# A Summer Internship report

on

“FULL STACK DEVELOPMENT(Web, Java & Python)”

Submitted in partial fulfillment of the requirement

for the award of the degree of

**Bachelor of Technology**

**in**

**Artificial Intelligence and Data Science**

**By**

THAVISHALA CHARPITHA SREE

(20701A0527)



**Department of Computer Science And Engineering**

**Annamacharya Institute of Technology and Sciences**

(An Autonomous Institution)

New Boyanapalli, Rajampet-516126 Kadapa (Dt), A.P.

Approved by AICTE, New-Delhi and affliated to J.N.T.U.A, Anantapuramu

Accredited by NBA & NAAC with A Grade

**2023-24**

**Department of Computer Science And Engineering**

**Annamacharya Institute of Technology and Sciences**

(An Autonomous Institution)

New Boyanapalli, Rajampet-516 126 Kadapa (Dt), A.P.

Approved by AICTE, New-Delhi and affliated to J.N.T.U.A, Anantapuramu

Accredited by NBA & NAAC with A Grade



**CERTIFICATE**

This is to certify that the summer Internship report entitled “**FULL STACK WEB DEVELOPMENT(Web,Java & Python)**” is submitted by

Name: THAVISHALA CHARPITHA SREE,

Roll Number: 20701A0527in partial fulfillment of the requirements for the award of Degree of **Bachelor of Technology** in **“Computer Science And Engineering”** for the academic year 2023-24.

**Summer Internship Supervisor Head of Department**

**Mr. C.V.LAKSHMI NARAYANA Dr. M.SUBBA RAO**

Assistant Professor of Associate Professor & Head of

Computer Science And Computer Science And

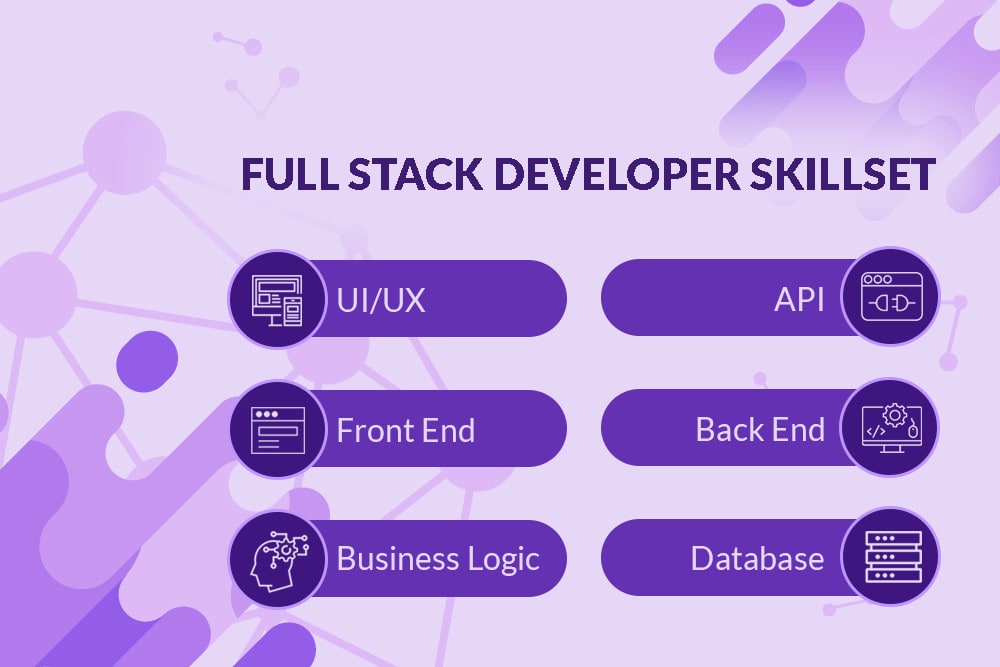
Engineering Engineering

ABSTRACT

*This internship report encapsulates a comprehensive exploration of my journey as a Full Stack Development intern, where I worked with a diverse stack comprising web technologies, Java, and Python. The report delves into the hands-on experiences and skills acquired during the internship, including front-end development using HTML, CSS, and JavaScript, back-end development using Java Spring Boot, and Python-based data analysis. It discusses the real-world projects undertaken, emphasizing their relevance and impact. The report also highlights the significance of teamwork, communication, and problem-solving in a dynamic development environment. Moreover, it provides insights into the challenges encountered and the strategies employed to overcome them. In conclusion, this internship report sheds light on the invaluable learning experiences and growth achieved while working as a Full Stack Development intern.*

