

DEDICATION

The work is dedicated to my siblings Augustine and Thomas. Section Break(Next Page)

WPS Office

ACKNOWLEDGEMENTS

I express my deepest gratitude to the Almighty God, the source of all wisdom, strength, and understanding, for granting me the endurance and determination to undertake this industrial training in the field of software development. Without His guidance and grace, this endeavor would not have been possible. I acknowledge His divine providence, which sustained me through the challenges faced during this period, and allowed me to acquire invaluable skills and knowledge.

I extend my heartfelt appreciation to my family for their unwavering support, encouragement, and prayers throughout this journey. Their belief in my abilities has been a constant source of motivation. I am profoundly indebted to my supervisors, colleagues, and the entire team at Manful Computer Institute, for their guidance, mentorship, and the opportunity to be part of meaningful projects. Their expertise and willingness to share knowledge have significantly enriched my learning experience.

I would also like to acknowledge my academic institution for providing a platform to participate in the Student Industrial Work Experience Scheme (SIWES), a program that bridges the gap between academic knowledge and practical applications.

Lastly, I thank all those who, directly or indirectly, contributed to the successful completion of this training. Your support has been instrumental, and I am grateful for the roles you've played in shaping this transformative journey.

Section Break(Next Page)

TABLE OF CONTENT

DEDICATION	i
ACKNOWLEDGEMENT	ii
TABLE OF CONTENT	iii
LIST OF FIGURES	v
ABSTRACT	vi
CHAPTER ONE: Introduction	
1.1 Preamble	1
1.2 History of SIWES	1
1.3 Objectives of SIWES	3
1.4 Brief History of Manful Computer Institute	3
1.5 Mission Statement	4
1.6 Vision Statement	4
1.7 Products/Services Rendered at Manful Computer Institute	4
1.8 Organization of Manful Computer Institute	4
CHAPTER TWO: Brief Overview of the Project/Training Involved	
2.1 Overview	6
2.2 Objectives of Learning Software Development	6
2.3 Type of Work Done at Manful Computer Institute	7
2.4 Hardware and Software Used for Executing Task	7
2.4.1 Hardware Used	7
2.5.2 Software Used	7
2.5 Theoretical/Practical Knowledge from Resources Used for the Industrial Training	8
2.6 Overview of Web Development	8
2.6.1 Popular Frontend Technologies	9
2.6.2 Popular Backend Technologies	9
2.6.3 Web Applications	10

2.6.4	Web Development Phases	10
2.7	Overview of Django (Python Backend Framework)	11
2.7.1	Django	11
2.7.2	Django Models	12
2.7.3	Django Views	13
2.7.3	Django Templates	14
2.8	Other Django Concepts Learned	14
2.8.1	The Request/Response Cycle in Django	14
2.8.2	The Django Middleware	15
2.8.3	Django ORM	16
2.8.4	Features of Django	17
2.9	Introduction to PostgreSQL	17
2.9.1	What is PostgreSQL	17
2.9.2	Accessing PostgreSQL	18
2.9.3	Executing SQL Queries with PgAdmin	19
2.9.4	Database Design Procedures	21

CHAPTER THREE: Overall Information of the Industrial Training

3.1	Work Ethics Learned from my Supervisor and Colleagues during the Training Period	23
3.2	Comment on Task Performed	24
3.3	Overall Achievement of the Industrial Training	25

CHAPTER FOUR: Summary, Challenges Encountered, Recommendations and Conclusion

4.1	Summary	26
4.2	Challenges Faced and How I Overcame Them	26
4.3	Conclusion	27
4.4	Recommendation	27
4.5	References	28 -

LIST OF FIGURES

1.1	Organizational Structure of Manful Computer Institute	5
3.1	Backend Roadmap	10
3.2	Web Development Phases	10
3.3	Django Project Structure	12
3.4	Django Architecture	12
3.5	Django Model Creating a Userprofile	13
3.6	Django Model and Its Corresponding Table in the Database	13
3.7	Sample Django View	13
3.8	Sample Django Template	14
3.9	Request/Response Cycle through Django Application	15
3.10	Django Middleware Demonstration	16
3.12	Default Middleware List Provided by Django	16
3.13	Django ORM Structure	17
3.14	PgAdmin Home Screen	18
3.15	Writing SQL Queries in PgAdmin SQL Query Editor	21 -

This report provides a comprehensive summary of the Student Industrial Work Experience Scheme (SIWES) training carried out in the field of software development at Manful Computer Institute. The primary aim of this training was to offer students practical work exposure, thereby bridging the gap between theoretical knowledge and its real-world application. Chapter One serves as an introduction to SIWES, spotlighting its objectives and historical context, while underlining its significance in Nigerian universities. Chapter Two furnishes a detailed overview of the project and the practical competencies acquired throughout the training, which encompassed web development and database management using PostgreSQL. Chapter Three delves into the ethics instilled and tasks accomplished during the internship, shedding light on fundamental work principles and accomplishments as a backend developer intern. Lastly, Chapter Four delivers a summary of the overall experience, outlines encountered challenges, and provides recommendations for future trainees. It concludes by strongly advocating for SIWES as a pivotal element in preparing students for the professional realm.Section Break(Next Page).....

CHAPTER ONE

INTRODUCTION

1.1 PREAMBLE

The Students Industrial Work Experience Scheme (SIWES) is a skills training programme designed to expose and prepare students of universities and other tertiary institutions for the Industrial Work situation they are likely to meet after graduation. It is also a planned and structured programme based on stated and specific career objectives which are geared towards developing the occupational competencies of participants. Consequently, the SIWES programme is a compulsory graduation requirement for all Nigerian university students offering certain courses.

