18ECP108L - INTERNSHIP

YEAR: 2023-24 (Year IV)

SEMESTER – VII (ODD)

COMPANY: LingoScriptAI

DOMAIN: Video Processing & Web Development

REPORT SUBMITTED BY:

Kunal Keshan

[RA2011004010051]

FACULTY IN-CHARGE

DR. ADITYA NATH BHATT



DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY S.R.M. Nagar,

Kattankulathur – 603203, Chengalpatt District

TABLE OF CONTENTS

S. No.	Title	Page No.
1.	Background of Organization	3
2.	Acknowledgement	4
3.	Description About Internship	5
4.	Abstract	6
5.	Working Experience	7
6.	Conclusion	25
7.	References	26
8.	Internship Certificate	27

BACKGROUND OF ORGANIZATION

LingoScriptAI, a pioneering startup founded in May 2023, represents an innovative force in the digital content creation domain. Rooted in the convergence of technology and creativity, the company is dedicated to reshaping the landscape of content creation through the deployment of cutting-edge tools and methodologies.

At its core, LingoScriptAI is guided by a vision to revolutionise the creative processes of content producers. The startup's mission centres around democratizing the art of content creation by providing accessible and advanced tools. By harnessing the power of machine learning, sophisticated video manipulation techniques, and web development, LingoScriptAI aims to optimize and streamline the content creation process.

LingoScriptAI's core focus lies in the development of state-of-the-art products engineered to meet the demands of modern content creation. Its commitment to incorporating machine learning algorithms forms the cornerstone of its offerings, intending to empower users to generate engaging, impactful, and short-form content with exceptional ease.

The startup actively integrates the latest advancements in video manipulation technology and web development. Through this integration, the company strives to provide an extensive suite of tools that cater to the dynamic requirements of content creators. Incorporating sophisticated video manipulation techniques aims to empower users to create, edit, and enhance content in previously unattainable ways.

LingoScriptAI is deeply committed to a user-centric design philosophy. Through in-depth market research and a thorough understanding of user requirements, the company endeavours to develop solutions that are not only aligned with user expectations but surpass them, thereby fostering enhanced usability and satisfaction.

The strategic initiative of LingoScriptAI extends beyond redefining content creation. The company envisions becoming a vital contributor to the broader technological domain. By cultivating a community of creators and innovators, it aims to transform content creation from a mere process into an immersive experience, ultimately facilitating the actualization of ideas with unprecedented ease and efficiency.

LingoScriptAI emerges as a vanguard in technological innovation, destined to reshape content creation paradigms by converging leading-edge technology, creative empowerment, and an unwavering commitment to user-centric solutions.

ACKNOWLEDGMENT

I am profoundly grateful for the invaluable opportunity granted to me by Govardhan, Sriram Selvan, and Karthik, who facilitated my enriching tenure as a product development engineer intern at LingoScriptAI. Their trust, mentorship, and encouragement empowered me to delve into the realms of multimedia processing using FFMPEG, expand my understanding of web development, and gain foundational insight into DevOps practices on the AWS platform.

Govardhan, with his sagacious leadership and astute guidance, provided the nurturing environment essential for learning and growth. His unwavering support and expertise in the field served as a cornerstone for my explorations into multimedia processing, fostering a deep appreciation for the intricacies of FFMPEG and its role in video processing.

Sriram Selvan, with his meticulous insights and technical acumen, offered me a canvas to comprehend the complexities of web development. His mentoring exposed me to the intricacies of creating dynamic web applications and laid the foundation for understanding the evolving landscape of web-based tools and technologies.

Karthik, with his adept understanding and guidance, introduced me to the essential principles of DevOps and cloud-based infrastructure on AWS. His mentorship not only expanded my knowledge but also provided a glimpse into the critical aspects of deploying applications, Continuous Integration/Continuous Deployment (CI/CD), and the fundamental principles of managing cloud resources.