CONTENTS

* Abstract
* Introduction
* Methodology
* Front-End Development
* What Does a Front-End Do?
* Front-End Advantages
* Back-End Development
* What Do Back-End Do?
* Key Back-End Development Concepts
* Back-End Advantages
* Conclusion
* References



**INTRODUCTION**

A Full Stack Python Developer is a software professional who specializes in developing

applications using the Python programming language. They are responsible for designing, coding,

testing, and deploying software solutions

A Full Stack Python Developer is a software professional who specializes in developing

applications using the Python programming language. They are responsible for designing, coding,

testing, and deploying software solutions

A Full Stack Python Developer is a software professional who specializes in developing

applications using the Python programming language. They are responsible for designing, coding,

testing, and deploying software solutions

A Full Stack Python Developer is a software professional who specializes in developing

applications using the Python programming language. They are responsible for designing, coding,

testing, and deploying software solutions

Full-stack development is a comprehensive approach to web development that involves working on both the front-end and back-end aspects of a web application. It encompasses a wide range of technologies and skills to build fully functional and dynamic websites and web applications. In the context of web development with Java and Python, here's a brief introduction to each component:

**FRONT-END DEVELOPMENT:**

Front-end development focuses on the user interface (UI) and user experience (UX) of a website or web application. It involves creating the visual elements that users interact with, such as web pages, forms, buttons, and navigation menus.HTML, CSS, and JavaScript are fundamental technologies for front-end development. HTML structures web content, CSS styles it, and JavaScript adds interactivity and dynamic behaviour.

Back-End Development:

Back-end development deals with the server-side logic and data processing of a web application. It manages data storage, security, authentication, and communication between the front-end and databases. Java and Python are both popular programming languages used for back-end development. Back-end frameworks like Spring (Java) and Django (Python) provide tools and libraries to streamline development tasks.

Full-Stack Development:

Full-stack developers are proficient in both front-end and back-end technologies, allowing them to work on the entire web application stack. They are responsible for building the complete architecture of a web application, ensuring that it functions smoothly and delivers a seamless user experience. Full-stack developers often work with databases, APIs, server management, and application deployment in addition to front-end and back-end coding. In essence, full-stack development combines front-end and back-end skills to create web applications that are interactive, secure, and capable of handling user requests and data effectively. This holistic approach is essential for developing robust and versatile web applications that can meet the demands of modern online platforms and services. It also enables developers to work on various aspects of a project, making them versatile and valuable assets in the field of web development.

**METHODOLOGY**

Python Methodology :

In Python web application development, Python is mostly used for building the backend. Python-frameworks take care of dealing with tasks that are related to backend application development, such as parsing HTTP requests, generating HTTP responses, accessing the database, authentication, authorization, etc.

Java script Methodology:

JavaScript is a scripting language used to develop web pages. Developed in Netscape, JS allows developers to create a dynamic and interactive web page to interact with visitors and execute complex actions. It also enables users to load content into a document without reloading the entire page.

**HTML METHODOLOGY:**

The marked up HTML document is said to be the structural layer of a web page. It is the foundation upon which the presentation layer (instructions for how the elements should be delivered or displayed) and the behavioural layer (scripting and interactivity) are applied.

CSS Methodology:

CSS is the language for describing the presentation of Web pages, including colours, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

**FRONT-END DEVELOPMENT**

Front-end development, also known as client-side development, is the process of creating the visual and interactive elements of a website or web application that users see and interact with directly in their web browsers. It involves three core technologies:

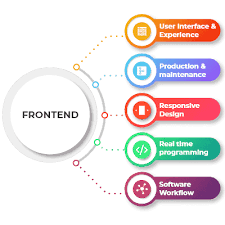
* HTML (Hyper Text Markup Language):

HTML is the backbone of a web page. It defines the structure and content of web documents.HTML uses a markup language composed of tags that describe elements like headings, paragraphs, links, images, and forms.

* CSS (Cascading Style Sheets):