The Students Industrial Work Experience Scheme (SIWES), is the accepted training programme, which is part of the approved Minimum Academic Standard in the various degree programmes for all Nigerian Universities. The scheme is aimed at bridging the existing gap between theory and practice of Sciences, Agriculture, Medical Sciences (including Nursing), Engineering and Technology, Management, and Information and Communication Technology and other professional educational programmes in the Nigerian tertiary institutions. It is aimed at exposing students to machines and equipment, professional work methods and ways of safeguarding the work areas and workers in industries, offices, laboratories, hospitals and other organizations.

1.2 HISTORY OF SIWES

SIWES was founded in 1973 by ITF (Industrial Training Funds) to address the problem of tertiary institution graduates' lack of appropriate skills for employment in

Nigerian industries. The Students' Industrial Work Experience Scheme (SIWES) was founded to be a skill trainingSection Break(Next Page).....

WPS Office

programme to help expose and prepare students of universities, Polytechnics and colleges of education for the industrial work situation to be met after graduation.

This system facilitates the transfer from the classroom to the workplace and aids in the application of knowledge. The program allows students to become acquainted with and exposed to the experience required in handling and operating equipment and machinery that are typically not available at their schools.

Prior to the establishment of this scheme, there was a rising concern and trend among industrialists that graduates from higher education institutions lacked appropriate practical experience for employment. Students who entered Nigerian universities to study science and technology were not previously trained in the practical aspects of their chosen fields. As a result of their lack of work experience, they had difficulty finding work.

As a result, employers believed that theoretical education in higher education was unresponsive to the needs of labor employers. Thousands of Nigerians faced this difficulty till 1973. The fund's main motivation for establishing and designing the scheme in 1973/74 was launched against this context.

The ITF (Industrial Training Fund) organization decided to aid all interested Nigerian students and created the SIWES program. The federal government officially approved and presented it in 1974. During its early years, the scheme was entirely supported by the ITF, but as the financial commitment became too much for the fund, it withdrew in 1978. The National Universities Commission (NUC) and the National Board for Technical Education (NBTE) were given control of the scheme by the federal government in 1979. The federal government handed over supervision and implementation of the scheme to ITF in November 1984. It was taken over by the Industrial Training Fund (ITF) in July 1985, with the federal government bearing entire responsibility for funding.

Section Break(Next Page)

1.3 OBJECTIVES OF SIWES

To expose the students to work methods and techniques in handling equipment and machinery that may not be available in their universities;

To prepare students for the work situation they are likely to meet after graduation;

To provide an avenue for students in the Nigerian universities to acquire industrial skills and experience during their course of study;

To provide students with an opportunity to apply their theoretical knowledge in real work situation thereby bridging the gap between theory and practice.

To allow the transition phase from school to the world of working environment easier and

facilitate students' contact for later job placements.

1.4 BRIEF HISTORY OF MANFUL COMPUTER INSTITUTE

Manful Computer Institute (A division of Manful Technologies), is a private company registered in 2014 to carry out; training, sales, supply installation, computer maintenance and consultancy services on Information and Communication Technology (ICT). In a competitive field like ICT, where quality and acceptance is required, Manful Computer Institute, with her aggressive drive for excellence have succeeded immensely by creating a niche for herself through sheer determination and our customers-first/customers-gets-best policy.

Considering the vast marketing potentials in Nigeria and the need to foster a close working relationship with our teeming clientele, we decided to site our corporate headquarters in Uyo and plans are on ground to spread our branches across every state of the nation, Nigeria in the near future. With this policy, our target-oriented, dedicated and hardworking personnel are free to create room for innovative ideas, thus are involved in risk sharing thereby eliminating Red tapism. - Section Break(Next Page)-

1.5 Mission Statement

To completely eradicate computer illiteracy in our society

1.6 Vision Statement

To blow up the nation's economy through technological revolution from an intellectual

platform

1.7 PRODUCTS/SERVICES RENDERED AT MANFUL COMPUTER INSTITUTE

- **Professional Training**

At Manful, we grasp the essence of distinguishing oneself in a competitive landscape. Therefore, we utilize smooth technical approaches to educate and involve Government, private agencies, and NGOs in leveraging ICT for optimal productivity.

- **Web/Mobile Applications Development**

We specialize in crafting exceptional websites and mobile apps that empower our clients to present their businesses to a global audience. Our focus is deeply rooted in achieving both efficiency and captivating design.

- **Business Solutions**

We equip the management of organizations with real-time, actionable data that enables them to make business-critical decisions so as to accelerate rapid development within a short period of time.

1.8 ORGANIZATION OF MANFUL COMPUTER INSTITUTE

The organization structure of Manful Computer Institute as shown in Figure 1.1 is so defined as to allow for flexibility in terms of decision making at the lower echelon

of the company.Section Break(Next Page).....

WPS Office

This factor is what keeps us ahead of our rivals as our training experts are vested with prompt decision making as regards customer satisfaction.

As the organogram depicts, commands flows from the General Manager (GM) down the ladder, with the Gm being answerable to the Managing Director (MD) and the board. Only strategic decisions are made at this level.

