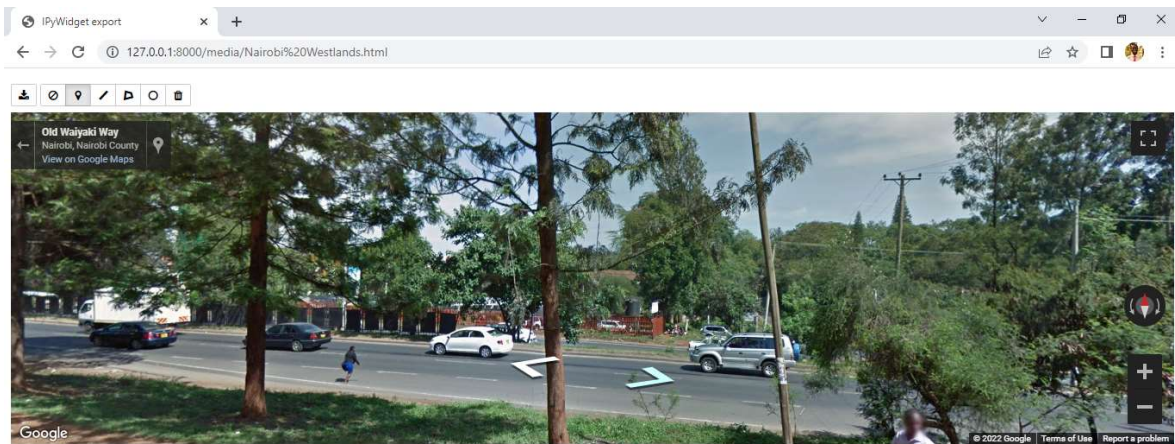


Figure 26 Google Maps Street View

The Google Maps street view feature can be used to zoom in into the network paths and determine their suitability. Street view can be activated by dragging a peg man on the map to any location of interest.

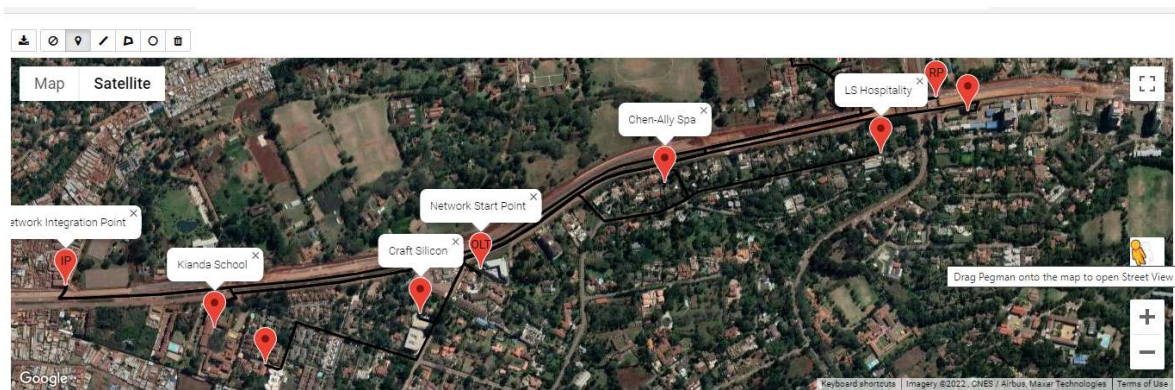


Figure 27 Starting Street View

5/25/22, 3:18 AM Fiber Plan 101

#	Category	SKU	Quantity	Unit Price	Price
1	Poles	Wooden Pole	25	9000.0	225000.0
2	Fibre Cable	ADSS 24 Core Cable	4803.8	130.0	624494.0
3	Man hole	Masonry Manhole	16	10000.0	160000.0
4	Hand Hole	Precast Handhole	4	3000.0	12000.0
5	Support Tangent	Tension Clump, Support Tangent	25	600.0	15000.0
6	ONU	4 Port GPON ONU	10	3500.0	35000.0
7	OLT	PON port EPON OLT		70000.0	70000.0
TOTAL				1141494.0	

VIEW DOCUMENT NETWORK PLOT

Fast and Efficient Design // Accurate Quotations // Readily Integratable // You are in charge

Figure 28 Network Deployment Cost

Above is a sample networking plotting cost result.

A record of the planning cost for a particular deployment is saved in a document and kept in the system database along with the network plot, and can be referred to in the future.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

From the results, the presented design successfully accomplishes the project objectives. An FTTx network-planning platform is developed.

The platform greatly reduces the time and effort spent in designing a Fiber network. Traditional methods involve using multiple tools for the network design process, which is tiresome. On the other hand, manually designing the network is likely to result in a network that is not optimized, for which it is difficult to provide a documentation.

The generated design is also practically implementable. Automating the design process may result in a wonderful blueprint that remains just as such as it is impossible to implement. Here, a remarkable design that represents a realizable network is presented. The resulting network topology is also easy to maintain as it uses well accessible routes.

Finally, the platform provides a reliable network documentation that can always be referred to. This come in a handy during repair and maintenance works on the network.

5.2 Recommendations for Future Work

The presented network-planning platform greatly simplifies the work involved in designing a new FTTx network or expanding an existing one. The platform can however be refined further and as such made more convenient to use by making the following improvements:

1. Automate the process of locating network components such as manholes, hand holes, cabinets and splitters and present on the map alongside the other features of the network. The present design allow these to be added to the plot using the Google Maps draw layer feature.
2. Make it possible to directly pick coordinates from the system and use them for the design.
3. Expand the system to support the design of larger networks.

REFERENCES

1. Vavi Solutions Inc, "FTTx Network Design and Deployment," Available: <https://www.viavisolutions.com/en-us/fttx-network-design-deployment>. [Accessed Feb 2022].
2. C-H. Lee, W. V. Sorin, B. Y. Kim: Fiber to the Home Using a PON Infrastructure (IEEE Journal of Lightwave Technology, Dec 2006)
3. Deeksha Kocher et al., "Simulation of fiber to the home triple play services at 2 Gbit/s using GE-PON architecture for 56 ONUs," *Optik*: Elsevier B.V, pp. 5007-5010, 2013
4. Juan Rendon Schneir and Yupeng Xiong, "Economic implications of a co-investment scheme for FTTH/PON architectures," *Telecommunications Policy*: Elsevier B.V, pp. 849-860. 2013.
5. TechTarget, "Passive Optical Network (PON)," Available: <https://www.techtarget.com/searchnetworking/definition/passive-optical-network-PON>. [Accessed Feb 2022]
6. Baudcom, "What is APON, BPON, EPON, GPON," 22 February 2021. [Online] Available: <https://www.e1-converter.com/TechnicalLectures/What-is-APON-BPON-EPON-GPON.html>. [Accessed Feb 2022].
7. F. Selmanovic and E. Skaljo: GPON in Telecommunication Network (2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops).
8. Yinghui Qiu, "Availability estimation of FTTH architectures based on GPON," in *proc. IEEE 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4.
9. Jani Saheb Shaik, "FTTH deployment options for telecom operators," www.sterlitetechnologies.com.
10. Satyanarayana Katlay and Abhinov Balagoni, "Technological and Cost based Analysis of Future-Proof Fiber Access Passive Networks: GPON and WDM PON," arXiv, pp. 1-4, 2013.
11. Junqiang Hu et al, Triple play services over a converged optical/wireless network, Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference, March 2006, Princeton, USA;
12. Edvin Skaljo et al, „Migration from G(E)PON to NG PON, ICUMT09, St Petersburg, Oct 2009;
13. ITU-T/G.984.4 "A Broadband Optical Access System with Increased Service Capability Using Dynamic Bandwidth Assignment";
14. Sayeed Rashid, article "Beyond bandwidth: QoS and service differentiation in GPON", Dec. 1, 2005
15. Smith, S, Business class services over a GPON network; OFC 2006, Richardson, TX, USA;
16. Edvin Skaljo et al, "GPON in TDM environment", MIPRO 2009, Opatija, Croatia May 2009;
17. Satyanarayana Katlay and Abhinov Balagoni, "Technological and Cost based Analysis of Future-Proof Fiber Access Passive Networks: GPON and WDM PON," arXiv, pp. 1-4, 2013.
18. Mahmoud Al-Quzwini, "Design and implementation of a Fiber to the Home FTTH access network based on GPON," International Journal of Computer Applications · March 2014
19. A. Mitsenkov et al: Topology Design and Capex Estimation for Passive Optical Networks.
20. **Ji Li and Gangxiang Shen: Cost Minimization Planning for Passive Optical Networks**, Optical Society of America, 2008.
21. Digisol, "FTTx Designing Best Practices," Available: www.digisol.com%2Fwp-content%2Fuploads%2F2020%2F03%2FDigisol-FTTx-Designing-Best-Practices-Whitepaper.pdf&clen=908974&chunk=true. [Accessed Feb 2022]
22. <https://giscience.github.io/openrouteservice/documentation/Documentation.html> [Accessed May 2022]

23. <https://github.com/googlemaps/google-maps-services-python/blob/master/googlemaps/directions.py> [Accessed May 2022]

APPENDIX

Appendix A: Project Code

The system was built using 4 Django Projects and spans well over 1500 lines of code.

Below is the manage.py code

```
import os
import sys
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'django_project.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Failed to load project"
        ) from exc
    execute_from_command_line(sys.argv)
if __name__ == '__main__':
    main()
```

1. Django_Project

Manages the application. Has the following files:

a) Asgi.py

```
import os
from django.core.asgi import get_asgi_application
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'django_project.settings')

application = get_asgi_application()
```

b) settings.py file

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
SECRET_KEY = 'django-insecure-g-o2_qe_z(vc5w%01z^b@7yu=j^d3z=u27wu1qd8*(t!=*x6'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['127.0.0.1',]

# Application definition
INSTALLED_APPS = [
    'blog.apps.BlogConfig',
    'users.apps.UsersConfig',
    'crispy_forms',
    'plotter.apps.PlotterConfig',

    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
```

```

'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'django_project.urls'

TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]

WSGI_APPLICATION = 'django_project.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': BASE_DIR / 'db.sqlite3',
}
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
{
'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{

```

```

    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]
# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/
STATIC_URL = 'static/'
CRISPY_TEMPLATE_PACK = 'bootstrap4'
# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
LOGIN_REDIRECT_URL = 'blog-home'
LOGIN_URL = 'login'
MEDIA_ROOT = 'media/'
MEDIA_URL = 'media/'

```

c) Urls.py file

```

"""django_project URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.0/topics/http/urls/
"""
from django.contrib import admin
from django.contrib.auth import views as auth_views
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static
from users import views as user_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('register/', user_views.register, name='register'),
    path('login/', auth_views.LoginView.as_view(template_name='users/login.html'), name='login'),
    path('logout/', auth_views.LogoutView.as_view(template_name='users/logout.html'), name='logout'),
    path('plotter/', include('plotter.urls')),
    path('', include('blog.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

d) Wsgi.py file

```

"""
WSGI config for django_project project.