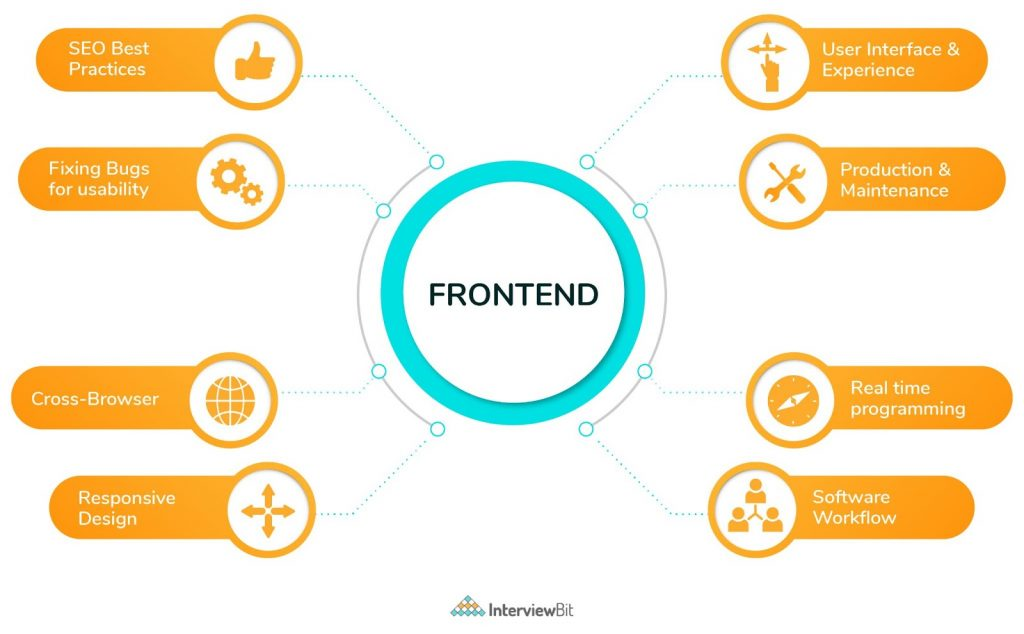
CSS is used to style and format the HTML elements, making them visually appealing. It defines properties such as colours, fonts, margins, padding, and layout positioning.

* JavaScript:

JavaScript adds interactivity and dynamic behaviour to web pages. It enables you to respond to user actions, manipulate the DOM (Document Object Model), and communicate with back-end services. ****

**WHAT DOES A FRONT-END DEVELOPER DO?**

A front-end developer is a professional who has a keen eye for programming and designing at the same time. It’s all about developing better things that users would see first in their web browsers. Although there are some variations across companies, you can generally expect a front-end developer to include some or all of the following responsibilities:



Front end web development is one of the processes in web development which converts data to graphical interface, for user to view and interact with data, through digital interaction using HTML, CSS and JavaScript. In simpler terms, front end development is how a web design actually gets implemented on the web.

A front-end developer is a programmer who writes codes to develop the front end of a website. In other words, front-end developer's convert website design files into raw HTML, JavaScript (JS) and/or CSS code which include the basic website design/layout, images, content, buttons, navigation and internal links. The end result is a code that serves as the website's front-end structure, which is used by a back-end developer, to add business logics and connect databases and processes.

A front-end developer is responsible for ensuring that a website's visual front end is free of errors and looks exactly as designed. A front-end developer also ensures that a website has the same visibility across different computing and mobile Web browsers.Similarly, in software applications, a front-end web developer creates the graphical user interface (GUI) that enables access to the software's back-end features and capabilities.In reinforcement learning settings, the learner has the ability to actively interact with the environment. The objective of the learner is to maximise the reward over its set of interactions with the environment . Due to its ability to consistently explore the environment to select the optimal strategy, researchers and practitioners have used reinforcement learning algorithms for risk-optimised dynamic portfolio allocation .

**ADVANTAGES OF FRONT-END**

Java scripts:

* Speed.
* Simplicity.
* Popularity.
* Interoperability.
* Server Load.
* Rich Interfaces.
* Extended Functionality.
* Versatility

HTML:

* simple to understand and implement.
* Completely free.
* Every browser supports HTML.
* The most search engine friendly.
* Simple to Edit.
* simple to integrate with other languages.

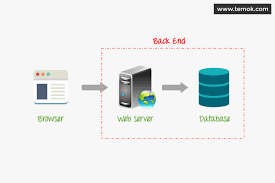
CSS:

* Faster Page Speed.
* More code means slower page speed.
* Better User Experience.
* CSS not only makes web pages easy on the eye, it also allows for user-friendly formatting.
* Quicker Development Time.
* Easy Formatting Changes.
* Compatibility Across Devices

**BACK-END DEVELOPMENT**

Back-end development, also known as server-side development, refers to the development of the behind-the-scenes logic that powers web applications**.**

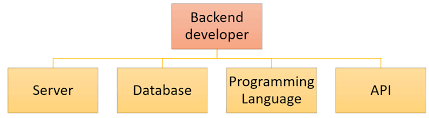
Back-end development is crucial for building the server-side functionality that powers webapplications, manages data, and provides services to the front end. It ensures that the application functions reliably, securely, and efficiently.