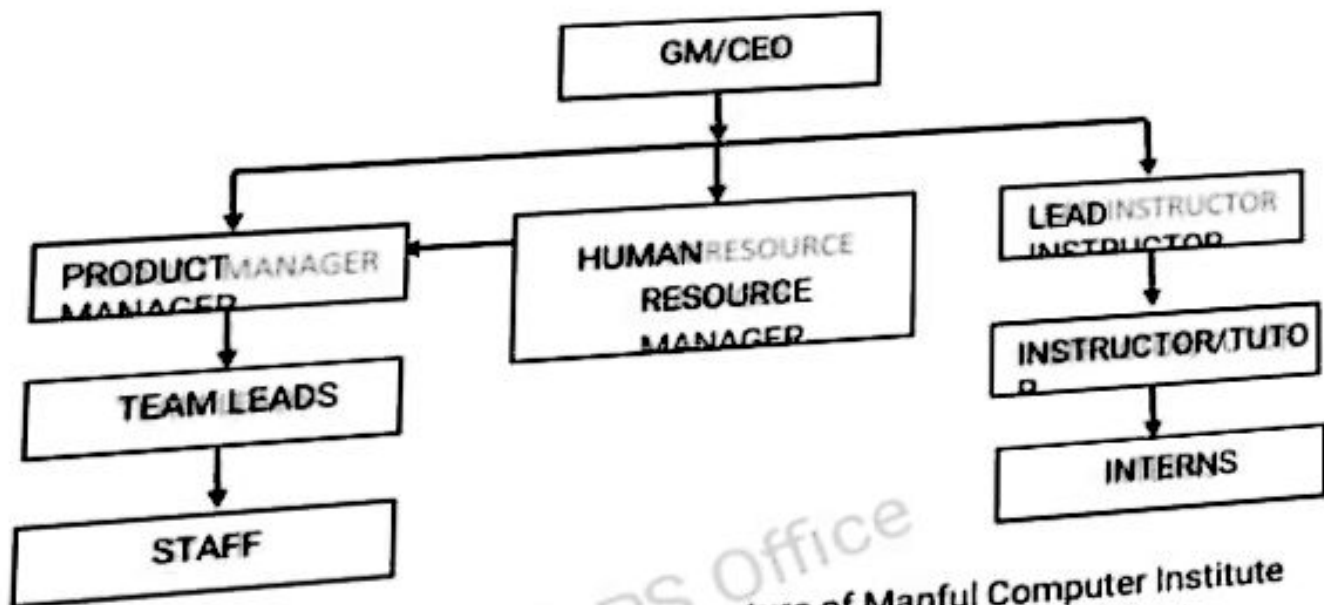


Figure 1.1: Organizational structure of Manful Computer Institute

CONCLUSION

To this resounding success lies in the choice of our personnel. Ours is a crop of dedicated, self-motivated and time-tested expertise.

We at Manful Computer Institute believe in excellence and do not compromise standard which is why our name best withstood the test of time. -Section Break(Next Page)-

CHAPTER TWO

BRIEF OVERVIEW OF THE PROJECT/TRAINING INVOLVED

2.1 Overview

Given the consistent rise in demand for software, particularly web applications, propelled by advancements in technology, shifts in consumer behavior, and evolving business requirements, I found it prudent and essential to acquire practical experience in software development.

2.2 Objectives of Learning Software Development.

- To gain a foundational understanding of different web technologies including HTML, CSS, and JavaScript, which are the building blocks of web development.
- To learn, practice and build software applications using server-side programming languages such as Python, to handle data, server interactions, and business logic.
- To familiarize myself with popular back-end frameworks and libraries such as Django and Flask to streamline development and enhance productivity.
- To acquire practical skills in database management systems (using PostgreSQL) and understand how to design, query, and manage databases.
- To learn how to manage servers, deploy and host web applications so as to make them accessible to users on the internet.
- To understand version control systems like Git and how to collaborate effectively with other developers on projects.
- To learn web security principles and best practices to protect applications and user data against potential threats and vulnerabilities.
- To understand the techniques for optimizing web apps performance.

- To learn how to work with external APIs, fetch and display data dynamically, and integrate third-party services into your web applications. -----

WPS Office

Development: At Manful Computer Institute, I really focused on Software development. I was involved in developing the backend of the project while adhering to industry best practices and coding standards as stated by my project manager.

Technologies Used:

- **Programming Language:** Python
- **Framework:** Django
- **Database:** PostgreSQL
- **Version Control:** Git

2.4 **HARDWARE AND SOFTWARE USED FOR EXECUTING TASK.**

Executing software development tasks often involves a combination of hardware and software tools to facilitate coding, testing, debugging, collaboration and project management. Below are some of the tools I used during the training period at Manful Computer Institute.

2.4.1 **Hardware Used**

- **Computer:** I used a laptop, which I installed an Integrated Development Environment (IDE) for writing, editing and debugging codes.
- **Mouse:** I use an external mouse to ease screen navigation.
- **Monitors:** We used a monitor to visualize our project locally before deployment.

2.4.2 **Software Used**

- **Integrated Development Environments (IDEs):** Visual Studio Code and PyCharm IDEs were mostly used for coding, debugging, and testing.
- **Version Control Systems (VCS):** Git was used to manage and track changes in the source code during development, enabling collaboration and history tracking.

- **Collaboration and Communication with team members:** Slack, Microsoft Teams, were mostly used for team communication and collaboration during development.
- **Database Management Systems (DBMS):** PostgreSQL and PgAdmin Software were used to design, manage, and interact with databases used in applications.
- **Virtualization and Containerization:** Docker, VirtualBox and python virtual environments were used to create and manage isolated environments for development, testing, and deployment. Although, I was only introduced to Docker, but did not eventually use it.
- **Code Review Collaboration:** GitHub platform was used to host code repositories, conduct code reviews, and facilitate collaboration with team members.

2.5 THEORETICAL/PRACTICAL KNOWLEDGE FROM RESOURCES USED FOR THE INDUSTRIAL TRAINING.

A rundown of activities performed during the course of the industrial training is given below

- Overview of Web Development
- The frontend and Backend Development
- Web Applications
- Django
- Using PgAdmin with PostgreSQL
- Version Control using Git

2.6 OVERVIEW OF WEB DEVELOPMENT

Reintroduction of the concept of web development was deemed necessary by my instructor. Web development refers to the creating, building, and maintaining of web applications. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet.

Web Development can be classified into Frontend Development and Backend Development. Section Break(Next Page)

WPS Office

Frontend and Backend are the two most popular terms used in web development. The part of a website where the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application. Backend is the server side of a website. It is part of the website that users cannot see and interact with. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

2.6.1 Popular Frontend Technologies

- **HTML:** HTML stands for HyperText Markup Language. It is used to design the front end portion of web pages using markup language. It acts as a skeleton for a website since it is used to make the structure of a website.
- **CSS:** Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable.
- **JavaScript:** JavaScript is a scripting language used to provide a dynamic behavior to our website.
- **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular CSS framework for developing responsive, mobile-first websites.