The collective efforts and mentorship provided by Govardhan, Sriram Selvan, and Karthik have been pivotal in sculpting my professional journey during this internship. I extend my heartfelt gratitude to them for their patience, encouragement, and unwavering support in helping me navigate the multifaceted domains of multimedia processing, web development, and DevOps practices on AWS.

I would also like to express my sincere appreciation to the entire team at LingoScriptAI, whose collective efforts and collaborative spirit created an environment conducive to learning, innovation, and growth. The experiences and insights gained during my internship have laid a strong foundation for my future endeavours in technology and content creation.

DESCRIPTION OF INTERNSHIP

During my tenure as a product development engineer intern at LingoScriptAI, the experience was both enriching and transformative. Engaging in a spectrum of technological initiatives and multifaceted projects, I gained comprehensive insights and hands-on experience across diverse facets of multimedia processing, full-stack development, and DevOps methodologies.

• FFMPEG and Video Processing:

 Extensive exposure to FFMPEG facilitated a deep exploration of video processing parameters. This encompassed an in-depth study of various video codecs, bitrates, resolutions, and compression techniques. The hands-on application of FFMPEG offered a practical understanding of manipulating and optimizing multimedia content, serving as a pivotal learning experience.

• Full Stack Development with Next.js and Tailwind:

 Delving into full-stack development using Next.js and Tailwind CSS for frontend design was a pivotal aspect of the internship. Next.js, with its server-side rendering (SSR) capabilities, coupled with the utility-first approach of Tailwind CSS, provided an in-depth comprehension of crafting user-centric and responsive interfaces.

• Node.js and Express for Backend Development:

O The internship facilitated a comprehensive journey into backend development, leveraging Node.js and Express.js. This deep dive into backend architecture and RESTful API creation was instrumental in understanding server-side operations and data exchange protocols. This alignment with academic theories further enriched my grasp of fundamental backend development practices.

• DevOps Methodologies and AWS Deployment:

The internship offered an invaluable opportunity to explore DevOps practices in conjunction with AWS, EC2 instances, and Linux fundamentals. Engaging in infrastructure management, containerization, and continuous integration/deployment (CI/CD) methodologies on the AWS platform enhanced my proficiency in handling cloud-based resources and understanding the orchestration of scalable systems.

ABSTRACT

My internship tenure at LingoScriptAI commenced on June 18, 2023, extending through to July 31, 2023. During this period, I was actively engaged in a compelling project focused on building a web-based video editor. The project involved an immersive exploration and utilization of several key tools and technologies, shaping the foundation of my learning experience:

• Multifaceted Project Objective:

The core objective of the internship revolved around the development of a web-based video editor. Leveraging a repertoire of tools, including FFMPEG for video processing, Next.js and Tailwind CSS for frontend design, and Node.js/Express for backend functionalities, the project aimed at amalgamating diverse technologies to create an intuitive, versatile, and efficient video editing platform.

• Flexible Working Hours and Structured Deadlines:

The internship work schedule featured flexible hours, providing an
accommodating environment conducive to productivity and innovation. The
workdays spanned from Monday to Friday, enabling a collaborative and creative
workspace. The team structure encompassed daily evening stand-ups, fostering
effective communication and project progress tracking.

• Open Team Culture and Collaborative Work Dynamics:

 The team environment embraced an open dialogue culture, encouraging discussions and idea-sharing for optimizing the project. Regular meetings and stand-ups facilitated a free exchange of ideas, enabling a collaborative approach towards problem-solving and innovation.

My experience during the internship at LingoScriptAI encompassed a detailed journey into the multifaceted aspects of video editing tools and techniques, amalgamating academic principles with practical, hands-on experiences. The interplay between structured deadlines, flexible work hours, and a collaborative team culture fostered an environment conducive to learning, creativity, and professional growth.