```

```
It exposes the WSGI callable as a module-level variable named ``application``.  
"""  
import os  
from django.core.wsgi import get_wsgi_application  
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'django_project.settings')  
application = get_wsgi_application()
```

2. Blog Project
Defines the website-landing page.
 - a) Apps.py

```
from django.apps import AppConfig
class BlogConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'blog'
```

- b) models.py

```
from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User

class Post(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    date_posted = models.DateTimeField(default=timezone.now)
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    def __str__(self):
        return self.title
```

- c) url.py file

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='blog-home'),
]
```

- d) Views.py

```
from django.shortcuts import render
from .models import Post
def home(request):
    context = {
        'posts': Post.objects.all()
    }
    return render(request, 'blog/home.html', context)
```

- e) HTML Files
Base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"> {% if title %}
    <title>Fiber Plan 101 - {{ title }}</title>
```



```
{% else %}
<title>Fiber Plan 101</title>
{% endif %}

<link rel="icon"
href="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAQ/2wCEAAKGBw8NDQ8NDw0QDQ4ODQ8OEAOQDRAOE
A4PFRlWfHURFh8YHighGBolGxUXIj0hJSkrLi8uFx8zODMsNyguLysBCgoKdG0OFxAQFS4hFR0tL50rKysrKysrKzcrKys
rLTcyMC0tKystL5suLSsrKzcrL50rKystLTcrL5srL58tN//AABEIALMBGQMBiGACEQEDEQH/xAAcAAEAAQUBAQAAAAAAAA
AAAAAAAAQIEBQYIAwf/xABHEAACAgEBAwGFCACHAwUAAAAAAQIDEQQFEiETfzFBUVWU0wYiMmFxFcNCUoGRodEHYnKckrH
BMzRTk7Kz0nTC8CRDY3OE/8QAGQEBAQEBAQEAAAAAAAAAAAECAwQF/8QALBEBAAIIBAQEYEBQAAAAAAAAAAAECEQMEE
hMUIEYQVGHBSJhgZEjQpKx0f/aAAwDAQACEQMRAD8A+HgA7oAAAAACgCQAABUAAVQABAAFWAGCcFwIJAaaaAAAAACgAGCgA
SUAaaBQAARQMFRJwiI5U4J3ScolSRdr6mUKsqVXvCmuwnlV2GsUtq1VLtX3lapj/wCZZT8o9y/En5U+xGv02fmesaY9j
/gmz1jRH6r/AIF/Utv1kuxfcvyJWumuz7vyNRfT9GJrdeQpX1Zfw1IuK6P1ZfxVoxq21Yuh4+2X51cdr3Lon+Mvz0ldTS+
rFtPUZmrSSf0H/mIu6tmyf0Jf5sjAw2/qF0T/AJ/mXfPqTq49E19z/M711dD0/pwvo689pj8y2KrZL642f59i/oXdex12W
r/9FhrtXpzrodE4fbF/mXdf6SNfH/A18a5/8jff0vKPaHh1Nn279uP5Sz0d1Q+vcvhZZL+pw9mVdd9y+KuMNX+k/WL2qNN
L92xf9xcR/ShY/b0NEvtf9UxxtP0hwjZ/iET1j8W/2Fzbs+rq1T+1pf6kwV2zas/3uL92/pP6pHo/0iUT9vZ1a98XH/iUP
0w0M+nS2V/CT/ozpGppT3mHr0qbRHirPtLwlsJS4px173RRPP8ACy1t2Bjpd7dPFD+XAupbX2dY/qvtnHP4yTPSqymX9l
qEvdv7v8ApcTcU0rek/d76TfzYwZyi6EqH71qZQ190keNmWZL/wBQ392Vdn5G0qFqXBuxe6af+pP+Z5Ssgn68FB9rpw/4o
Cdj05+j1U692n2bLkuuUf8A7KbIfik0eD0M+pxn7ozi393Sb3UozXqzb/Zt5T8J9At0Sn0quxfr1Yf3r8jM/Donwy9NdDe
7Pn06pR9qMo/GLRRg3qzZUf8ADnD312b8ftTw/wACznsSux7sZQ1L6m7yVn3cG/uZ5r7FeqX0Jq1IGb1Po/KLe71NLLi4u
TXxSW8v4ftMXdpJwW845jnG/Fqcc/FH1tS1e80Exh4AagAAIAAooAB5WgAkIAAoAAoAAAAACgACiQAUAAAABQAAHtTqrK/Y
slHHZJoyWn9JNTDg5qxdkl/VYMODrTVvTw2mCJw2aHpfTzjlqMPptJKEPf1NfezJaTVV2f2N8s/VU1Y/4Z419zNHJTPTTb
bx4uvtLrXWm08PpFGotTxuRuS6eTe5YvjCeP513XbRqPmp0T6abYbsvjuyWfuNB003r6sKTV0F9GzLa+Eu1Gy6HbdGpSh
JpS/wdQt+Of1ZdX4M9tNsrfpn7S7ztEWhmrtjvd+asainlV2p31J+7L3ofGMLgxuq0GXm2PIWPhG7fzXLSirelfsZ2Z1My
Wnv1U01Y4L/AA750cJL9Wzpj+91F9KTnmLTrnJcYOMXvLtx7Ni+HH3MmpWlu0YebUmHz3amzVCx12w5Gx53bVDdjJ9k4rh
9seP6rMLqtNKqW7JccJpp5jKL6JRfWn2n0rU6am1fI9Q1UpR+anxcI9jg3xjHP0XwXaug1DUa0dd0tm6r1Zxk+RufQpPjF
5+rLh9+elHztTT3ZcYs14FVkhGTjJOMotx1F8Gmnhp1JyaAAUUAEnKUABQABQAAAAFAAFakAAACgACgAAAAKABJQABcIAA
uB19k7ft0/qSfK09dcvWx70bjodbv1crpp0yuPrT0z9aVfH2opcce9ce1TPm5ebL2lbpLVbVJqSfR1P3M601JjpPZJfR7b
oayrEsyq9pzjh2ad9C1NbXtQXHK7M9GHFYbV7Ns1sbdFfu/LtDBS0964LU6dv2H710p9WxnoZ7V69RjDaem+br1PGqpSyq
LHwd0V1xfRKK6eHQ8NZ3UaeLv02qrxFTqmuHrwUXB78PFepJdqu03Pzd3C1sPm22Y0Sq1DTU7K921NYaug3Bt+9q0fjk
xZtnpdV6qeN17u/JdPrckLafXn0c9uUzH5u1ZzAASGLAAPKoACgAAAAKAAKBIAAAFAAFAAAAAUACSgADUQgCQbioAA1u
oAExi28JZb4JLi2yYGxehuuVcr6prepn7Jzg+uMVma+2G99qj2H0L0foa0d1Gd6dMro1S+142bzo4/sY+yxGkehmyVZ0/
fe7XHT2wvu6Y178d3k12y4vo68H0fVSen0S50Cg7p1Vwg4pS3ZKEeP6zjGS93q9jN0jEZebWnrHjz0ssjKp2ReYz5CqDz
nehCL4r93cf2momy+mE4xjP6YJKMYSxjrjF8mpfBuEn8MGtH0e7vTsAAjagAH1UAAAAFAAFakAAACgACgAAAAKABJQABYQ
JIIJ01YakFSR2rV1GBgu9Fs6694qq1PtaWIr4t8F9psexPRY9E+Fnz+o9WUYVWwVdMYqTnys5YgsrGHvY4YxJvB1mmIy1FL
T5NX0osrMYWE3u7zzhv6q62/css3DYnodJpTvK9LU115SwpS4TYinwrT6OPHi11m8bE2JVCMZU11xsUcTudme15g5pNQxh
Zit3K6eJ70UapRnVKMp728tZ0tzrh9H/wBPW+N0+00U16vHCcvZ0VsQtqbsJWhr0dMKoxhTp60rnXiW/wAMdk9qXFtZ6Zy
aS3VxMT6R7VslyMt1RnN2LT1yksb7i1vya+jCub1KS4Zw1w6bnelqbHpkS7jsi75TmrJStUuHKWdMp9WierBK5XGKRofpr
txW6i2FcspR+TxcX6tWng8KuPvk1vSfv3ep5zMzDzbmbMBtvWk++UoNuuCjVX1Ye5BYT92eL/eZYAHN6AAAAUAA8igAKAAK
BIAAAFAAFAAAAAUACSgAAAAARqESSiEe+ihXK2EbbHVU5YnZGv1XCPao5W98Mo61R6Vwp5Le1ZN2co48ioYW5upqe8325WM
dWTK7EnhSsjs1apV20Zm5Wbtae+3XLHXNR6eGNx4MbVufRpdjX0pveFufj/V1/pdpThCdeKnvUdSjUpOvd3uC+hx3uOVJ8F
0dfspDVektq2fs/VXwgrqYJ3WqVEeUnJcm8xVca4tJ+svabfRjBnnp1p61OajTGiMG3KShGLfQKaj7Sy39CFf7RpC2pbKm
uctRRo4Q3qoyqjG3VycfWcuHGHtcGtxcOHQRpNow08+X5BKMoze1E52aq5uDUZQw1uYbTymuhLefXq1Zt3dp1Yb3yknXy
sktLRJyfl6nCs3Vx1CESpY4etNTfamYfa+05SXqlw0tE8uWqn/fdWt3D+Txk8q07wds21hvDSW69ap25dxhCqPKS1GVBc
VZdHpy+PqRy2nvbuVjhjLZkad3RyldqZ8vrU1ZZv/OR0kfoys3s79jeN2t8c41Po3ThqViHG1ss16V7ajpdDCMYzq1mqpd
XJPdj8m07W67GopfOzgoLLSaTaSjlo+ZMvNqa+equndNtuT4Jycml2ZfS+vPW231lmct3DEIYDBiVAARVAAPKoACgSAAAB
QABQAAAAFAAAkoAAAAACgEAjUIku53VKMOTqe8oJWSskp5sy8uKSUSucY4PPXxLQk6VRdUardsh0cFdGE1J0zc1XNL6L3Wm18
MFDtcuC4JvhGKwvh7zxLvQV3KcbaVKMq5qStXqquccNPefBPoPRW2DrK6ssqrm3p4y30G7dqIxVnQuiKbS4/H7CrZ2neps
k+vJ1UGurNXfvbsWo5UcpPenjCXa1xRbTdcW5T18osbbaTahntb6ZFzj4njqNXOzCk/Vj7M1pRhH4JcPt0u/OMQSy0tr16
WLhok1N8Ja2axbn/4/qfte1x4bvHOP1utj0uuquMoQjF5s3pbzvtvftWN4XDqS61ntZZNkH0206DIBBxtKgDBzkaAFUAA8q
hIAAAFAAFAAAAAUACSgAAAAKAACANQJBBJqJEnq9RPk+S35cmpuxV7z3FNpJyx0Zwks+48STpEonIIBveEkaEmwEAGJkA
AZAAAUeGH1aAAUAAUAAAABQAJKAAAAAoAAIAAoAAoEkE1AAFYAAAZQAAyAAIAAKAAA26j9F23LIRshs6UoTjGcZfKNMsa
ynxn2M90ajb3dsV6EXzDpzYP9y0v/S0f7cS/PmcWzeHKNrRt7u2XidL5g5qNvd2y8TpFm0qwOLYw5U5qNvd2y8TpFMHNRt
7u2XidL5h1WBxbGhKNrRt7u2XidL5g5qNvd2y8TpFm0qwOLYw5U5qNvd2y8TpFMHNRt7u2XidL5h1WC8axhypzU7e7t14n
```

```

S+Y0anb3dsvE6XzDqsxG39py00VGMVmyu7dnv8ArQnGGY4jh7/4JY4jjWM0a+anb3dsvE6XzBzU7e7t14nS+YdB3+k1sIW
QenitRC0IR5aUo2T3LJNx9RNpbkM80me0xuqHpJarVW6KrN++EIzr1UVGMJQqlnM0t6T5RtJcWovsHG5Yc9c103u7ZeJ0v
mE8103u7ZeJ0vmHQdnPjBuQfIVVynXpbsS1MpKNdt1cZb2K/UxGa4v34zhkT9J5yU4xpjVLk5uNkruEZKPCr2P7ddPJ/iO
NYw595qdv2y8TpfMHNTt7u2XidL5h0Bxt5/J6rJQjCydnJzhbdyap9v1rGovdy4YWE020k2nkt6/SayctyG1i3w6dQ0o5
shBQ1828WJzWY9S62Xj2M0euanb3dsvE6XzBzU7e7t14nS+Yd07P1krq6pupwV1TnJ76kq5ppbnU30vj6PVnBrsvS05Kr
E6bm7tTCzk4NqSjqdyEYJSbb5N5aXulwQ49jD4JzU7e7t14nS+Y0anb3dsvEaXzD79otvX2K10ypSnfyc1GG9vQcE3ZB5x
Ldbw8Zxh56GelG2NTvxUp1pT3Z1RnU4Sspm7PnHx4bsYxm+XSw8ZTHHsYc+8103u7ZeI0vmDmp293bLxG18w6B0npF0fye
Mp14no42W6itJr1XGT1JR3niEeTknxfGcV8fXZ21dS9XVp7t3140Uo7m7Ysx10Lay+CW7Fy6N7K6S8xYw555qdv2y8Rpf
MHNTt7u2XiNL5h1UBzFzD1Xmp293bLxG18wc103u7ZeI0vmHVQHMXM0Veanb3dsvEaXzBzU7e7t14jS+YdVAcxcw5V5qdv
d2y8RpfMJ5qdv2y8RpfMQqXmbm65W5qdv2y8RpfMHNVt7u2XiNL5h1SBzNzdcrc1W3u7ZeI0vmDmq293bLxG18w6pA5
m5urDYP9y0v/S0f7cS/AP0oAAAAAAAAAABAAAMACABIAAAAAAIIJQAegAAAAAAAAAAAAAA//Z"

    type="image/x-icon" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" />

    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700;900&display=swap" />
    <link rel="stylesheet" href="{% static 'blog/css/mdb.min.css' %}" />
    <link rel="stylesheet" href="{% static 'blog/css/style.css' %}" />

</head>

<header>
    <nav class="navbar navbar-expand-lg navbar-dark d-none d-lg-block" style="z-index: 2000;">
        <div class="container-fluid">
            <a class="navbar-brand nav-link" target="_blank">
                <strong>Fiber Plan 101</strong>
            </a>
            <button class="navbar-toggler" type="button" data-mdb-toggle="collapse" data-mdb-
target="#navbarExample01" aria-controls="navbarExample01" aria-expanded="false" aria-
label="Toggle navigation">
                <i class="fas fa-bars"></i>
            </button>
            <div class="collapse navbar-collapse" id="navbarExample01">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                </ul>

                <ul class="navbar-nav d-flex flex-row">
                    <li class="nav-item active">
                        <a class="nav-link" aria-current="page" href="#intro">Home</a>
                    </li>
                    {% if user.is_authenticated %}
                    <li class="nav-item">
                        <a class="nav-link" href="{% url 'plotter-index' %}">Plot</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{% url 'plotter-item_list' %}">Inventory</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{% url 'logout' %}">Logout</a>
                    </li>
                    {% else %}