2.6.2 Popular Backend Technologies

- **PHP:** A server-side scripting language designed specifically for web development.
- **Java:** Java is one of the most popular and widely used programming languages.
- **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside a browser. -----Section Break(Next Page)-----

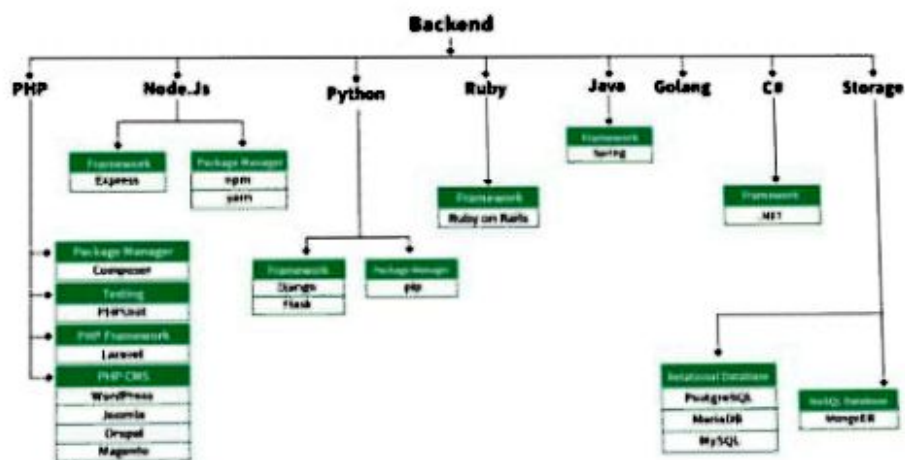


Figure 3.1: Backend
Roadmap Source:
GeeksforGeeks

Web applications: Web applications are the software deployed and implemented over the web. The concept of web applications comes from the web development solutions which used to run on the desktop. That desktop software needed to be downloaded on the individual systems, whereas web applications are deployed on the web and anyone can access the software without actually downloading them on the physical storage. The web application can also be called as a web-based software. Popular Web Apps: Facebook, Gmail, Google Drive, Google (Slides, Docs, Sheets), LinkedIn, Instagram, X, WhatsApp, Discord.

WEB DEVELOPMENT PHASES



**Figure 3.2: Web Development
Phases**
Source: Semidot Info Tech

WPS Office

All scale of web development is distributed into the following phases:

Planning

- What is the main purpose of the web project or website?
- Who are the users of the website?
- What is the required computing environment of the website?
- Who owns the website and who authorizes the information?

Analysis

- What information is **useful** for the end-user of the website?
- Where do you want the users to route through your website?

Design and Development

- What is the recommended UI/UX?
- What is the preferred layout for the website?
- Which multimedia can we use in the website?

Testing

- Are images and content of the website are appropriate?
- Are website components working correctly?
- Are users able to find the relevant information?
- Are website navigations working correctly?

Implementation and Maintenance

- How the website will be deployed?
- How to update the website?
- Who will update the content and how frequently?
- Website development versus web application

2.7 OVERVIEW OF DJANGO (PYTHON BACKEND FRAMEWORK)

2.7.1 Django

Django is an open source python framework, and Django Architecture follows structure. In MVT, M stands for **Model**, which is used to create table and their fields, a is represent with a Python class, with the class attributes as database columns

for **views**, which is a Python function that accepts a Web request and delivers a Web page. **T** stands for **Templates** contains the static content of a Django project like HTML, CSS, Javascript, along with the image used in the



Figure 3.3: Django Project

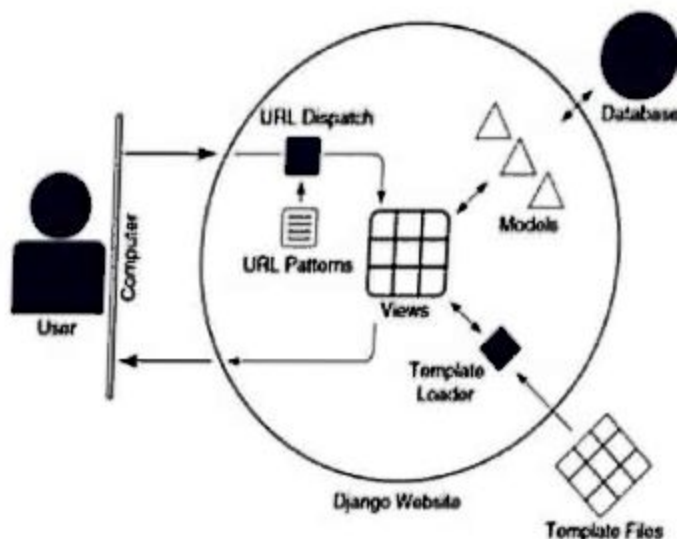


Figure 3.4: Django

Source: Research

2.7.2 Django Models

Django applications use Python objects called models to access and manage data. These models define the internal structure of the storing data, like field types and their maximum size, default values, label text for forms, etc. Models creation is independent of which database you are using. You can use any Database. You don't need to talk to the database directly. You only need to create your model structure and other code, and Django will take care of the dirty work of talking with the database.