WORKING EXPERIENCE

Concepts and tools related to the work involved:

Understanding Audio & Video Parameters

• Bitrate

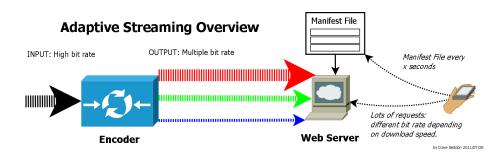
Bitrate serves as a fundamental metric in the domain of multimedia processing, determining the rate at which data is processed or transmitted in audio and video files. Particularly in the context of video, bitrate plays a pivotal role in quantifying the amount of data allocated to represent each frame. This allocation directly impacts both the quality and the resulting file size of the video content. Acquiring a comprehensive understanding of bitrates is crucial for professionals engaged in video processing, as it enables precise control over the video's quality and size, ultimately influencing the audience's viewing experience.

Example using FFmpeg:

To control the video bitrate with FFMPEG, you can use the -b:v option followed by the desired bitrate value. For example, to set the video bitrate to 2000 kbps:

ffmpeg -i input.mp4 -b:v 2000k output.mp4

Proficiency in bitrate control finds extensive application across various multimedia industries, including film production, streaming services, and online content creation. It contributes significantly to efficient data transmission, ensuring optimal video quality while accommodating the constraints of bandwidth and storage capacities. The integration of bitrate management in multimedia technologies forms a cornerstone in creating immersive and accessible multimedia experiences.



• Framerate

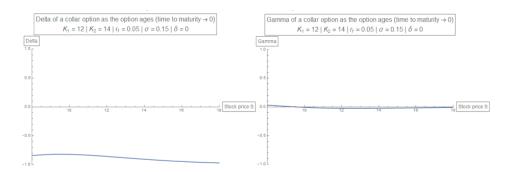
Framerate stands as a crucial parameter in multimedia, characterizing the number of frames or images displayed per second in a video. This metric significantly influences the visual experience by impacting the perceived smoothness of motion in videos. A higher framerate usually translates to a more fluid and lifelike video playback. Mastery of framerates equips professionals with the ability to exert precise control over video playback speed and animation fluidity, enhancing the overall visual appeal and viewing experience.

Example Using FFmpeg:

To change the framerate with FFMPEG, use the -r option. For instance, to convert a video to 30 frames per second:

ffmpeg -i input.mp4 -r 30 output.mp4

Proficiency in framerate control finds extensive application across various multimedia industries, encompassing film production, animation, gaming, and online streaming platforms. It caters to the demand for smoother motion, higher visual fidelity, and enhanced user experience. Additionally, in the realm of technological advancements, the manipulation of framerates aligns with the constant pursuit of higher visual quality and innovation in multimedia content delivery.



• Compression and Its Algorithms

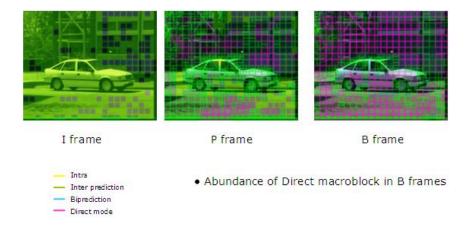
Compression serves as a fundamental process in multimedia technology, aimed at reducing the size of audio and video files while maintaining optimal quality. This pivotal procedure facilitates efficient storage and seamless transmission of multimedia content. A nuanced understanding of compression algorithms, such as H.264 or MP3, empowers professionals to optimize media content for diverse applications, ensuring an equilibrium between file size and quality.

Example Using FFmpeg:

FFMPEG supports various video and audio codecs. For compression, you can specify the codec to use. For example, to compress a video with the H.264 codec and AAC audio:

ffmpeg -i input.mp4 -c:v libx264 -c:a aac output.mp4

Compression algorithms find widespread application across various multimedia industries, including film production, streaming services, and online content creation. It plays a pivotal role in enabling high-quality multimedia experiences while optimizing storage and bandwidth usage. Moreover, within the context of technological advancements, the integration of compression algorithms aligns with the continuous quest for higher-quality multimedia content delivery and enhanced user experiences.



Transcoding

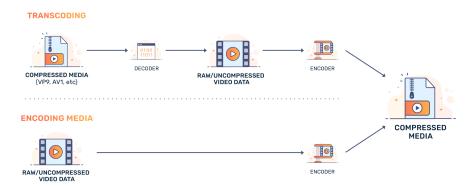
Transcoding serves as a pivotal process in multimedia processing, involving the conversion of a media file from one format to another. This transformation is often performed to ensure the compatibility of content across various devices and platforms. Proficiency in transcoding empowers professionals to seamlessly adapt media, addressing compatibility issues and ensuring content accessibility to a wider audience.