```

```

        <li class="nav-item">
            <a class="nav-link" href="{% url 'login' %}">Login</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{% url 'register' %}">Register</a>
        </li>
        {% endif %}
    </ul>
</div>
</div>
</nav>
</header>

<body>

    {% block content %}{% endblock %}

    <footer class="bg-light text-lg-start">

        <hr class="m-0" />

        <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
            <h2 class="text-dark" href="#"> © 2022 Fiber Plan 101</h2>
            <a class="text-black" > This work was submitted in partial fulfillment of the
requirements for the award
            of BSc in Electrical and Electronics Engineering.</a>
            <p class="text-black" > Allan Ooko, Department of Electrical and Information
Engineering, University of Nairobi.</p>
        </div>
    </footer>

    <script type="text/javascript" src="{% static 'blog/js/mdb.min.js' %}"></script>
    <script type="text/javascript"></script>

</body>
</html>

```

Home.html

```

{% extends "blog/base.html" %} {% block content %}

<!-- Carousel wrapper -->
<div id="introCarousel" class="carousel slide carousel-fade shadow-2-strong" data-mdb-ride="carousel">
    <!-- Indicators -->
    <ol class="carousel-indicators">
        <li data-mdb-target="#introCarousel" data-mdb-slide-to="0" class="active"></li>
        <li data-mdb-target="#introCarousel" data-mdb-slide-to="1"></li>
        <li data-mdb-target="#introCarousel" data-mdb-slide-to="2"></li>
    </ol>

    <!-- Inner -->

```

```

<div class="carousel-inner">
  <!-- Single item -->
  <div class="carousel-item active">
    <div class="mask" style="background-color: rgba(0, 0, 0, 0.6);">
      <div class="d-flex justify-content-center align-items-center h-100">
        <div class="text-white text-center">
          <h1 class="mb-3">Faiba Plan 101</h1>
          <h5 class="mb-4">Plan your fiber network efficiently today!</h5>
          <a class="btn btn-outline-light btn-lg m-2" href="{% url 'plotter-index' %}" role="button"
rel="nofollow" target="_blank">Plot</a>
        </div>
      </div>
    </div>
  </div>

  <!-- Single item -->
  <div class="carousel-item">
    <div class="mask" style="background-color: rgba(0, 0, 0, 0.3);">
      <div class="d-flex justify-content-center align-items-center h-100">
        <div class="text-white text-center">
          <h2>Fast and Efficient Design || Accurate Qoutations</h2>
        </div>
      </div>
    </div>
  </div>

  <!-- Single item -->
  <div class="carousel-item">
    <div class="mask" style="
      background: linear-gradient(
        45deg,
        rgba(29, 236, 197, 0.7),
        rgba(91, 14, 214, 0.7) 100%
      );
    ">
    <div class="d-flex justify-content-center align-items-center h-100">
      <div class="text-white text-center">
        <h2>You call the shots</h2>
        <h5>You define how the design takes place</h5>
      </div>
    </div>
  </div>
</div>

<!-- Inner -->

<!-- Controls -->
<a class="carousel-control-prev" href="#introCarousel" role="button" data-mdb-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#introCarousel" role="button" data-mdb-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>

```

```

        <span class="sr-only">Next</span>
    </a>
</div>

<main class="mt-5">
    <div class="container">
        <!--Section: Content-->
        <section>
            <div class="row">
                <div class="col-md-6 gx-5 mb-4">
                    <div class="bg-image hover-overlay ripple shadow-2-strong rounded-5" data-mdb-ripple-color="light">
                        
                        <a href="#!">
                            <div class="mask" style="background-color: rgba(251, 251, 251, 0.15);"></div>
                        </a>
                    </div>
                </div>
                <div class="col-md-6 gx-5 mb-4">
                    <h4><strong>Welcome to Faiba Plan 101</strong></h4>
                    <p class="text-muted">
FTTX (Fiber to the X) represents the technology aspirations and ambitions of today's businesses that seek reliable technology investment guidance and reasonably assured Return on Investment (ROI).

                    </p>
                    <p class="text-muted">
                        It is a means of providing Next Generation Access (NGA) to corporates that rely on a robust IT communications mechanism to get more done, faster.
                    </p>
                    <p class="text-muted">
                        Fiber Plan 101 allows you to plan your FTTX network efficiently. <br> Plan your Fiber network efficiently today!
                    </p>
                </div>
            </div>
        </section>
        <!--Section: Content-->

        <hr class="my-5" />

        <!--Section: Content-->
        <section class="text-center">
            <h1 class="mb-5"><strong>Why Fiber Plan 101?</strong></h1>

            <div class="row">
                <div class="col-lg-3 col-md-12 mb-4">
                    <div class="card">
                        <div class="bg-image hover-overlay ripple" data-mdb-ripple-color="light">
                            
                            <a href="#!">

```

```

        <div class="mask" style="background-color: rgba(251, 251, 251, 0.15);"></div>
    </a>
</div>
<div class="card-body">
    <h5 class="card-title">Accuracy</h5>
    <p class="card-text">
        Fiber Plan 101 lets you plan your fiber network accurately .
    </p>
</div>
</div>

<div class="col-lg-3 col-md-6 mb-4">
    <div class="card">
        <div class="bg-image hover-overlay ripple" data-mdb-ripple-color="light">
            
            <a href="#">
                <div class="mask" style="background-color: rgba(251, 251, 251, 0.15);"></div>
            </a>
        </div>
        <div class="card-body">
            <h5 class="card-title">Input</h5>
            <p class="card-text">
                All you need to do is input your GPS coordinates and we'll handle the rest. </p>
            </div>
        </div>
    </div>

    <div class="col-lg-3 col-md-6 mb-4">
        <div class="card">
            <div class="bg-image hover-overlay ripple" data-mdb-ripple-color="light">
                
                <a href="#">
                    <div class="mask" style="background-color: rgba(251, 251, 251, 0.15);"></div>
                </a>
            </div>
            <div class="card-body">
                <h5 class="card-title">Output</h5>
                <p class="card-text">
                    The planned network is shown on a map, with the network paths <br> clearly drawn. </p>
                </div>
            </div>
        </div>

        <div class="col-lg-3 col-md-6 mb-4">
            <div class="card">
                <div class="bg-image hover-overlay ripple" data-mdb-ripple-color="light">
                    
                    <a href="#">
                        <div class="mask" style="background-color: rgba(251, 251, 251, 0.15);"></div>
                    </a>
                </div>
            </div>
        </div>
    </div>

```

```
</div>
<div class="card-body">
  <h5 class="card-title">Cost Calculation</h5>
  <p class="card-text">
    You define the parameters used in the cost calculation. </p>
  </div>
</div>
</div>

</div>
</section>
<hr class="my-5" />
</div>
</main>
{% endblock content %}
```

3. Users project
Provides user authentication.
 - a) Apps.py file

```
from django.apps import AppConfig
class UsersConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'users'
```

- b) forms.py file

```
import email
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm

class UserRegisterForm(UserCreationForm):
    email = forms.EmailField()

    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2']
```