Section Break(Next Page)

```

1 from django.db import models
2
3 # import the default Django Model
4 from django.db import models
5
6 # declare a new model with a name
7 class UserProfile(models.Model):
8     # fields of the model (translated to database fields)
9     username = models.CharField(max_length=100, unique=True)
10    first_name = models.CharField(max_length=32)
11    last_name = models.CharField(max_length=32)
12    email = models.EmailField(max_length=200, unique=True)
13
14    # renames the instances of the model
15    def __str__(self):
16        return self.username

```

Figure 3.5: Django Model Creating a UserProfile (Content of models.py file)

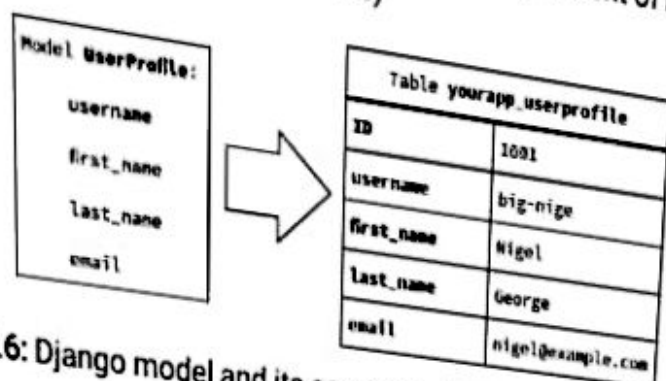


Fig. 3.6: Django model and its corresponding table in the database.

Source: Mastering Django

2.7.3 Django Views: A view function in Django is a Python function that accepts a Web request and delivers a Web response. The response in Django views can be anything that a web browser can show, including the HTML of a Web page, a redirect, an XML document, a 404 error, an image, etc. Views in Django are part of the user interface in a Django application because they return the web pages which contain HTML/CSS/JAVASCRIPT in the template of the Django

```

1 # views.py
2 # ListOfUsers
3
4 # import the default Built in generic list view
5 # Provided by Django
6 from django.views.generic.list import ListView
7
8 # Importing the UserProfile Model
9 # Defined in the models.py file
10 from .model import UserProfile
11
12 # This view will return all users in the database
13 class ListOfUsers(ListView):
14     model = UserProfile
15     template_name = "templates/user_list.html"
16     context_object_name = "all_users"

```


Figure 3.7: Sample Django ViewSection Break(Next Page).....

WPS Office

7.4 Django Template

Template in Django contains the static content of a Django project like Html, CSS, and Javascript, along with the image used in the project. We can set the path of the Django template from the setting.py file of the Django project.

```
<> users.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width,
7          initial-scale=1.0">
8      <title>First app</title>
9  </head>
10 <body>
11     <h1>List of Users</h1>
12
13     <!-- Using a loop to access database users
14          using the variable specified in the view
15     -->
16     {% for user in all_users %}
17         <p> {{ user.username }} | {{ user.firstname }} </p>
18     {% endfor %}
19 </body>
20 </html>
```

Figure 3.8: Sample Django Template

2.8 OTHER DJANGO CONCEPTS LEARNED

2.8.1 The Request/Response Cycle in Django

All backend technologies and web APIs are based on one system that is The Request-Response Cycle. The system is responsible for data interchange between client and servers. Request/Response objects are transmitted over the web. Content like images, HTML, CSS, JavaScript is all Response Objects. When a page is requested, Django creates an HttpRequest object that contains metadata about the request. Then Django loads the appropriate view, passing the HttpRequest as the first argument to the view function. Each view is responsible for returning an

HttpResponse object. Figure 3.9 clearly explains the Django Section Break(Next Page).....

WPS Office

Request/Response Cycle. Well, it does that on a very high-level perspective. In this figure, both the client and server are using HTTP. If both of them don't have the same protocol, the connection cannot be done. Just like humans need to have a common language to understand each other. The same way, various computers and machines over the internet need to have the same protocol. At least the one through which they communicate.



Figure 3.9: Request/Response Cycle through Django Application

The Django Middleware

Django middleware is basically a mechanism that comes in the *middle* of the usual request- response cycle, usually to provide some sort of intermediate functionality. A middleware component takes an HTTP request, performs some operation upon it, and then passes it on to the next component in line, which may be either another middleware or the final view. Middleware is always runs in the background while the request and response are being processed. Figure 3.10 represents a typical Django Middleware. ----- Section Break(Next Page) -----

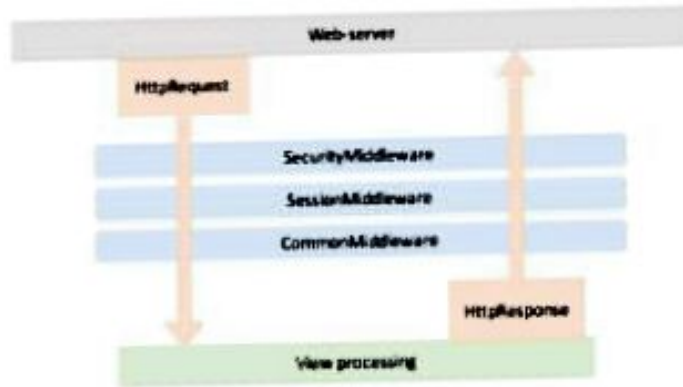


Figure 3.10: Django Middleware demonstration

Middleware can also return a response directly instead of forwarding the request further down the chain. For example, Django's CSRF Middleware checks if the request comes from a valid origin (usually the same domain as the server) and immediately returns with a 403 Forbidden response if not. This helps prevent a potentially malicious request from entering the system. Figure 3.12 shows the default middleware setting in Django

```

27
28 MIDDLEWARE = [
29     'django.middleware.security.SecurityMiddleware',
30     'django.contrib.sessions.middleware.SessionMiddleware',
31     'django.middleware.common.CommonMiddleware',
32     'django.middleware.csrf.CsrfViewMiddleware',
33     'django.contrib.auth.middleware.AuthenticationMiddleware',
34     'django.contrib.messages.middleware.MessageMiddleware',
35     'django.middleware.clickjacking.XFrameOptionsMiddleware',
36 ]
37

```

Figure 3.12: Default middleware list provided by Django

2.8.3 DJANGO ORM

ORM stands for Object-Relational Mapping. It's a programming technique and a

intuitive and developer-friendly. Figure 3.13 shows how ORM translates python codes to create SQL queries.