Example Using FFmpeg:

To transcode a video to a different format, specify the output format with the -f option. Here's an example of transcoding to MP4:

ffmpeg -i input.avi -f mp4 output.mp4

Transcoding finds extensive application across various multimedia industries, including film production, streaming services, and online content creation. It plays a significant role in ensuring content accessibility and compatibility across different devices and platforms. Furthermore, in the domain of technological advancements, transcoding aligns with the ever-evolving landscape of digital media, ensuring content reaches a diverse audience without compatibility constraints.



• Scaling and Animation

Scaling plays a vital role in multimedia by allowing adjustments to the size of images or video frames. It enables content adaptation to various screen sizes and resolutions, ensuring an optimal viewing experience across different devices. Additionally, animation contributes dynamic motion to video content, fostering user engagement and enhancing the visual appeal of multimedia projects.

Example Using FFmpeg:

To scale a video to a specific size, use the -vf option with the scale filter. For example, to scale a video to 1280x720 pixels:

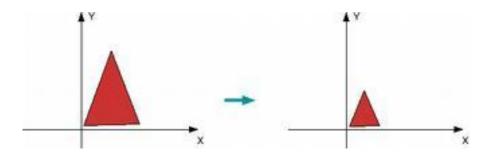
ffmpeg -i input.mp4 -vf "scale=1280:720" output.mp4

For adding animation or overlays, use the -vf option with filters like overlay:

ffmpeg -i input.mp4 -i overlay.png -filter complex "[0:v][1:v]overlay=10:10" output.mp4

Scaling and animation find applications across various multimedia industries, including film production, game development, and web design. It plays a pivotal role in designing responsive

content and delivering immersive user experiences. Furthermore, the integration of these techniques nurtures creative video manipulation and innovative multimedia projects.



Using Web-Based Technologies

React.js

• Understanding React.js

React.js, commonly referred to as React, is a powerful open-source JavaScript library primarily used for building user interfaces (UIs) and managing the view layer of web applications. Developed by Facebook, React was introduced to address the need for an efficient and flexible front-end development tool. It's designed around the concept of reusable and composable UI components, facilitating the creation of dynamic and interactive web applications.

Key Principles and Features of React.js

- Component-Based Architecture: React's fundamental principle revolves around breaking down the user interface into reusable components, allowing developers to build complex UIs by combining and nesting these components. This modular approach enhances code reusability, maintainability, and scalability.
- Virtual DOM (Document Object Model): React employs a virtual representation of the actual DOM, enabling efficient and optimized updates. The Virtual DOM helps in minimizing the re-rendering of the entire page and only updates the components that have changed, leading to improved performance.
- **JSX** (**JavaScript XML**): JSX is a syntax extension for JavaScript that allows HTML to be written within JavaScript code. This feature simplifies the creation of UI components and facilitates the integration of JavaScript and HTML seamlessly.

• Unidirectional Data Flow: React follows a unidirectional data flow, where data flows in a single direction, ensuring predictability and control over the application state. This pattern enhances application stability and facilitates easier debugging.

Benefits of Using React.js

- **Reusability and Componentization**: The component-based structure of React promotes code reusability, enabling developers to efficiently reuse components across different parts of the application.
- Efficiency and Performance: React's Virtual DOM and its selective rendering approach enhance the performance of web applications by reducing unnecessary re-rendering and improving rendering speed.
- **Declarative Syntax:** React's declarative nature simplifies the codebase and makes it more readable and predictable. Developers focus on what the interface should look like rather than manually managing the DOM.
- Community and Ecosystem: React boasts a vast community of developers and an extensive ecosystem of libraries and tools, offering comprehensive support and resources for development.