- c) views.py file

```
from django.shortcuts import render, redirect
from django.contrib import messages
from .forms import UserRegisterForm

def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            messages.success(request, f' {username}, your account has been created, you are now able to login!')
            return redirect('login')
        else:
            form = UserRegisterForm
    return render(request, 'users/register.html', {'form':form})
```

- d) HTML files
Login.html

```
{% extends "blog/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
    <form class="form-group" method="POST">
        {% csrf_token %}
        <legend class="border-bottom mb-4">Login</legend>
        <div class="form-group">
            {{ form|crispy }}
        </div>
```



```

        <!-- 2 column grid layout for inline styling -->
        <div class="row mb-4">
            <div class="col d-flex justify-content-center">
                <!-- Checkbox -->
                <div class="form-check">
                    <input class="form-check-input" type="checkbox" value="" id="form1Example3"
checked />
                    <label class="form-check-label" for="form1Example3"> Remember me </label>
                </div>
            </div>

            <div class="col">
                <!-- Simple link -->
                <a href="{% url 'register' %}">Don't have an account? Register.</a>
            </div>
        </div>

        <!-- Submit button -->
        <button type="submit" class="btn btn-primary btn-block">Login</button>
    </form>

</div>
{% endblock content %}

```

Logout.html

```

{% extends "blog/base.html" %} {% load crispy_forms_tags %} {% block content %}

<div class="col p-4">
    <h2>You have been logged out!</h2>
    <p>Login again <a href="{% url 'login' %}">here</a></p>
</div>

{% endblock content %}

```

Logout.html

```

{% extends "blog/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
    <form class="form-group" method="POST">
        {% csrf_token %}
        <legend class="border-bottom mb-4">Sign Up</legend>
        <div class="form-group">
            {{ form|crispy }}
        </div>
        <!-- 2 column grid layout for inline styling -->
        <div class="row mb-4">
            <div class="col d-flex justify-content-center">
                <!-- Checkbox -->
                <div class="form-check">
                    <input class="form-check-input" type="checkbox" value="" id="form1Example3"
checked />
                    <label class="form-check-label" for="form1Example3"> Remember me </label>
                </div>
            </div>
        </div>
    </form>
</div>

```

```
    </div>

    <div class="col">
        <!-- Simple link -->
        <a href="{% url 'login' %}">Already have an account? Login.</a>
    </div>
</div>

<!-- Submit button -->
<button type="submit" class="btn btn-primary btn-block">Register</button>
</form>

</div>
{% endblock content %}
```

4. Plotter Project

Is the engine of the project design.

a) Apps.py

```
from django.apps import AppConfig
class PlotterConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'plotter'
```

b) forms.py

```
from django import forms
from django.forms import ModelForm
from django.db.models import Q

from plotter.models import Coordinate, Plot, Item

class PointsForm(forms.Form):
    your_name = forms.CharField(label='Your name', max_length=100)

class ItemForm(ModelForm):
    class Meta:
        model = Item
        fields = ['sku', 'unit_price', 'category']

class PlotForm(ModelForm):
    class Meta:
        model = Plot
        fields = ['pole_separation', 'Extra_Fiber_Length_After',
'Extra_Fiber_Length', 'man_hole_separation', 'hand_hole_separation']

class PlotConnectForm(ModelForm):
    class Meta:
        model = Plot
        fields = ['set_start_point', 'connect_coordinate']

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['set_start_point'].queryset =
Coordinate.objects.filter(plot=self.instance)

class QuoteItemsFrom(ModelForm):
    class Meta:
        model = Plot
        fields = ['pole', 'fibre_optic', 'duct', 'man_hole', 'hand_hole',
'Support_Tangent', 'onu', 'olt']

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['pole'].queryset = Item.objects.filter(category='PL')
        self.fields['fibre_optic'].queryset = Item.objects.filter(category='FO')
        self.fields['duct'].queryset = Item.objects.filter(category='DT')
```

```

        self.fields['hand_hole'].queryset = Item.objects.filter(category='HH')
        self.fields['Support_Tangent'].queryset =
Item.objects.filter(category='ST')
        self.fields['onu'].queryset = Item.objects.filter(category='ON')
        self.fields['olt'].queryset = Item.objects.filter(category='OL')

```

c) models.py

```

from django.db import models

# Create your models here.

class Item(models.Model):
    CATEGORY_CHOICES = [
        ('PL', 'Pole'),
        ('FO', 'Fibre optic'),
        ('DT', 'Duct'),
        ('MH', 'Man hole'),
        ('OL', 'OLT'),
        ('ON', 'ONU'),
        ('HH', 'Hand Hole'),
        ('ST', 'Support Tangent'),
        ('OR', 'Others'),
    ]

    sku = models.CharField(max_length=100)
    unit_price = models.FloatField()
    category = models.CharField(max_length=2, choices=CATEGORY_CHOICES)

    def __str__(self):
        return self.sku

class Plot(models.Model):
    name = models.CharField(max_length=100)
    start = models.CharField(max_length=100, null=True, blank=True)
    distance = models.FloatField(null=True, blank=True)
    # items
    pole = models.ForeignKey(Item, related_name='plot_pole',
on_delete=models.PROTECT, null=True, blank=True)
    pole_separation = models.FloatField(null=True, blank=True)
    fibre_optic = models.ForeignKey(Item, related_name='plot_fibre',
on_delete=models.PROTECT, null=True, blank=True)
    Extra_Fiber_Length_After = models.FloatField(null=True, blank=True)
    Extra_Fiber_Length = models.FloatField(null=True, blank=True)
    duct = models.ForeignKey(Item, related_name='plot_duct',
on_delete=models.PROTECT, null=True, blank=True)
    onu = models.ForeignKey(Item, related_name='plot_onu',
on_delete=models.PROTECT, null=True, blank=True)
    olt = models.ForeignKey(Item, related_name='plot_olt',
on_delete=models.PROTECT, null=True, blank=True)
    man_hole = models.ForeignKey(Item, related_name='plot_man_hole',
on_delete=models.PROTECT, null=True, blank=True)

```

```

    man_hole_separation = models.FloatField(null=True, blank=True)
    hand_hole = models.ForeignKey(Item, related_name='plot_hand_hole',
on_delete=models.PROTECT, null=True, blank=True)
    hand_hole_separation = models.FloatField(null=True, blank=True)
    Support_Tangent = models.ForeignKey(Item, related_name='plot_support_tangent',
on_delete=models.PROTECT, null=True, blank=True)
    set_start_point=models.ForeignKey('Coordinate', on_delete=models.SET_NULL,
null=True, blank=True, related_name='plot_set_start_point')
    connect_coordinate = models.TextField(null=True, blank=True)

    def __str__(self):
        return self.name

class Coordinate(models.Model):
    plot = models.ForeignKey(Plot, on_delete=models.CASCADE)
    name = models.CharField(max_length=100)
    coordinates = models.TextField()

    def __str__(self):
        return f'{self.name} {self.coordinates}'

```

d) urls.py file

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='plotter-index'),
    path('plot/create', views.create, name='plotter-create'),
    path('plot/<int:id>/populate', views.populate, name='plotter-populate'),
    path('plot/<int:id>/more_settings', views.more_settings, name='plotter-
more_settings'),
    path('plot/<int:id>properties', views.properties, name='plotter-properties'),
    path('plot/coordinate/<int:id>/delete', views.coordinate_delete,
name='plotter-coordinate-delete'),
    path('plot/<int:id>', views.plot, name='plotter-plot'),
    path('plot/<int:id>/item', views.quote_item, name='plotter-quote_item'),
    path('plot/<int:id>/price', views.quote_price, name='plotter-quote_price'),
    path('network/<int:id>', views.network, name='plotter-network'),
    path('plot/<int:id>/delete', views.plot_delete, name='plotter-plot-delete'),
    # items
    path('items', views.item_list, name='plotter-item_list'),
    path('items/create', views.item_create, name='plotter-item_create'),
    path('items/<int:id>/update', views.item_update, name='plotter-item_update'),
    path('items/<int:id>/delete', views.item_delete, name='plotter-item_delete'),
]

```

e) Utility.py

```

from calendar import c
from ssl import AlertDescription
from turtle import color
import openrouteservice as ors
import folium
import math
import gmplot
import json
import googlemaps
import os
import gmaps.geojson_geometries
from ipywidgets.embed import embed_minimal_html

from plotter.models import Coordinate

Client1 =
ors.Client(key='5b3ce3597851110001cf624820c44dd22f3c4a128773998371409a6f') #ors
api key 1
Client2 =
ors.Client(key='5b3ce3597851110001cf6248e6b29a12715344b7a18a42c037cceb60') #ors
api key 2
apikey =
'AIzaSyA_xtmQprzuxWdhVy0JxjDPMA0loHExtx8' #Goog
leMaps api key

#invert coordinates
def Gmaps_distance(coordinates):
    gmaps=googlemaps.Client(key=apikey)
    MinTot=999999999
    MaxTot=0

    Start = list()
    Altstart=list()

    for i in coordinates:
        Tdistance=0
        Gstart=i
        for y in coordinates:
            end = y
            route=gmaps.directions((Gstart[1], Gstart[0]),(end[1], end[0]),
mode="driving", units='metric')
            distance=route[0]['legs'][0]['distance']['value']
            Tdistance+=distance

            if MinTot>Tdistance:
                MinTot=Tdistance
                Start=i

        #Altstart to be used to create redundancy for network security
        if Tdistance>MaxTot:
            MaxTot=Tdistance
            Altstart=i

```

```

        return Start, Altstart

def ors_route(coordinates):
    route=Client1.directions(coordinates=coordinates,
                              profile='foot-walking',
                              optimized= False,
                              preference='shortest',
                              continue_straight=True,
                              format='geojson')

    return route

def ors_distance(coordinates):
    route=Client2.directions(coordinates=coordinates,
                              profile='foot-walking',
                              optimized= False,
                              preference='shortest',
                              continue_straight=True,
                              format='geojson')

    return route['features'][0]['properties']['segments'][0]['distance']

def Start_Distance(points, connect_point=None):
    Start, Altstart= Gmaps_distance(points)
    for i in points:
        Tdistance=0
        now=[Start,i]
        distance=ors_distance(now)
        Tdistance+=distance

    if connect_point:
        now=[Start, connect_point]
        distance=ors_distance(now)
        Tdistance += distance

        now = [Altstart,connect_point]
        distance=ors_distance(now)
        Tdistance += distance
    print('Not Set Start Distance ran')
    return Tdistance, Start, Altstart

def Set_Start_Distance(points, set_start, connect_point=None):
    Start,Altstart= Gmaps_distance(points)
    for i in points:
        Tdistance=0
        now=[set_start,i]
        distance=ors_distance(now)
        Tdistance+=distance

    if connect_point:
        now=[set_start, connect_point]
        distance=ors_distance(now)
        Tdistance += distance

```