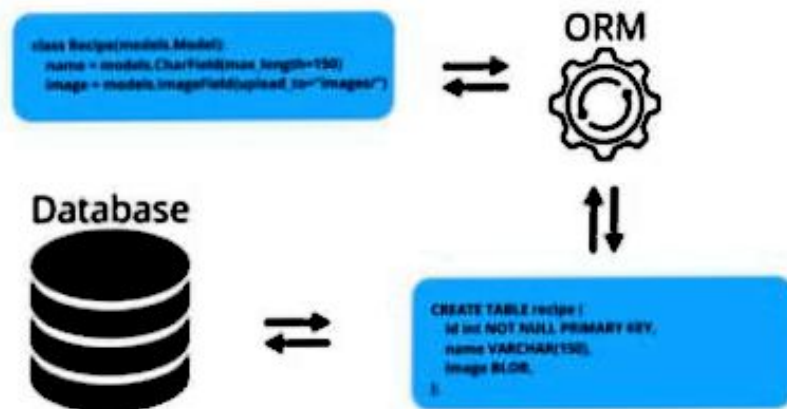


Figure 3.13: Django ORM Structure

2.8.4 Features of Django

Django is open source python framework, and Django Architecture follows the MVT structure. Many features make Django a popular framework. A few of them are listed below.

- Python based Web-framework
- Well-written Documentation
- Scalable in Nature
- Versatile in Nature
- Secured
- Provides Rapid Development

2.9 INTRODUCTION TO POSTGRESQL

2.9.1 What is PostgreSQL:

PostgreSQL (pronounced as **post-gress-Q-L**) is an open source relational database management system (DBMS) developed by a worldwide team of volunteers. PostgreSQL is not controlled by any corporation or other private entity and the

source code is available freeSection Break(Next Page).....

WPS Office

of charge. PostgreSQL runs on all major operating systems, including Linux, UNIX (Mac OS X, Solaris), and Windows. It supports text, images, sounds, and video, and includes programming interfaces for C / C++, Java, Perl, Python, Ruby, Tcl and Open Database Connectivity (ODBC).

2.9.2 Accessing PostgreSQL

There are several methods, including command-line tools, GUI-based applications, and programming libraries. Here are the common ways to access and interact with PostgreSQL:

- i. **Command-Line Interface (CLI) using psql:** psql is the PostgreSQL interactive terminal, allowing you to execute SQL commands, manage databases, and perform various administrative tasks from the command line.

CONNECTING TO PostgreSQL using PgAdmin (CLI)

```
psql -U username -d databaseName -h Host  
t -p Port
```

EXAMPLE: *psql -U brightagbor -d bookdb -h localhost*

calhost -p 5432

- ii. **Graphical User Interface (GUI) Clients:** pgAdmin4 is a popular application to manage PostgreSQL databases. All types of PostgreSQL features are supported by this application. The user can easily create a new user or role in postgres database by using pgAdmin tree control. pgAdmin is a powerful and flexible tool that simplifies database management and development for PostgreSQL databases, making it a popular choice among developers, database administrators, and data

analysts. Its user-friendly interface and rich feature set contribute to efficient database operations and optimal performance. pgAdmin makes it easier to manage, develop, and administer databases. ----- Section Break(Next Page)-----

WPS Office



Figure 3.14: pgAdmin Home Screen

2.9.3 EXECUTING SQL QUERIES WITH PGADMIN

Writing queries in pgAdmin, the PostgreSQL administration and development platform, involves using the SQL query editor to create, modify, and execute SQL (Structured Query Language) statements. Here's a step-by-step guide on how to write queries using pgAdmin:

- **Connect to a Database:** Open pgAdmin and connect to the PostgreSQL server where your target database resides. Navigate to the "Servers" or "Browser" panel and expand the server to view the databases.
- **Select the Database:** Expand the database you want to work with and navigate to the "Schemas" > "public" to access the database schema.
- **Open the SQL Query Editor:** Right-click on the "SQL Queries" under the desired schema and choose "Query Tool" to open the SQL query editor.
- **Write SQL Queries:** In the SQL query editor, start writing your SQL queries. Here

are some common SQL statements to get you started: Section Break(Next Page)

WPS Office

Sql – Select statement to retrieve data from a table

SELECT * FROM your_table;

Sql – Insert statement to add data to a table

INSERT INTO your_table (column1, column2) VALUES ('value1', 'value2');

– Update statement to modify data in a table

UPDATE your_table SET column1 = 'new_value' WHERE condition;

– Delete statement to remove data from a table

DELETE FROM your_table WHERE condition;

- **Execute Queries:** After writing your query, click the "Execute/Refresh" button or press 'F5' to run the query. The query results will be displayed in the "Data Output" or "Messages" tab.
- **Query History and Management:** pgAdmin keeps track of executed queries in the "History" panel, allowing you to review and re-execute previous queries.
- **Query Formatting:** Use the "Beautify" button to format your SQL query for better readability and organization.
- **Saving Queries:** Save your queries using the "Save" button or by selecting "File" > "Save" from the menu. You can save them for future use or share them with others.
- **Query Execution Plan:** Analyze the query execution plan using the "Explain" option to optimize and understand how PostgreSQL processes the query.
- **Advanced SQL Features:** Utilize advanced SQL features such as subqueries, joins, aggregates, functions, and more to craft complex and efficient queries.