Code Example Using React.is

Here's an example illustrating a basic React component displaying a simple greeting message:

export default Greeting;

This example demonstrates the creation of a functional React component called Greeting, which accepts a 'name' property and displays a greeting message.

Absolutely! Here is an in-depth coverage of Next.js for your report:

Next.js

Understanding Next.js

Next.js is an open-source React framework, designed to streamline the development of server-side-rendered (SSR) and statically-generated web applications. Built on top of React, Next.js introduces a powerful and flexible approach to building modern web applications. It offers a comprehensive solution by integrating server-side rendering, routing, and other functionalities into a single framework.

Key Principles and Features of Next.js

- Server-Side Rendering (SSR): Next.js enables server-side rendering, enhancing performance and search engine optimization (SEO) by pre-rendering pages on the server before serving them to the client. SSR leads to faster initial page loads and improved accessibility for search engine crawlers.
- Static Site Generation (SSG): Next.js supports static site generation, allowing the creation of static pages at build time. This feature is beneficial for content-heavy websites, ensuring quick and efficient loading of content.
- **File-Based Routing**: Next.js simplifies routing by utilizing the file system. Each JavaScript file in the pages directory corresponds to a route, facilitating clean and intuitive routing without the need for additional configuration.
- **API Routes**: Next.js provides an integrated API routes feature for creating serverless functions. This simplifies the process of building API endpoints, enabling backend functionalities without a separate server.
- **Hybrid Approach**: Next.js allows developers to use both SSR and SSG selectively, providing the flexibility to choose the rendering method that best suits specific pages or sections of the application.

Benefits of Using Next.js

- **Optimized Performance**: Leveraging server-side rendering and static site generation, Next.js enhances performance by delivering optimized, pre-rendered content, resulting in faster load times.
- **SEO-Friendly**: Next.js helps improve SEO by serving fully-rendered HTML to web crawlers, enabling better indexing and search ranking for websites.
- **Simplified Routing**: With file-based routing, Next.js simplifies the organization and navigation of application routes, promoting clean and straightforward project structures.
- **Developer Experience**: Next.js provides a highly developer-friendly environment with built-in features and intuitive tools, reducing the complexities of setting up configurations.

Code Example Using Next.js

Here's an example of a simple Next.js component illustrating server-side rendering:

export default Home;

This code exemplifies a basic Next.js component with server-side rendering functionality, fetching data from an API and passing it to the page.

Tailwind CSS

• Understanding Tailwind CSS

Tailwind CSS is a modern utility-first CSS framework that provides low-level utility classes to build custom and responsive user interfaces efficiently. Unlike traditional CSS frameworks that utilize pre-built components, Tailwind CSS offers a set of utility classes that directly apply styling to HTML elements, allowing for more flexibility and customization in web design.

Key Principles and Features of Tailwind CSS

- **Utility-First Approach**: Tailwind CSS operates on a utility-first methodology, providing a wide range of utility classes that directly apply CSS styles. This approach empowers developers to create custom designs by composing utility classes within HTML elements.
- **Responsive Design**: Tailwind CSS promotes responsive design by offering utility classes that allow quick adaptation to various screen sizes. It facilitates the creation of mobile-first, responsive layouts without the need to write custom media queries.
- Customization and Extensibility: Developers can extend or customize Tailwind's default configuration by adding or modifying utility classes, colours, spacings, and other design elements as needed.
- **Performance Optimization**: Tailwind CSS optimizes performance by generating minimal and specific CSS, reducing the overall file size and rendering lightweight and efficient styles.

Integration with React.js and Practical Examples

Tailwind CSS integrates seamlessly with React.js, allowing for the creation of component-based UI design. By applying Tailwind classes directly to React components, developers can quickly style and design interfaces without the need for separate CSS files.

• Example of Using Tailwind CSS with React.js

In this example, Tailwind utility classes directly style the React component. The 'bg-gray-200', 'p-6', 'text-2xl', 'font-bold', 'text-indigo-700', 'text-lg', 'text-gray-600', and 'mt-4' classes apply background color, padding, text styles, and margins to the elements.