```

        now = [Altstart,connect_point]
        distance=ors_distance(now)
        Tdistance += distance
    Start=set_start

    return Tdistance, Start, Altstart

def plot(points, names, map_name, set_start=None, connect_point=None):
    if set_start:
        Tdistance, Start, Altstart= Set_Start_Distance(points, set_start,
connect_point)
    else:
        Tdistance, Start, Altstart = Start_Distance(points, connect_point)

    gmaps.configure(api_key=apikey)

    fig=gmaps.figure()
    drawing = gmaps.drawing_layer()
    fig.add_layer(drawing)

    for i in range(len(points)):
        if Start == points[i]:
            continue

        now=[Start,points[i]]
        route=ors_route(now)
        geojson_layer = gmaps.geojson_layer(route)
        fig.add_layer(geojson_layer)

    location2=list()
    for i in points:
        start=i
        location2.append((start[1],start[0]))
    marker_layer = gmaps.marker_layer((location2),
                                     hover_text='Client',

                                     info_box_content=names)

    fig.add_layer(marker_layer)

    Start1=list()
    Start1.append((Start[1], Start[0]))
    marker_layer = gmaps.marker_layer((Start1),
                                     hover_text='Build Cabinet',
                                     label='OLT',
                                     info_box_content='Network Start Point')
    fig.add_layer(marker_layer)

    if connect_point:
        now=[Start, connect_point]
        route = ors_route(now)

```



```

    geojson_layer = gmaps.geojson_layer(route)
    fig.add_layer(geojson_layer)

    location3=list()
    location3.append((connect_point[1], connect_point[0]))
    marker_layer = gmaps.marker_layer((location3),
                                       hover_text='Integartion Point',
                                       label='IP',
                                       info_box_content='Network Integration
Point')
    fig.add_layer(marker_layer)

    now=list()
    now.append((connect_point[0],connect_point[1]))
    now.append((Altstart[0],Altstart[1]))
    route = ors_route(now)
    geojson_layer = gmaps.geojson_layer(route)
    fig.add_layer(geojson_layer)

    Altstart1=list()
    Altstart1.append((Altstart[1], Altstart[0]))
    marker_layer = gmaps.marker_layer((Altstart1),
                                       hover_text='Redundant Link Point',
                                       label='RP',
                                       info_box_content='Redundant Start Point')
    fig.add_layer(marker_layer)

    embed_minimal_html(map_name, views=[fig])

    return Tdistance, Start

def item_count(plot):

    total_fibre_excess = math.ceil(plot.distance / plot.Extra_Fiber_Length_After)
* plot.Extra_Fiber_Length
    poles = math.ceil(plot.distance / plot.pole_separation) + 1
    fibre_optic = plot.distance + total_fibre_excess
    man_holes = math.floor(plot.distance / plot.man_hole_separation) + 1
    hand_holes = math.floor(plot.distance / plot.hand_hole_separation)

    return poles, fibre_optic, man_holes, hand_holes

from docx import Document
from docx.shared import Inches

# https://python-docx.readthedocs.io/en/latest/#what-it-can-do
def save_quote_to_file(plot, poles, poles_price,
fibre_optic, fibre_optic_price,

```

```

man_holes, man_hole_price,
hand_holes, hand_hole_price,
Support_Tangent, support_tangent_price,
onu, onu_price,
olt, olt_price,
total
):
    document = Document()
    records = (
        (1, "Poles", plot.pole.sku, poles, plot.pole.unit_price, poles_price),
        (2, "Fibre Cable", plot.fibre_optic.sku, fibre_optic,
plot.fibre_optic.unit_price, fibre_optic_price),
        (3, "Man hole", plot.man_hole.sku, man_holes, plot.man_hole.unit_price,
man_hole_price),
        (4, "Hand Hole", plot.hand_hole.sku, hand_holes, plot.hand_hole.unit_price,
hand_hole_price),
        (5, "Support Tangent", plot.Support_Tangent.sku, Support_Tangent,
plot.Support_Tangent.unit_price, support_tangent_price),
        (6, "ONU", plot.onu.sku, onu, plot.onu.unit_price, onu_price),
        (7, "OLT", plot.olt.sku, olt, plot.olt.unit_price, olt_price),
        ("", "", "TOTAL", "", "", total),
    )

    table = document.add_table(rows=1, cols=6)
    hdr_cells = table.rows[0].cells
    hdr_cells[0].text = '#'
    hdr_cells[1].text = 'Category'
    hdr_cells[2].text = 'SKU'
    hdr_cells[3].text = 'Quantity'
    hdr_cells[4].text = 'Unit Price'
    hdr_cells[5].text = 'Price'

    for index, category, sku, quantity, unit_price, price in records:
        row_cells = table.add_row().cells
        row_cells[0].text = str(index)
        row_cells[1].text = category
        row_cells[2].text = sku
        row_cells[3].text = str(quantity)
        row_cells[4].text = str(unit_price)
        row_cells[5].text = str(price)

    document.add_page_break()
    filename = f'media/{plot.name}.docx'
    # Tthttps://www.w3schools.com/python/python\_file\_remove.asp
    import os
    if os.path.exists(filename):
        os.remove(filename)

    else:
        document.save(filename)

```

f) views.py

```

from django.shortcuts import render, redirect, get_object_or_404
from django.urls import reverse
from django.contrib.auth.decorators import login_required
from django.db.models import Q
from django.contrib import messages

from .models import Item, Plot, Coordinate
from .forms import PointsForm, ItemForm, PlotForm, PlotConnectForm, QuoteItemsFrom
from .utility import plot as plot_network, item_count, save_quote_to_file

# Create your views here.
@login_required
def index(request):
    plots = Plot.objects.filter(~Q(name='settings'))

    context = {'plots': plots}
    return render(request, 'plotter/index.html', context)

@login_required
def create(request):
    if request.method == 'POST':
        name = request.POST['name']
        plot = Plot(name=name)
        plot.save()
        return redirect(reverse('plotter-populate', args=[plot.id]))

@login_required
def populate(request, id):
    plot = get_object_or_404(Plot, pk=id)
    if plot.pole_separation:
        form = PlotForm(instance=plot)
    else:
        settings_plot = Plot.objects.filter(name='settings').first()
        if settings_plot:
            form = PlotForm(instance=settings_plot)
        else:
            form = PlotForm(instance=plot)

    if request.method == 'POST':
        name = request.POST['name']
        coordinates = request.POST['coordinates']
        coordinates = Coordinate(plot=plot, name=name, coordinates=coordinates)
        coordinates.save()

    context = {
        'plot': plot,
        'coordinates': plot.coordinate_set.all(),
        'form': form,
        'title': f'Plot N0# {plot.id}'
    }
    return render(request, 'plotter/populate.html', context)

```

```

@login_required
def more_settings(request, id):
    plot = get_object_or_404(Plot, pk=id)
    form = PlotConnectForm(instance=plot)

    if request.method == 'POST':
        form = PlotConnectForm(request.POST, instance=plot)
        if form.is_valid():
            form.save()
            messages.success(request, 'Settings saved successfully')
            return redirect(reverse('plotter-populate', args=[plot.id]))

    context = {
        'plot': plot,
        'form': form,
        'title': f'Plot N0# {plot.id}'
    }
    return render(request, 'plotter/connect.html', context)

@login_required
def properties(request, id):
    plot = get_object_or_404(Plot, pk=id)
    if request.method == 'POST':
        form = PlotForm(request.POST, instance=plot)
        if form.is_valid():
            form.save()
            messages.success(request, 'Properties saved successfully')

    plot_settings = Plot.objects.filter(name='settings').first()
    form_settings = PlotForm(request.POST, instance=plot_settings)
    if form_settings.is_valid():
        form_settings.save()
    return redirect(reverse('plotter-populate', args=[plot.id]))

@login_required
def coordinate_delete(request, id):
    coordinate = get_object_or_404(Coordinate, pk=id)
    plot = coordinate.plot
    coordinate.delete()

    return redirect(reverse('plotter-populate', args=[plot.id]))

@login_required
def plot(request, id):
    plot = get_object_or_404(Plot, pk=id)
    coordinates = plot.coordinate_set.all()

    array_of_names = []
    array_of_array = []
    for coordinate in coordinates:
        x_y = coordinate.coordinates
        x, y = x_y.split(',')

```

```

        x_y_array = [float(x), float(y)]
        array_of_array.append(x_y_array)
        array_of_names.append(coordinate.name)

    # points=[[36.8159, -1.2795], [36.8219,-1.2921],[36.8259, -1.2850], [36.8145,
-1.2870], [36.8222, -1.2935]]
    points = array_of_array
    names = array_of_names
    map_name = 'media/'+plot.name+'.html'

    set_start = None
    if plot.set_start_point:
        x_y = plot.set_start_point.coordinates # "36.8159, -1.2795"
        x, y = x_y.split(',') # "36.8159" "-1.2795"
        x_y_array = [float(x), float(y)] # [36.8159 -1.2795]
        set_start = x_y_array

    connect_point = None
    if plot.connect_coordinate:
        x_y = plot.connect_coordinate # "36.8159, -1.2795"
        x, y = x_y.split(',') # "36.8159" "-1.2795"
        x_y_array = [float(x), float(y)] # [36.8159 -1.2795]
        connect_point = x_y_array

    MinTot, Start = plot_network(points, names, map_name, set_start, connect_point)

    plot.distance = round(MinTot, 2)
    plot.start = Start
    plot.save()

    return redirect(reverse('plotter-network', args=[plot.id]))

@login_required
def quote_item(request, id):
    plot = get_object_or_404(Plot, pk=id)
    form = QuoteItemsFrom(instance=plot)
    if request.method == 'POST':
        form = QuoteItemsFrom(request.POST, instance=plot)
        if form.is_valid():
            form.save()
            return redirect(reverse('plotter-quote_price', args=[plot.id]))

    context = {'form':form}
    return render(request, 'plotter/quote_items.html', context)

@login_required
def quote_price(request, id):
    plot = get_object_or_404(Plot, pk=id)
    poles, fibre_optic, man_holes, hand_holes = item_count(plot)

    fibre_optic_price = fibre_optic * plot.fibre_optic.unit_price

    poles_price = 0

```

```

man_hole_price = 0
hand_hole_price = 0
support_tangent_price = 0
olt_price = 0
onu_price = 0

onus = plot.coordinate_set.count()

if plot.pole:
    poles_price = poles * plot.pole.unit_price
if plot.man_hole:
    man_hole_price = man_holes * plot.man_hole.unit_price
if plot.hand_hole:
    hand_hole_price = hand_holes * plot.hand_hole.unit_price
if plot.Support_Tangent:
    support_tangent_price = poles * plot.Support_Tangent.unit_price
if plot.olt:
    olt_price = plot.olt.unit_price
if plot.onu:
    onu_price = plot.onu.unit_price * onus
total = poles_price + fibre_optic_price + man_hole_price + hand_hole_price +
support_tangent_price + olt_price + onu_price

Support_Tangent = poles
onu = onus
olt = "1"
save_quote_to_file(plot, poles, poles_price,
    fibre_optic, fibre_optic_price,
    man_holes, man_hole_price,
    hand_holes, hand_hole_price,
    Support_Tangent, support_tangent_price,
    onu, onu_price,
    olt, olt_price,
    total
)

context = {
    'plot':plot,
    'fibre_optic_price': fibre_optic_price,

    # quantity
    'poles': poles,
    'fibre_optic':fibre_optic,
    'man_holes':man_holes,
    'hand_holes':hand_holes,
    'onu': onus,
    'Support_Tangent':poles,