With pgAdmin's SQL query editor, you can easily create, edit, and execute SQL queries against your PostgreSQL databases, allowing you to interact with and manage your data

effectively. This is shown in Figure 3.15. -----Section Break(Next Page)-----

WPS Office

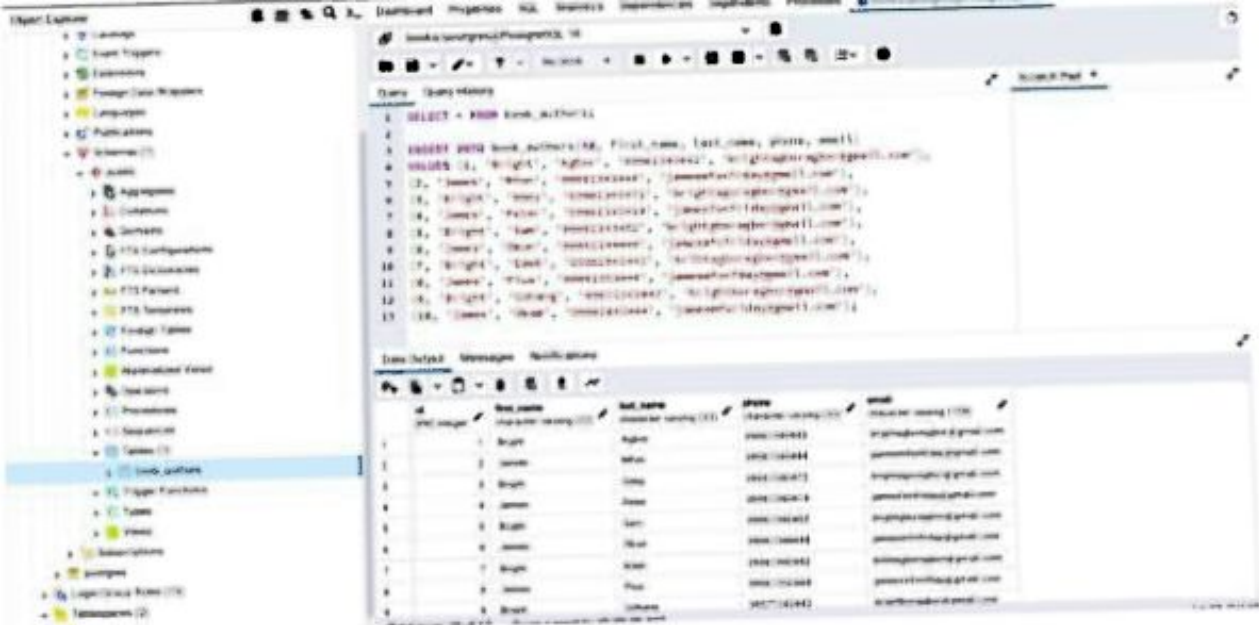


Figure 3.15: Writing SQL queries in pgAdmin SQL query editor

3.4 DATABASE DESIGN PROCEDURES

At Manful Computer Institute, I was also reintroduce to the steps involved in designing a database with PostgreSQL which involves several important steps to ensure efficient storage, retrieval, and management of data. Here's a step-by-step approach taken to design a database using PostgreSQL and pgAdmin.

- **Requirement Analysis:** Understand the requirements of the system or application that will use the database. Work closely with stakeholders to gather information about data entities, relationships, and constraints.
- **Conceptual Design:** Create a high-level, conceptual model of the database. Identify the main entities, their attributes, and the relationships between them. Use techniques like Entity-Relationship Diagrams (ERDs) to visualize this conceptual model.
- **Normalization:** Apply normalization techniques (e.g., 1NF, 2NF, 3NF) to ensure the database is free of redundancy and anomalies. Normalization helps optimize

the structure of the database and minimizes data duplication. Section Break(Next Page)

WPS Office

- **Logical Design:** Translate the conceptual model into a logical model. At this stage, we actually used Django models to create the class which translates to database tables and its attributes translating to columns. In other words, Django models was used to define tables, columns, primary keys, foreign keys, and data types for each entity. We also ensured the design follows normalization rules.
- **Indexing and Constraints:** Define indexes for columns that will be frequently searched or used in joins to improve query performance. Add constraints like unique constraints, check constraints, and foreign key constraints to maintain data integrity.
- **Optimization:** Continuously monitor and optimize the database design based on real-world usage patterns. Optimize queries, indexes, and other database objects to improve performance.
- **Documentation:** Create comprehensive documentation that includes database schemas, relationships, constraints, indexing strategy, backup procedures, and any other relevant information for future reference and maintenance.

CHAPTER THREE

OVERALL INFORMATION OF THE INDUSTRIAL TRAINING

3.1 WORK ETHICS LEARNED FROM MY SUPERVISOR AND COLLEAGUES DURING THE TRAINING PERIOD

Work ethics for a software developer are crucial for maintaining professionalism, achieving success, and building trust with colleagues and clients. Here are some key principles and practices I gained while learning software development during the period of internship at Manful Computer Institute.

- **Honesty and Integrity:** I have gained an understanding of the importance of honesty and openness in my actions and communications. Additionally, I have learned to acknowledge errors and be accountable for my work at all times.
- **Respect:** Maintaining a consistent approach of treating coworkers, clients, and users with courtesy and professionalism is essential. Actively listening and appreciating diverse viewpoints and ideas are values I've embraced.
- **Accountability:** To take ownership of my work and meet deadlines. And to be accountable for the quality and performance of the software you develop.
- **Professionalism:** I have also learned to follow company policies, guidelines dress code and to always act professionally in the workplace.
- **Confidentiality:** Respecting and safeguarding the confidentiality of sensitive information and proprietary data is a priority. I've also learned to refrain from discussing or sharing such information beyond the workplace.
- **Communication:** Clearly communicate ideas, progress, and challenges and constructive feedback to colleagues and stakeholders. -----Section Break(Next Page)-----

- **Continuous Learning:** Stay updated with industry trends and advancements in technology, which includes dedicating time to learning new skills and improving existing ones.
- **Collaboration:** Collaborate effectively with team members and other stakeholders.
- **Efficiency and Productivity:** Manage my time effectively and prioritize tasks based on their importance and urgency.
- **Client Focus:** To always understand and prioritize the needs of the client or end-users, in order to create software solutions that meet or exceed their expectations.
- **Adherence to Standards:** I have learned to follow coding standards, best practices, and guidelines set by your team or organization. This include writing clean, well-documented, and maintainable code.
- **Avoiding Plagiarism:** I have gained an understanding of the importance of not duplicating or utilizing someone else's work without giving appropriate credit and, when required, obtaining permission.
- **Environmental and Social Responsibility:** To always consider the environmental and social impact of your work and strive to create sustainable and responsible solutions.