Node.js

• Understanding Node.js

Node.js is an open-source, cross-platform runtime environment that enables server-side execution of JavaScript code. It is built on the V8 JavaScript engine and facilitates the development of scalable and high-performance applications by using event-driven, non-blocking I/O models.

Key Principles and Features of Node.js

- **JavaScript Runtime**: Node.js enables the execution of JavaScript on the server-side, expanding its capabilities from client-side web development to backend programming, thereby allowing developers to use a single programming language across the stack.
- **Event-Driven Architecture**: Node.js utilizes an event-driven architecture that leverages asynchronous, non-blocking I/O operations. This approach allows handling multiple concurrent connections without thread blocking, resulting in enhanced performance and scalability.
- **NPM (Node Package Manager)**: Node.js comes bundled with NPM, a package manager that hosts a vast repository of open-source packages. NPM simplifies dependency management, enabling developers to integrate libraries, modules, and tools seamlessly into their applications.
- Scalability and Real-Time Applications: Node.js is well-suited for building scalable applications and real-time features like chat applications, streaming platforms, and collaborative tools due to its event-driven nature and ability to handle a large number of concurrent connections.

Practical Use and Integration with Web Development

Node.js is commonly integrated with web development frameworks and libraries like Express.js, allowing the development of server-side logic and APIs. It plays a crucial role in backend development, supporting data processing, server setup, and API creation for web applications.

• Example of Using Node.js with Express.js

```
const express = require('express');
const app = express();
app.get('/', (req, res) => {
  res.send('Welcome to Node.js and Express!');
});
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

In this example, a basic Express.js application using Node.js sets up a server that responds with a welcome message upon visiting the root URL.

Express.js

• Understanding Express.js

Express.js is a fast, minimalistic, and flexible Node.js web application framework, designed for building robust and feature-rich web applications. It provides a simple and organized way to create server-side logic, handle routes, and manage middleware, making it an ideal choice for developing APIs and web services.

Key Principles and Features of Express.js

- **Middleware**: Express.js operates on a middleware system that allows developers to define a series of functions or actions to handle HTTP requests. It enables the execution of code at various request-handling stages, such as logging, authentication, and error handling.
- **Routing**: Express.js simplifies routing by allowing the creation of routes for different HTTP methods like GET, POST, PUT, DELETE, etc. It helps in handling various endpoints and defining corresponding functionalities.
- **Template Engines Integration**: Express js provides support for various template engines such as EJS, Pug, and Handlebars, allowing the rendering of dynamic HTML content, which is beneficial for building user interfaces.
- Error Handling: It offers mechanisms for centralized error handling, ensuring smooth execution and handling of errors that might occur during the application's lifecycle.

Practical Use and Integration with Node.js

Express.js is commonly integrated with Node.js for server-side development. It simplifies server setup and API creation by providing a set of features to handle requests, define routes, and manage responses.

• Example of Using Express.js with Node.js

```
const express = require('express');
const app = express();

app.get('/', (req, res) => {
  res.send('Welcome to Express.js!');
});

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

In this example, a basic Express.js application is set up using Node.js. It responds with a welcome message upon visiting the root URL.

TypeScript

• Understanding TypeScript

TypeScript is an open-source programming language developed by Microsoft that extends JavaScript by adding static typing. It aims to enhance JavaScript productivity by enabling the use of static types, providing improved tooling, error checking, and better code maintainability.

Key Principles and Features of TypeScript

- **Static Typing**: TypeScript introduces static typing, allowing developers to define types for variables, function parameters, and return types. This feature enables catching errors during development and facilitates better code structuring.
- **Enhanced IDE Support**: The use of static types in TypeScript enhances IDE support, enabling features like code navigation, refactoring tools, auto-completion, and type inference, resulting in more productive development workflows.

- **Compatibility with JavaScript**: TypeScript is a superset of JavaScript, meaning that JavaScript code can be easily integrated into TypeScript files. This compatibility allows for a smooth transition to TypeScript in existing projects.
- Improved Code Readability and Maintainability: The explicit type definitions in TypeScript contribute to better code understanding, making it more readable and maintainable. It also enhances collaboration among developers in larger projects.