    # price
    'poles_price': poles_price,
    'man_hole_price': man_hole_price,
    'hand_hole_price': hand_hole_price,
    'support_tangent_price': support_tangent_price,

```

```

        'olt_price': olt_price,
        'onu_price': onu_price,
        'total': total,
    }

    return render(request, 'plotter/quote_price.html', context)

@login_required
def network(request, id):
    plot = get_object_or_404(Plot, pk=id)
    context = {
        'plot': plot,
    }
    return render(request, 'plotter/network.html', context)

@login_required
def plot_delete(request, id):
    plot = get_object_or_404(Plot, pk=id)
    plot.delete()

    return redirect(reverse('plotter-index'))

# items
@login_required
def item_list(request):
    items = Item.objects.all()
    context = {'items': items}
    return render(request, 'plotter/item/index.html', context)

@login_required
def item_create(request):
    form = ItemForm()
    if request.method == 'POST':
        form = ItemForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect(reverse('plotter-item_list'))

    context = {'form': form}
    return render(request, 'plotter/item/create_and_update.html', context)

@login_required
def item_update(request, id):
    item = get_object_or_404(Item, pk=id)
    form = ItemForm(instance=item)
    if request.method == 'POST':
        form = ItemForm(request.POST, instance=item)
        if form.is_valid():
            form.save()
            return redirect(reverse('plotter-item_list'))

    context = {'form': form}
    return render(request, 'plotter/item/create_and_update.html', context)

```

```
@login_required
def item_delete(request, id):
    item = get_object_or_404(Item, pk=id)
    try:
        item.delete()
    except:
        messages.warning(request, 'This action is not allowed')

    return redirect(reverse('plotter-item_list'))
```


g) Plotter project HTML files

i. Base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  {% if title %}
  <title>Fiber Plan 101 - {{ title }}</title>
  {% else %}
  <title>Fiber Plan 101</title>
  {% endif %}

  <link href="#" rel="icon"
href="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAAkGBw8NDQ8NDw0QD
Q4ODQ8OEAAQDRARAA4PFRIWFhURFh8YHIGBoGxUXIj0hJSkrLi8uFx8zODMsNyguLysBCgoKDg0
OFxAXFS4hFR0tLS0rKysrKysrKzcrKysrLTcyMC0tKystLSsuLSsrKzcrLS0rKystLTcrLSsrLS8tN
//AABEIALMBGQMBIGACEQEDEQH/xAAcAAEAAQUBAQAAAAAAAAAAAAAAAAQIEBQYIAAw/xABHEAACgE
BAWgFCACHAwUAAAAAAAAQIDEQQFEiETFzFBUVWU0wYiMmFxFNCUoGRodEHYnKCKrHBMzRTk7Kz0nTC8
CRDY3OE/8QAGQEBAQEBAQEAAAAAAAAAAAECAwQF/8QALBEBAAIIBAQYEBQUAAAAAAAAAAAECEQM
EEhMUITEyQVGHBSJhgZEjQpKx0f/aAAwDAQACEQMRAD8A+HgA7oAAAACGCAABUAAVQABAAFWAGCcF
wIJAaaaAaAAACgACgASUAAAABQAARQMFRJwii5U4J3ScolSRdr6mUKsqVXvCmuwnlV2GsUtq1VLtX
3lapj/wCZZT8o9y/En5U+xGv02fmesaY9j/gmz1jRH6r/AIF/Utv1kuxfcvYJWumuz7vyNRfT9GJrd
eQpX1Zfw1IuK6P1ZfxVoxq21Yuh4+2X51cdr3Lon+Mvz0ldTS+rFtPUZmrSSf0H/mIu6tmyf0Jf5sj
Aw2/qF0T/AJ/mXFfpTq49E19z/M711dD0/pwvo689pj8y2KrZL642f59i/oXdex12Wr/9FhrtXpzro
dE4fbF/mXdf6SNfH/A18a5/8jff0vKPaHh1Nn279uP5Szd1Q+vcvhZZL+pW9mVdd9y+KuMNX+k/WL
2qNNL92xf9xcR/ShY/b0NEvtf9UxxtP0hwjZ/iET1j8W/2Fzbs+rq1T+1pf6kWV2zas/3uL92/pP6p
Ho/0iUT9vZ1a98XH/iUP0w0M+nS2V/CT/ozpGppT3mHr0qbRHirPtLwlsJS4px173RRPP8ACy1t2Bj
pjD7dPfD+XAupbX2dY/qvtnHP4yTPSqymX91qEvdv7v8ApcTcU0rek/d76TfzYwzYi6EqH71qZQ190
keNmWZL/wBq392Vdn5G0qFqXBuxe6af+pP+Z5Ssgn68FB9rpw/4oCdj05+j1U692n2bLkuuUf8A7Kb
Ifik0eD0M+pxn7ozi393Sb3UozXqzb/Zt5T8J9At0S0nquxfr1Yf3r8jM/Donwy9NdDe7Pn06pR9qM
o/GLRRg3qzZuf8ADnD312b8ftTw/wACznsSux7sZQ1L6m7yVn3cG/uZ5r7FeqX0Jq1IGb1Po/KLe71
NLLi4uTXxSW8v4ftMXdpJwW845jnG/Fqcc/FH1tS1e80Exh4AAgAAIAAoAB5WgAkIAAoAAoAAAACg
ACiQAUAAAABQAAHtTqrK/Ys1HHZJoyWn9JNTDg5qxdK1/VYMODrTVvTw2mCJw2aHpFTZj1qMPPTJKe
Pf1NfezJaTVV2f2N8s/VU1Y/4Z419zNHJTPTTbbx4uvLtLrXWmO8PpFGotTxuRuS6eTe5YvYjCeP513X
bRqPmpOT6abYbsvjuyWfuNB003r6sKTV0F9GzLa+EulGy6HbdGpShJpS/wdQt+0f1ZdX4M9tNsrfp
n7S7ztEWhmrtjvd+asainlV2p31J+7L3ofGMLgxuq0GXm2PIWPhG7fzXLsirelfs2Zz1MyWnv1U01Y
4L/AA750cJL9Wzpj+91F9KTnMLTrnJcYOMXvLtx7Ni+HH3MmpWlu0YebUmHz3amzVCx12w5Gx53bVD
djJ9k4rh9seP6rMLqtNKqW7JccJpp5jKL6JRfWn2n0rU6am1fI9Q1UpR+anxcI9j3xjHP0XwXaug1
DUa0dd0t6r1Zxk+RufQpPjF5+rLh9+eLHztTT3ZcYs14FVkhGTjJOMotx1f8Gmnhp1JyAAAUAAEnk
UABQABQAAAAFAAFakAAACgACgAAAAAKABJQABcIAAuB19k7ft0/qSfK09dcvWx70bjodbv1crpp0yuP
rT0z9aVfH2opcce9ce1TPm5ebL21bpLVbVJqSfR1P3M601JjpPZJfR7boayrEsyq9pzjh2ad9C1NbX
tQXHK7M9GHFYbV7Ns1sbdFfu/LtDBS0964LU6dv2H710p9WxnoZ7V69RjDaem+br1PGqpSyqLHwd0V
1xfrKK6eHQ8NZ3Uaelv02qrxFTqmuHrwUXB78PfFepJdqcu03Pzd3C1sPm22YOSq1DTU7K921NYaug
3Bt+9qOfjkxZtnpdV6qeN17u/JdPrCrKLafXn0c9uTUzh5u1ZzAASGLAAPKoACgAAAAKAAKBIAAAFA
AFAAAAAUACSGADUQgCQbioAA1uoAExi28JZb4JLi2yYGxehuuVcr6prepn7Jzg+uMVma+2G99qj2H
0L0foa0dlGd6dMro1S+i42bzo4/sY+yxGkehmyVZ0/fe7XHT2wvu6Y178d3k12y4vo68H0fVSen0S5
0Cg7p1Vwg4pS3ZKEeP6zjGS93q9jN0jEZebWnrHjz0ssjKp2ReYz5CqDznehCL4r93cf2momym+mE4
xjp6YJKMYsxjrjF8mpfBuEn8MGtHOe7vTsAAjagAH1UAAAAFAAFakAAACgACgAAAAKABJQABYQJJIJO
```

```

1YAkFSR2rVlGBgu9Fs6694qq1PtawIr4t8F9psexPRyE9+Fnz+o9WUYVWwVdMYqTnys5YgsrGHvY4Y
xJvB1mmIy1FLt5NXo0srMYWE3u7zzhv6q62/css3DYnodJpTvk9LU1l5SWpsT4YinwrT6OPHil1m8b
E2JVCmZU1lxsUcTudme15g5pNQxhZit3K6eJ70UapRnVKMp728tZ0tzrh9H/wBPW+N0+00U16vHCcv
Z0VsQtqbsJWhr0dMKoxhTp60rnXiW/wAMdK9qXfTz6ZyaS3VxMT6R7VslyMt1RnN2LT1yksb7i1vya
+jCub1KS4Zw1w6bnelqbpKs7jsi75TmrJStUuHKWdMp9WIerBKSXGKRofprtxW6i2FcspR+TxcX6t
Wng8KuPvk1vSfv3ep5zMzDzbmbMBtvWK++UoNuuCjVXlYe5BYT92eL/eZYAHN6AAAUA8igAKAAKBI
AAFAAFAAAAUACSGAAAAARqESSiEe+ihXK2EbbHVU5YnZGv1XCPao5W98Mo61R6Vwp5Le1ZN2co48i
oYW5upqe8325WmdWTK7EnhSsjs1apV20Zm5Wbtae+3XLHXNR6eGNx4MbVufRpdjX0pvEfuj/V1/pdp
ThCdeKnvuDSjUp0vd3uC+hx3u0VJ8F0dfspDVektq2fs/VXwgrqYJ3WqVEeUnJcm8xVca4tJ+svabf
RjBnnp1p610ajTGiMG3KShGLfqKaj7Sy39CFf7RpC2pbKmuctRRo4Q3qoyqjG3VycfWcuHGHtcGtxc
OHQRpNow08+X5BKMoz+e1E52aq5uDUZQw1uYbTymuhLefXq1Zt3dp1Yb3yknXysktLRJyfl6nC5s3V
x1CEspY4etNTfamYfa+05SXqWz0tE8uWqn/fdWt3D+Txk8q07wds21hvDSW69ap25dxhCqPKS1Gvbc
VZdHpy+PqRy2nvvuVjhjLZkad3RyldqZ8vrU1ZZv/OR0kfoys3s79jeN2t8c41Po3ThqViHG1ss16V
7ajpdDCMYzq1mqpdXJPdj8m07W67Gopf0zgoLLSaTaSjlo+ZMvNqa+equndNtuT4Jycm12ZfS+vPW2
31lmct3DEIYDBiVAARVAAPKoACGSAAABQABQAAAAFAAkoAAAACgEAjUIku53VKMOTqe8oJWSskp5sy
8uKSSUcY4PPXxLQk6VRdUardshOcFdGE1J0zclXNL6L3Wml8MFDtcuC4JvhGKwvh7zxLxVQV3KcbaVK
Mq5qStXqquccNPefBPoPRW2DrK6ssqrm3p4y30G7dqIxVnQuiKbS4/H7CrZ2nepSk+VjU1GUrNXfVb
sWo5UcpPEnjCXa1xRbTdcW5Tl8osbbaTahntb6ZfZj4njqNXOzCk/Vj7MIpRHh4JcPtOu/OMQSy0tr
16WLhok1N8Ja2axbn/4/qfte1x4bvHOP1utjOuuquMoQjFSs3pbztvftWN4XDqS61ntZZNkH0206DI
BBxtKgDBzKAFAUA8qhIAAAFAFAAAAAUACSGAAAAKAACAANQJBBJqJEnq9RPk+S35cmpuxV7z3FNp
Jyx0Zwks+48STpEonIIBveEkAEmwEAGJkAAZAAAUeGHlaAAUAAUAAAABQAJKAAAAAoAAIAAoAAoEkE
lAAfYAANZQAAYAAIAAKAAA26j9F23LIRshs6UoTjGcZfKNMsxaynxn2M90ajb3dsvE6XzDpZYP9y0v
/S0f7cS/PmcWzeHKnNRt7u2XidL5g5qNvd2y8TpFM0qwOLYw5U5qNvd2y8TpFMHNrt7u2XidL5h1WB
xbGHKnNRt7u2XidL5g5qNvd2y8TpFM0qwOLYw5U5qNvd2y8TpFMHNrt7u2XidL5h1WC8axhypzU7e7
t14nS+Y0anb3dsvE6XzDqsxG39py00VGMVmyu7dnv8ArQnGGY4jh7/4JY4jjWM0a+anb3dsvE6XzBz
U7e7t14nS+YdB3+k1sIWQenitRCOIR5aUo2T3LJN9RNpbkM8Ome0xuqHpJarVW6KrN++EIzr1UVGM
JQqlNm0t6T5RtJcWovsHGsy9c9c103u7ZeJ0vmE8103u7ZeJ0vmHQdnPjbuQfIVVynXpbsS1MpKNdt1
cZb2K/UxGa4v34zhkT9J5yU4xpjVLk5uNkruEZKPCr2P7ddPJ/iONYw595qdvd2y8TpFMHNTt7u2Xi
dL5h0bXt5/J6rJQjCydnJzhbdiap9v1rGovdy4YWE020k2nkt6/SayctyGli3w6dQ0o5shBQ1828WJ
zWY9S62Xj2M0euanb3dsvE6XzBzU7e7t14nS+Yd07P1krq6pupwV1TnJ76kq5ppbnU30vj6PVnBrs
vS05KrE6bm7tTCzk4NqSjQdyEYJSbb5N5aXulWQ49jD4JzU7e7t14nS+Y0anb3dsvEaXzD79otvX2K
l0ypSnfyfclGG9vQcE3ZB5xLdbw8Zxh56Ge1G2NTvxUp1pT3Z1RnU4Sspm7PnHx4bsYxm+XSw8ZTHS
Yc+8103u7ZeI0vmDmp293bLxGL8w6B0npFOfyEmp14no42W6itJr1XGT1JR3niEeTknxfGcV8fXZ21
dS9XVp7t3140Uo7m7Ysx10Lay+CW7Fy6N7K6S8xYw555qdvd2y8RpfMHNTt7u2XiNL5h1UBzFzD1Xm
p293bLxGL8wc103u7ZeI0vmHVQHMXM0Veanb3dsvEaXzBzU7e7t14jS+YdVAcxcw5V5qdvd2y8RpfM
J5qdvd2y8RpfM0qQXmbm65W5qdvd2y8RpfMHNVt7u2XiNL5h1SBzNzdcrc1W3u7ZeI0vmDmq293bLx
GL8w6pA5m5urDYP9y0v/S0f7cS/APOoAAAAAAAAAABAAAMAACABIAAAAAAIJQAEgAAAAAAAAAAAA
AA/Z" type="image/x-icon" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" />
<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700;900
&display=swap" />
<link rel="stylesheet" href="{% static 'blog/css/mdb.min.css' %}" />
<link rel="stylesheet" href="{% static 'blog/css/styles.css' %}" />
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light" >
<div class="container">
<a class="navbar-brand me-2" href="/">