3.2 COMMENT, ON TASK PERFORMED

At Manful Computer Institute, while on industrial training, I delivered a solid performance as a backend developer intern, showcasing a strong grasp of programming principles and a keen eye for detail. I demonstrated proficiency in designing and implementing efficient backend solutions, contributing positively to the overall project. Exhibited a proactive approach to problem-solving and collaborated effectively with the team to achieve project objectives. Showed

dedication to learning and adapting to new technologies, enhancing skills and adding value to the team's efforts. -----Section Break(Next Page)-----

WPS Office

3.3 OVERALL ACHIEVEMENT FROM THE INDUSTRIAL TRAINING

Here are my key achievements during the Industrial Training period as a Backend Software Developer Intern:

- **Effective Database Design and Management:** Implemented optimized database structures and efficient management techniques, enhancing application performance and data retrieval.
- **Scalable Backend Solutions:** Engineered scalable backend solutions, ensuring the application's ability to handle increased load and user demand while maintaining performance.
- **Collaborative Project Contributions:** Actively participated in collaborative projects, providing valuable input and expertise to the team, resulting in successful project deliveries.
- **Troubleshooting and Debugging Proficiency:** Developed strong skills in identifying and resolving bugs and issues within the backend codebase, contributing to a stable and reliable application.
- **Adoption of Best Practices:** Adhered to industry best practices in coding, documentation, and code reviews, promoting maintainability and code quality.
- **Client Communication and Requirement Understanding:** Developed excellent client communication skills, ensuring a clear understanding of requirements and successful alignment of backend solutions with client needs.
- **Continuous Learning and Skill Growth:** Demonstrated a commitment to continuous learning, staying updated with the latest technologies and methodologies in backend development, enhancing overall skills and knowledge.
- **Meeting Project Deadlines:** Consistently met project deadlines, showcasing effective time management and organizational skills to deliver high-quality work

on time.Section Break(Next Page).....

WPS Office

CHAPTER FOUR

SUMMARY, CHALLENGES ENCOUNTERED, RECOMMENDATIONS AND CONCLUSION

4.1 SUMMARY

The industrial training in software development provided invaluable hands-on experience, allowing me to apply theoretical knowledge to real projects. It significantly enhanced my skills, offering a glimpse into the dynamic world of software development and better prepared me for a future career in this rapidly evolving field.

4.2 CHALLENGES FACED AND HOW I OVERCAME THEM

- **Financial Constraints and Transportation Costs:** I successfully managed financial difficulties, especially dealing with expensive transportation. This necessitated meticulous budgeting and scheduling to maintain regular attendance and punctuality throughout the internship.
 - **Unpaid Internship and Financial Strain:** I was able to handle the financial pressure of an unpaid internship by innovatively managing to meet living expenses and essential needs, all while gaining valuable professional experience.
 - **Time management and multitasking.** I had to improved time management skills to balance work commitments with personal responsibilities effectively, maximizing productivity and achieving project goals despite time constraints.
- Section Break(Next Page) -----

4.3 CONCLUSION

The Student Industrial Work Experience Scheme (SIWES) introduced me to practical work experiences and essential skills required to succeed in the job market. It allowed me to bridge the gap between theoretical knowledge acquired in school and its practical application. Throughout the training and continuous engagement with the company's staff, I gained a profound appreciation for the education I received its potential to positively contribute to society.

Furthermore, it was an enriching period that I believe is starting to influence my professional life. I am deeply grateful for this dedicated time that allowed me to gain valuable skills, knowledge, meaningful connections, and lasting relationships that could significantly shape my career trajectory.

4.4 RECOMMENDATION

After participating in the Student Industrial Work Experience Scheme (SIWES) and personally witnessing its advantages, I strongly urge every Nigerian student to value and utilize this opportunity. Embracing SIWES not only prepares students for the job market but also contributes to decreasing the population of inadequately skilled graduates in Nigeria. -----Section Break(Next Page)-----

TITLE PAGE

400 LEVEL SIWES REPORT

COURSE CODE: GRE 421 [STUDENT INDUSTRIAL WORK EXPERIENCE SCHEME (IT)]

COMPANY ATTACHED: MANFUL COMPUTER INSTITUTE UYO, AKWA IBOM STATE

INDUSTRY-BASED SUPERVISOR: MISS EDIDIONG JEREMIHA

AGBOR, BRIGHT AGBOR

18/EG/CO/1172

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF UYO

WPS Office

SUBMITTED TO

ENGR. C. S. ANEKE - SIWES COORDINATOR

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF UYO, UYO, AKWA IBOM STATE.

PERIOD OF INTERNSHIP: MAY 2023 - SEPTEMBER 2023

4.5 REFERENCES

1. J. Forcier, P. Bissex et al., "Python Web Development with Django".
2. Understanding Django Architecture | Model View Template (MVT) Architecture. Retrieved June 10 2023, from <https://www.scaler.com/topics/django/django-architecture/>
3. Django Architecture Seen as MVT. Retrieved June 10 2023, from https://www.researchgate.net/figure/Django-Architecture-seen-41_fig4_3568331
4. Request and Response Retrieved August 22 2023, from Django 4.2 documentation, <https://docs.djangoproject.com/en/4.2/ref/request-response/>
5. A Comprehensive Guide to Django Middleware. Retrieved 29 September, <https://dev.to/bikramjeetsingh/a-comprehensive-guide-to-django-middleware-2fee>