Practical Use and Integration with JavaScript Projects

TypeScript is frequently integrated with JavaScript projects, especially in large-scale applications, to bring the benefits of static typing and better tooling.

• Example of Using TypeScript

```
interface Person {
  name: string;
  age: number;
}

function greet(person: Person): string {
  return `Hello, ${person.name}!`;
}

const user: Person = { name: 'John', age: 30 };

console.log(greet(user));
```

In this example, a simple TypeScript code snippet defines an interface 'Person' with specific properties, ensuring type safety for the 'greet' function and 'user' object.

Working with DevOps Principles

Introduction to DevOps

• Understanding DevOps

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and deliver high-quality software applications more rapidly, frequently, and reliably by aligning and automating processes between development and IT teams.

Key Principles and Features of DevOps

- Collaboration and Communication: DevOps emphasizes seamless collaboration and communication between development, operations, and quality assurance teams. It promotes breaking down silos and fostering a culture of shared responsibility and accountability.
- Continuous Integration (CI) and Continuous Deployment (CD): CI involves the integration of code changes into a shared repository frequently, ensuring early detection of defects. CD involves the automatic deployment of code changes to testing, staging, and production environments, ensuring rapid and reliable application delivery.
- **Automation**: Automation is a fundamental aspect of DevOps. It involves using tools and scripts to automate processes, such as testing, deployment, configuration management, and infrastructure provisioning, leading to increased efficiency and reduced human error.
- Monitoring and Feedback: DevOps emphasizes continuous monitoring of applications and infrastructure, providing feedback on performance, user experience, and system health. This feedback loop helps in making data-driven decisions and improvements.

Practical Use and Integration with Software Development

DevOps practices and tools are integral in the software development lifecycle. Continuous integration, deployment, and automation tools like Jenkins, GitLab, Docker, Kubernetes, and configuration management tools like Ansible or Chef are commonly used to implement DevOps practices.

• Example of Using DevOps Practices:

Example CI/CD pipeline in a .gitlab-ci.yml file stages:

- build
- test
- deploy

```
build:
    stage: build
    script:
        - npm install
        - npm run build

test:
    stage: test
    script:
        - npm test

deploy:
    stage: deploy
    script:
        - ssh user@your-server 'bash -s' < deploy.sh</pre>
```

This GitLab CI/CD pipeline example demonstrates stages for building, testing, and deployment. It triggers the pipeline upon pushing changes to a repository, executes defined tasks, and deploys code changes to a server.

Amazon EC2 Instances

• Understanding AWS EC2 Instances

Amazon EC2 (Elastic Compute Cloud) is a web service provided by Amazon Web Services that offers resizable compute capacity in the cloud. It allows users to create and manage virtual machines for various computing needs, enabling scalable and flexible solutions for diverse workloads.

Key Principles and Features of AWS EC2 Instances

• **Virtual Computing Environments**: EC2 instances provide users with virtual computing environments known as instances. Users can select the instance type, such as general-purpose, memory-optimized, or compute-optimized, based on specific workload requirements.

- Scalability and Elasticity: EC2 offers scalability, allowing users to scale up or down based on computing demands. It provides elasticity, enabling the addition or removal of instances as needed, contributing to cost-efficiency and flexibility.
- **Variety of Instance Types**: EC2 offers a wide range of instance types with varying CPU, memory, storage, and networking capabilities, catering to specific use cases like web applications, databases, machine learning, and more.
- Security and Networking: Users have control over the security configuration of EC2
 instances, including firewall settings, security groups, and encryption. It also offers
 networking features such as VPC (Virtual Private Cloud) for private and isolated cloud
 resources.

Practical Use and Integration with Cloud Services

EC2 instances are utilized in various scenarios such as web hosting, application hosting, data processing, and machine learning. They are often integrated with other AWS services like Amazon S3 for storage, Amazon RDS for databases, and AWS Lambda for serverless computing.