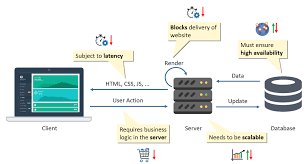


* Server Logic: Back-end developers work on the server-side of a web application. They handle data processing, business logic, and interactions with databases.
* Data Storage: Back-end development involves managing databases where application data is stored, retrieved, and manipulated. Common databases include MySQL, PostgreSQL, MongoDB, and others.
* APIs (Application Programming Interfaces): Back-end developers often create APIs to allow communication between the front end and back end. APIs enable data exchange and functionality access, typically using formats like JSON or XML.
* Security: Back-end developers are responsible for implementing security measures to protect user data, authenticate users, and ensure the overall security of the application.
* Server Management: They handle server deployment, maintenance, and scalability to ensure the application can handle increased traffic and remains available.
* Testing and Debugging: Rigorous testing and debugging are essential in backend development to identify and fix issues before they impact users. Automated testing and monitoring tools are often used.
* Documentation: Clear and comprehensive documentation is crucial for backend developers. It helps other team members understand the code and APIs, making collaboration more efficient.
* Collaboration: Backend developers work closely with front-end developers, database administrators, and other team members to ensure a cohesive and functional web application.
* Scaling: As the application grows, backend developers are responsible for scaling the infrastructure to handle increased traffic and user load. This may involve load balancing, caching, and optimizing code for performance.
* Server Infrastructure: Backend developers handle server deployment, configuration, and management. They ensure that the web application can handle traffic efficiently and remains available with minimal downtime.
* Server-Side Logic: Backend development focuses on the server-side logic and functionality of a web application. It involves writing code that runs on the server and handles various tasks such as data processing, business logic, and managing user requests.

**WHY DO WE NEED BACK-END DEVELOPMENT?**

Back-end development is a crucial part of websites since back ends are comprised of databases, authentication/authorization features, application programming interfaces (APIs) and all the different components that, combined with a front-end, allow users to view and interact with applications in their browser.

Semi-supervised learning problem

is

**WHY DO BACK-END DEVELOPERS DO?**

Back-end developers must focus on a great many tasks and responsibilities to ensure that the application runs smoothly. These can include any or all of the following:

* Writing clean, maintainable code:

This can include building prototypes, creating entire systems from scratch, adding new features to existing systems and more.

* Database design and management:

Almost every back-end stores data in a database. A back-end developer should have extensive knowledge of both relational and non-relational databases.

* Building and managing APIs:

Whether REST, SOAP, RPC or GraphQL, APIs are used extensively in back-end development as a means of communication with the client and other systems.

* Troubleshooting and debugging applications:

Every developer, regardless of whether they work on the front-end or the back-end, must know how to debug and solve any problems that may arise in the application they’re building. Solving problems is the day-to-day life of any developer.

* Maintaining the web server:

It isn’t unusual for back-end developers to dabble in configuring and maintaining the web server that hosts their code, especially in smaller businesses and startups that lack the budget to hire a systems administrator.

* Collaborating with front-end developers:

A back-end developer must have excellent communication skills in order to collaborate with front-end developers.

**KEY BACK-END DEVELOPMENT CONCEPTS**

Back-end developers have a great many tools and programming languages at their disposal. These tools include different types of databases, frameworks, web servers and more.

* Java
* Python
* PHP
* JavaScript (Node.js)
* Ruby
* C#

**FRAMEWORKS**

Frameworks are designed to make building and deploying web applications much easier. They provide a standard way to build applications while providing different libraries, templates, data-access solutions, authentication and authorization solutions, security measures and boilerplate code, among many other features. Some of the most popular frameworks include:

* Spring (Java)
* Django (Python)
* Laravel (PHP)
* Express (JavaScript)
* Ruby on Rails (Ruby)
* .NET (C#)

**WEB SERVERS**

Web servers are a combination of software and hardware that stores, processes and delivers web pages to users. Communication between users and web servers happens by way of the Hyper Text Transfer Protocol (HTTP). All of the back-end code is hosted on a web server. Some of the most popular web servers are:

* Apache
* Nginx
* Lite Speed
* Microsoft-IIS
* Open Resty

**DATABASES**

Databases are an essential part of any web application. They are not just in charge of storing the application’s data, but also providing a way for the developer to query that data in order to return it and display it to the user.

Databases are divided in two main categories: non-relational and relational. In a nutshell, relational databases organize the data in tables that are related to each other (hence the name relational), while non-relational databases don’t use tables and instead store data in a variety of formats.