```

```


</a>

<button class="navbar-toggler" type="button" data-mdb-
toggle="collapse" data-mdb-target="#navbarButtonsExample" aria-
controls="navbarButtonsExample" aria-expanded="false" aria-label="Toggle
navigation">
  <i class="fas fa-bars"></i>
</button>

<div class="collapse navbar-collapse" id="navbarButtonsExample">
  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
    <li class="nav-item">
      <a class="nav-link" href="/#">Home</a>
    </li>

    <li class="nav-item">
      <a class="nav-link" href="{% url 'plotter-index'
%}">Plot</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{% url 'plotter-item_list'
%}">Inventory</a>
    </li>
  </ul>

  <div class="d-flex align-items-center">
    {% if user.is_authenticated %}
    <button type="button" class="btn btn-light me-3"><a href="{%
url 'logout' %}">Logout</a></button>
    <i class="fab fa-profile"></i></a> {% else %}
    <button type="button" class="btn btn-link px-3 me-2"><a
href="{% url 'login' %}">Login</a></button>
    <button type="button" class="btn btn-link px-3 me-3"><a
href="{% url 'register' %}">Sign up!</a></button> {% endif %}
  </div>
</div>
</div>
</nav>

<div class="container mt-2">
  {% if messages %}
    {% for message in messages %}
      <div class="alert alert-{{ message.tags }}">
        {{ message }}
      </div>
    {% endfor %}
  {% endif %}
  <div class="container">
    <div class="row">

```

```

        <div class="col-lg-8">
            {% block content %}{% endblock %}
        </div>
        <div class="col-lg-4 p-4">
            <div class="card">
                {% block sidebar %}{% endblock %}

            </div>

        </div>
    </div>
</div>
<footer class="bg-light text-lg-start">

    <hr class="m-0" />

    <div class="text-center p-3" style="background-color:
lightcyan">
        <h2 class="text-dark" href="/#"> © 2022 Fiber Plan 101</h2>

    </div>
</footer>

    <script type="text/javascript" src="{% static 'blog/js/mdb.min.js'
%}"></script>
    <script type="text/javascript"></script>

</body>
</html>

```

ii. Connect.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
    <form class="form-group" method="POST">
        {% csrf_token %}
        <legend class="border-bottom mb-4">Set Start, Integration Point</legend>
        <div class="form-group mb-4">
            {{ form|crispy }}
        </div>

        <!-- Submit button -->
        <button type="submit" class="btn btn-primary btn-block">Submit</button>
    </form>

</div>

<!-- Footer -->
<footer class="page-footer font-small blue pt-2 bg-blue text-center">

```

```

</div>

    <p>
        <i>
            <a class="text-center py-3"> Fast and Efficient Design || </a>
            <i>
            <i>
            <a class="text-center py-3">Accurate Qoutations || </a>
            </i>
            <i>
            <a class="text-center py-3"> Readily Integratable || </a>
            </i>
            <i>
            <a class="text-center py-3"> You are in charge of the design</a>
            </i>
        </p>
    </div>
</div>
</footer>
{% endblock content %}

```

iii. Index.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<form method="POST" action="{% url 'plotter-create' %}" class="row row-cols-lg-
auto g-3 align-items-center">
    {% csrf_token %}
    <div class="col-12">
        <label class="visually-hidden" for="inlineFormInputGroupUsername">Plot
name</label>
        <div class="input-group">
            <input type="text" name="name" class="form-control"
id="inlineFormInputGroupUsername" placeholder="Plot name" required />
        </div>
    </div>

    <div class="col-12">
        <button type="submit" class="btn btn-primary">Submit</button>
    </div>
</form>
<div class="card p-3 mb-3">
    <div class="list-group list-group-light">
        {% for plot in plots %}
        <a
            href="{% url 'plotter-populate' plot.id %}"
            class="list-group-item list-group-item-action px-3 border-0 ripple"
            >{{ plot.name }}</a>
        >
        {% empty %}
        <a
            href="#"
            class="list-group-item list-group-item-action px-3 border-0 ripple"
            >Sorry, no plots in this list.</a>

```

```

    >
    {% endfor %}
  </div>
</div>

{% endblock content %}

```

iv. Network.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
  <div class="card p-3 mb-3">
    <ul class="list-group list-group-light">
      <li class="list-group-item">Plot Name: {{ plot.name }}</li>
      <li class="list-group-item">Start Point is {{ plot.start }}</li>
      <li class="list-group-item">The Distance is {{ plot.distance }} m.</li>
      <li class="list-group-item">
        <a href="/media/{{plot.name}}.html">
          <button type="submit" class="btn btn-primary">
            <i class="fas fa-download me-2"></i>
            View Network Plot
          </button>
        </a>
      </li>
    </ul>
  </div>
</div>

<!-- Footer -->
<footer class="page-footer font-small blue pt-2 bg-blue text-center">

</div>
<p>
  <i>
    <a class="text-center py-3"> Fast and Efficient Design || </a>
    <i>
    <i>
    <a class="text-center py-3">Accurate Qoutations || </a>
    </i>
    <i>
    <a class="text-center py-3"> Readily Integratable || </a>
    </i>
    <i>
    <a class="text-center py-3"> You are in charge of the design</a>
    </i>
  </p>
</div>

</footer>
{% endblock content %}

```

```
{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
  <form method="POST">
    {% csrf_token %}
    <legend class="border-bottom mb-4">Enter plotting points</legend>
    <!-- 2 column grid layout with text inputs for the first and last names -
->

    <div class="row mb-4">
      <div class="col">
        <div class="form-outline">
          <input type="text" name="name" id="form3Example1" class="form-
control" required />
          <label class="form-label" for="form3Example1">Name</label>
        </div>
      </div>
      <div class="col">
        <div class="form-outline">
          <input type="text" name="coordinates" id="form3Example2"
class="form-control" required />
          <label class="form-label" name="coordinates"
for="form3Example2">Coordinates</label>
        </div>
      </div>
    </div>
    <div class="row mb-4">
      <div class="col">
        <div class="form-outline">
          <input type="text" name="name" id="form3Example1" class="form-
control" required />
          <label class="form-label" for="form3Example1">Name</label>
        </div>
      </div>
      <div class="col">
        <div class="form-outline">
          <input type="text" name="coordinates" id="form3Example2"
class="form-control" required />
          <label class="form-label" for="form3Example2">Coordinates</label>
        </div>
      </div>
    </div>
    <!-- Submit button -->
    <button type="submit" class="btn btn-primary btn-block mb-4">Plot </button>

  </form>
</div>

<!-- Footer -->
<footer class="page-footer font-small blue pt-2 bg-blue text-center">