• Example of Using AWS EC2 Instances

\$ aws ec2 run-instances --image-id ami-12345678 --count 1 --instance-type t2.micro --key-name MyKeyPair

This AWS CLI command creates an EC2 instance using a specific Amazon Machine Image (AMI), defines the instance type as t2.micro, and specifies the key pair used for authentication.

AWS Certificate Manager (ACM)

The AWS Certificate Manager (ACM) is the service responsible for managing SSL/TLS certificates on the AWS platform. It simplifies the process of deploying, managing, and renewing SSL/TLS certificates for use with AWS services.

<u>Understanding AWS Certificate Manager (ACM)</u>

- Managed SSL/TLS Certificates: ACM automates the management of SSL/TLS certificates. It enables users to request, deploy, and renew public and private SSL/TLS certificates for use with AWS services or integrated third-party services.
- **Secure Communication**: ACM ensures secure communication between web applications and clients by providing encryption, securing data in transit, and establishing trust through certificate validation.
- Integration with AWS Services: ACM integrates seamlessly with various AWS services, such as Elastic Load Balancing, CloudFront, and API Gateway, to provide secure connections, reducing the operational overhead associated with manual certificate management.
- Certificate Types: ACM offers various types of certificates, including domain-validated (DV), organization-validated (OV), and extended validation (EV) certificates, providing different levels of validation and trust for different use cases.
- **Benefits for Users**: By utilizing ACM, users can automate the complex and error-prone process of obtaining and managing SSL/TLS certificates, ensuring that their services are secure and trusted by end-users.
- Example of ACM Usage

Request a public SSL/TLS certificate in the AWS Management Console aws acm request-certificate --domain-name example.com --validation-method DNS

In this example, a command is initiated to request a public SSL/TLS certificate for the domain "example.com" using DNS validation through the AWS CLI.

CONCLUSION

• Completion of Internship as a Product Development Engineer:

• The one-month internship journey at LingoScriptAI, in the role of a product development engineer, has been a transformative and insightful experience.

• In-depth Learning in Video Processing, Web Development, and DevOps:

• The internship tenure facilitated a comprehensive exploration and immersion into the realms of video processing, web development, and DevOps methodologies.

• Exploration of Tools like FFMPEG, Next.js & AWS:

• The internship provided a comprehensive exploration of tools such as FFMPEG for video processing and manipulation, and Next.js for creating sophisticated and responsive web applications.

• Practical Application and Hands-On Experience:

 The hands-on application of FFMPEG and Next.js, along with other tools, further deepened my understanding and proficiency in multimedia processing and web development.

The internship at LingoScriptAI has not only broadened my technical skill set but has also fostered a deeper comprehension of the intricate interplay between video processing, web development, and the principles of DevOps.

REFERENCES

- https://ffmpeg.org/
- https://react.dev/
- https://nextjs.org/
- https://tailwindcss.com/docs/
- https://docs.aws.amazon.com/
- https://www.geeksforgeeks.org/introduction-to-devops/

CERTIFICATE

Phone: +91 9025817537 Email: hello@lingoscriptai.com

LingoScriptAl

CERTIFICATE OF INTERNSHIP

This is to certify that Mr. Kunal Keshan (RA2011004010051), B.tech (ECE) student at SRM Institute of Technology Kattankalathur Chennai, has successfully completed the internship as a Product Engineer Intern at LingoScriptAI. The internship commenced on 18-06-2023 and concluded on 31-07-2023.

During the internship period, **Mr. Kunal Keshan** actively participated in various projects and assignments, contributing valuable insights and demonstrating exceptional dedication and enthusiasm. They displayed a strong aptitude for problem-solving, critical thinking, and technical skills, making a significant impact on our team and projects.

Based on their outstanding performance and commitment, we believe **Mr. Kunal Keshan** possesses great potential and a promising future in the field of product engineering.

We extend our sincere appreciation for their valuable contributions to our organization and wish them the best for their future endeavors.

With Best Regards, GOVARDHAN.S,

Co-Founder, LingoScriptAi.

Email - govardhan@lingoscriptai.com

Date - 02-08-2023