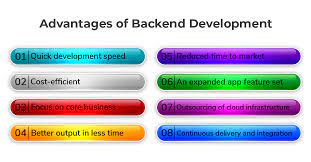
Some of the most popular databases include:

* MySQL
* Oracle
* Microsoft SQL Server
* PostgreSQL
* MongoDB
* IBM Db2
* Redis

**ADVANTAGES**

Backend development offers several advantages that contribute to the overall success and functionality of web applications. Here are some key advantages:

* Data Management: Backend development allows for efficient management of data. Backend systems can store, retrieve, and manipulate data from databases, ensuring data integrity and consistency.
* Security: Backend developers implement robust security measures to protect user data, authentication processes, and application resources. They can control access to sensitive information and safeguard against common web vulnerabilities.
* Scalability: Backend systems can be designed and scaled to handle increased traffic and user load. This scalability ensures that the application can grow and accommodate a growing user base without sacrificing performance.
* Business Logic: Backend development enables the implementation of complex business logic and rules. This is essential for applications that involve calculations, workflows, and decision-making processes.
* Performance Optimization: Backend developers can optimize code and database queries to improve application performance. This includes reducing load times and ensuring responsive user experiences.
* Cross-Platform Compatibility: Backend systems can provide data and services to various front-end interfaces, such as web browsers, mobile apps, and other platforms. This flexibility ensures a consistent user experience across different devices.
* Third-Party Integrations: Backend development allows for seamless integration with third-party services, APIs, and tools. This enables the application to leverage external functionalities, such as payment gateways, social media platforms, or cloud services.
* Robust APIs: Backend developers create APIs that define how data and services can be accessed by the front end or external applications. These APIs provide clear and structured communication between different parts of the system.
* Centralized Control: Backend development provides centralized control over the application's logic and data, making it easier to maintain and update the core functionality.
* Data Analysis and Reporting: Backend systems can collect and process large volumes of data, making it possible to generate insights through data analysis and reporting. This is valuable for making informed business decisions.
* Data Backup and Recovery: Backend developers implement data backup and recovery strategies to ensure data is not lost in case of unexpected events or system failures.
* Version Control: Backend code can be managed using version control systems, making it easier to collaborate with other developers, track changes, and roll back to previous versions if needed.



**CONCLUSION**

In conclusion, full-stack development that combines web development with Java and Python encompasses a comprehensive and versatile approach to building modern web applications. It bridges the gap between the front-end and back-end aspects of web development, utilizing a range of technologies and tools to create functional, secure, and user-friendly web applications.

Front-end development focuses on creating visually appealing and interactive user interfaces using HTML, CSS, and JavaScript. It ensures a positive user experience and engages users with the application.

Back-end development, on the other hand, handles server-side logic, data management, security, and scalability. It ensures data integrity, business logic execution, and efficient communication between the client and server. Java and Python are popular choices for back-end development due to their versatility and robust ecosystems.

Together, full-stack developers possess the skills to work on the entire web application stack, from designing user interfaces to handling server-side logic and data storage. This holistic approach enables them to create well-rounded, feature-rich web applications that meet the demands of today's digital landscape.

In summary, full-stack development is a valuable skillset that empowers developers to create end-to-end solutions, delivering seamless and engaging web experiences for users while also ensuring the robustness and functionality of the underlying infrastructure. It's an exciting and dynamic field that continues to evolve with emerging technologies and changing user expectations.

**REFERENCES**

* Books on Full-Stack Development:

1. "Full Stack Development with Java and React" by Piotr Gankiewicz and Yavor Georgiev: This book provides insights into building full-stack applications using Java on the back end and React on the front end.
2. "Full Stack Python" by Matt Makai: This online book covers the full-stack development process using Python, including front-end frameworks like Flask and Django.

* Online Resources and Documentation:

1. MDN Web Docs: Mozilla Developer Network provides extensive documentation on web technologies including HTML, CSS, and JavaScript. MDN Web Docs
2. Java Documentation: The official documentation for Java can be found on the Oracle website. Oracle Java Documentation
3. Python Documentation: The official Python documentation offers comprehensive resources for Python developers. Python Documentation

* Coding Platforms and Practice:

1. LeetCode: LeetCode offers coding challenges and contests that can help you practice your coding skills, including algorithms and data structures often used in full-stack development.
2. HackerRank: HackerRank provides coding challenges and competitions with a focus on algorithms and data structures. It's a great platform for practicing coding skills.

* Web Development Frameworks:
* Spring Framework (Java): The official website for the Spring Framework provides extensive documentation, guides, and tutorials for Java-based web development.
* Django Framework (Python): The official Django website offers documentation and resources for Python web development using the Django framework.