</div>
```

```

    <p>
      <i>
        <a class="text-center py-3"> Fast and Efficient Design || </a>
      </i>
      <i>
        <a class="text-center py-3">Accurate Qoutations || </a>
      </i>
      <i>
        <a class="text-center py-3"> Readily Integratable || </a>
      </i>
      <i>
        <a class="text-center py-3"> You are in charge of the design</a>
      </i>
    </p>
  </div>
</footer>
{% endblock content %}

```

vi. Populate.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<form method="POST" class="row row-cols-lg-auto g-3 align-items-center">
  {% csrf_token %}
  <div class="col-12">
    <label class="visually-hidden"
for="inlineFormInputGroupUsername">Name</label>
    <div class="input-group">
      <input type="text" name="name" class="form-control"
id="inlineFormInputGroupUsername" placeholder="Name" required/>
    </div>
  </div>

  <div class="col-12">
    <label class="visually-hidden"
for="inlineFormInputGroupUsername">Coordinates</label>
    <div class="input-group">
      <input type="text" name="coordinates" class="form-control"
id="inlineFormInputGroupUsername" placeholder="longitude, latitude" required/>
    </div>
  </div>

  <div class="col-12">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
<div class="d-flex gap-2 my-3">
  <a class="btn btn-primary" href="{% url 'plotter-quote_item' plot.id %}">
    Quote
  </a>
  <a class="btn btn-success" href="{% url 'plotter-plot' plot.id %}">
    Plot

```



```

</a>
<a class="btn btn-warning" href="{% url 'plotter-more_settings' plot.id %}">
    More settings
</a>

<div class="flex-fill"></div>
<a class="btn btn-danger" href="{% url 'plotter-plot-delete' plot.id %}">
    Delete
</a>
</div>
<div class="card p-3 mb-3">
    <ul class="list-group list-group-light">
        {% for coordinate in coordinates %}
            <li class="list-group-item d-flex justify-content-between">
                {{ coordinate.name }} {{ coordinate.coordinates }}
                <a href="{% url 'plotter-coordinate-delete' coordinate.id %}">
                    <button type="button" class="btn btn-danger btn-sm btn-floating">
                        <i class="fas fa-trash"></i>
                    </button>
                </a>
            </li>
            {% empty %}
                <li class="list-group-item">Sorry, no coordinates in this plot.</li>
            {% endfor %}
        </ul>
    </div>
<div class="d-flex gap-2 my-3">
    <a class="btn btn-info">
        You can add a Maximum of 35 Points for this Plot Session
    </a>

</div>
<div class="d-flex gap-2 my-3">
    <a class="btn btn-primary" href="/media/{{plot.name}}.docx">
        Previous Quote
    </a>
    <a class="btn btn-success" href="/media/{{plot.name}}.html">
        Previous Plot
    </a>
</div>

{% endblock content %}

{% block sidebar %}
<form class="form-group" method="POST" action="{% url 'plotter-properties' plot.id %}">
    {% csrf_token %}
    <legend class="border-bottom mb-4">Plot properties (length in metres)</legend>
    <div class="form-group mb-4">
        {{ form|crispy }}
    </div>

    <!-- Submit button -->

```

```

    <button type="submit" class="btn btn-primary btn-block">Submit</button>
</form>

{% endblock %}

```

Vii. Quote_items.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
    <form class="form-group" method="POST">
        {% csrf_token %}
        <legend class="border-bottom mb-4">Item</legend>
        <div class="form-group mb-4">
            {{ form|crispy }}
        </div>

        <!-- Submit button -->
        <button type="submit" class="btn btn-primary btn-block">Submit</button>
    </form>

</div>

<!-- Footer -->
<footer class="page-footer font-small blue pt-2 bg-blue text-center">

</div>
    <p>
        <i>
            <a class="text-center py-3"> Fast and Efficient Design || </a>
            <i>
            <i>
            <a class="text-center py-3">Accurate Qoutations || </a>
            </i>
            <i>
            <a class="text-center py-3"> Readily Integratable || </a>
            </i>
            <i>
            <a class="text-center py-3"> You are in charge of the design</a>
            </i>
        </p>

</div>

</footer>
{% endblock content %}

```

vii. Quote_price.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}

```

```

<form method="POST" class="row row-cols-lg-auto g-3 align-items-center">
  {% csrf_token %}

<div class="card p-3 mb-3">
  <table class="table">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Category</th>
        <th scope="col">SKU</th>
        <th scope="col">Quantity</th>
        <th scope="col">Unit Price</th>
        <th scope="col">Price</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row">1</th>
        <td>Poles</td>
        <td>{{plot.pole|default:"-"}}</td>
        <td>{{poles}}</td>
        <td>{{plot.pole.unit_price}}</td>
        <td>{{poles_price}}</td>
      </tr>
      <tr>
        <th scope="row">2</th>
        <td>Fibre Cable</td>
        <td>{{plot.fibre_optic|default:"-"}}</td>
        <td>{{fibre_optic}}</td>
        <td>{{plot.fibre_optic.unit_price}}</td>
        <td>{{fibre_optic_price}}</td>
      </tr>
      <tr>
        <th scope="row">3</th>
        <td>Man hole</td>
        <td>{{plot.man_hole|default:"-"}}</td>
        <td>{{man_holes}}</td>
        <td>{{plot.man_hole.unit_price}}</td>
        <td>{{man_hole_price}}</td>
      </tr>
      <tr>
        <th scope="row">4</th>
        <td>Hand Hole</td>
        <td>{{plot.hand_hole|default:"-"}}</td>
        <td>{{hand_holes}}</td>
        <td>{{plot.hand_hole.unit_price}}</td>
        <td>{{hand_hole_price}}</td>
      </tr>
      <tr>
        <th scope="row">5</th>
        <td>Support Tangent</td>
        <td>{{plot.Support_Tangent|default:"-"}}</td>
        <td>{{Support_Tangent}}</td>

```

```

        <td>{{plot.Support_Tangent.unit_price}}</td>
        <td>{{support_tangent_price}}</td>
    </tr>
    <tr>
        <th scope="row">6</th>
        <td>ONU</td>
        <td>{{plot.onu|default:"-"}}</td>
        <td>{{onu}}</td>
        <td>{{plot.onu.unit_price}}</td>
        <td>{{onu_price}}</td>
    </tr>
    <tr>
        <th scope="row">7</th>
        <td>OLT</td>
        <td>{{plot.olt|default:"-"}}</td>
        <td>{{olt}}</td>
        <td>{{plot.olt.unit_price}}</td>
        <td>{{olt_price}}</td>
    </tr>
    <tr>
        <td colspan="4" style="text-align: center;">TOTAL</td>
        <td>{{total}}</td>
    </tr>
</tbody>
</table>
</div>

<div class="d-flex gap-2 my-3">
    <a class="btn btn-primary" href="/media/{{plot.name}}.docx">
        View Document
    </a>
    <a class="btn btn-success" href="/media/{{plot.name}}.html">
        Network Plot
    </a>
</div>
<!-- Footer -->
<footer class="page-footer font-small blue pt-2 bg-blue text-center">

</div>
    <p>
        <i>
            <a class="text-center py-3"> Fast and Efficient Design || </a>
            <i>
            <a class="text-center py-3">Accurate Qoutations || </a>
            </i>
            <i>
            <a class="text-center py-3"> Readily Integratable || </a>
            </i>
            <i>
            <a class="text-center py-3"> You are in charge</a>
            </i>
        </p>

```

```

</div>

</footer>
{% endblock content %}

```

viii. Item index.html

```

{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="d-grid gap-2 my-3">
  <a class="btn btn-success" href="{% url 'plotter-item_create' %}">
    Add item
  </a>
</div>
<div class="card p-3 mb-3">
  <table class="table">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">sku</th>
        <th scope="col">category</th>
        <th scope="col">unit price</th>
        <th scope="col">Actions</th>
      </tr>
    </thead>
    <tbody>
      {% for item in items %}
        <tr>
          <th scope="row">{{ item.id }}</th>
          <td>{{ item.sku }}</td>
          <td>{{ item.category }}</td>
          <td>{{ item.unit_price }}</td>
          <td class="d-flex">
            <a href="{% url 'plotter-item_update' item.id %}">
              <button type="button" class="btn btn-primary btn-sm btn-floating me-
2">
                <i class="fas fa-edit"></i>
              </button>
            </a>
            <a href="{% url 'plotter-item_delete' item.id %}">
              <button type="button" class="btn btn-danger btn-sm btn-floating">
                <i class="fas fa-trash"></i>
              </button>
            </a>
          </td>
        </tr>
      {% empty %}
        <tr>
          <td colspan="3">Sorry, no items in this list.</td>
        </tr>
      {% endfor %}
    </tbody>

```

```
</table>
</div>
{% endblock content %}
```

ix. Create_and_update_item.html

```
{% extends "plotter/base.html" %} {% load crispy_forms_tags %} {% block content %}
<div class="card p-3 mb-3">
  <form class="form-group" method="POST">
    {% csrf_token %}
    <legend class="border-bottom mb-4">Item</legend>
    <div class="form-group mb-4">
      {{ form|crispy }}
    </div>

    <!-- Submit button -->
    <button type="submit" class="btn btn-primary btn-block">Submit</button>
  </form>
</div>
{% endblock content %}
```

APPENDIX B: Item Price List

List of item prices*

1. Fiber Optic Cable | 12-core, Multimode OM4, (GYXTW) KSh 480.00
2. Fiber Optic Cable | 12-core, Multimode OM2, (GYXTW) KSh 170.00
3. Fiber Optic Cable | 8-core, Multimode OM2, (GYXTW) KSh 130.00
4. Fiber Optic Cable | 4-core, Multimode OM2, (GYXTW) KSh 80.00
5. Fiber Optic Cable | 6-core, Multimode OM2, (GYXTW) KSh 100.00
6. Fiber Optic Cable | 12-core, Singlemode, OS2, (GYXTW) KSh 100.00
7. Fiber Optic Cable | 8-core, Singlemode, OS2, (GYXTW) KSh 80.00
8. Fiber Optic Cable | 48-core, Singlemode, OS2, (GYTS) KSh 220.00
9. Fiber Optic Cable | 24-core, Singlemode, OS2, (GYTS) KSh 150.00
10. Fiber Optic Pigtail SC/UPC, Singlemode, 1.5m KSh 200.00
11. Fiber Optic Patch Panel/ODF: 24-port, Duplex KSh 5,000.00
12. Fiber Patchcord: 2LC/UPC-2LC/UPC, Singlemode, 3m KSh 700.00
13. Fiber Patchcord: 2LC/UPC-2LC/UPC, Singlemode, 1m KSh 500.00
14. Fiber Splitter -1:8, SC/UPC, ABS Box KSh 1,600.00
15. Fiber Splitter -1:64, SC/UPC, ABS Box KSh 11,000.00
16. Fiber Splitter -1:16, SC/UPC, ABS Box KSh 3,000.00
17. Fiber Splitter -1:32, SC/UPC, ABS Box KSh 5,500.00
18. Fiber Blank Insert, Duplex KSh 50.00
19. SC/UPC Duplex Fiber Adapter (Singlemode) KSh 150.00
20. Fiber Patchcord: 2LC/UPC-2LC/UPC, OM2, 1m KSh 500.00
21. Fiber Connector – SC/UPC (Fast Connector) KSh 250.00

*The prices are exclusive of VAT and were picked with permission from ‘Optace Fiber Optic Solutions’