Android App Development

Internship Report

Submitted for the partial fulfilment of the degree of

Bachelor of Technology

In

Engineering Mathematics & Computing

Submitted By

Shrikant Suryawanshi

0901MC201061

UNDER THE SUPERVISION AND GUIDANCE OF

Dr. Divya Chaturvedi

Assistant Professor

Department of Engineering Mathematics and Computing

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA

Deemed to be University

NAAC ACCREDITED WITH A++ GRADE

January-June 2024

DECLARATION BY THE CANDIDATE

I hereby declare that the work entitled **Android App Development** is my work, conducted under the supervision of **Dr. Divya Chaturvedi, Assistant Professor,** during the session Jan-May 2024. The report submitted by me is a record of bonafide work carried out by me.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Shrikant Suryawanshi
0901MC201061
B.Tech. VIII Sem

Date: 16/05/24
Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided By:

Dr. Divya Chaturvedi
Assistant Professor
Department of Engineering
Mathematics & Computing
MITS, Gwalior

Departmental Project Coordinator

Dr. D K Jain
Professor
Department of Engineering
Mathematics & Computing
MITS, Gwalior

Approved by HoD

Dr. Vikas P. Shinde
Professor
Department of Engineering
Mathematics & Computing
MITS, Gwalior

PLAGIARISM CHECK CERTIFICATE

This is to certify that I/we, a student of B.Tech. in **Engineering Mathematics & Computing** have checked my complete report entitled **Android App Development** for similarity/plagiarism using the "Turnitin" software available in the institute.

This is to certify that the similarity in my report is found to be 10% which is within the specified limit (20%).

The full plagiarism report along with the summary is enclosed.

Shrikant Suryawanshi

0901MC201061

Checked & Approved By:

Dr. J K Muthele
Associate Professor
Department of Engineering
Mathematics and Computing
MITS, Gwalior

EXECUTIVE SUMMARY

This internship report gives a comprehensive diagram of the Android app advancement extend embraced in Kotlin. Over the course of the internship, the centre was on leveraging Kotlin's capabilities to form a robust and user-friendly mobile application.

The report starts with an presentation to the project's destinations and scope, sketching out the require for the advancement of the Android app and its expecting functionalities. It at that point dives into the strategy utilized, enumerating the instruments, advances, and frameworks utilized all through the development process.

Key angles of the app advancement lifecycle, including requirement analysis, plan, execution, testing, and sending, are examined in profundity. Uncommon consideration is given to the utilize of Kotlin's highlights to improve code coherence, viability, and performance.

The internship given priceless down to earth involvement in Kotlin app advancement, permitting for the application of hypothetical information in a real-world setting. The bits of knowledge gained from this internship will serve as a strong foundation for future tries in Android app development.

In conclusion, the Android app improvement internship was a fulfilling and improving encounter, advertising valuable insights into the complexities of mobile application improvement. The coming about app stands as a confirmation to the viability of Kotlin as a programming dialect for building imaginative and feature-rich Android applications.

ACKNOWLEDGEMENT

I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit**, and the Dean Academics, **Dr. Manjaree Pandit** for this project. It is my great pleasure to express sincere gratitude to my mentor and supervisor, **Dr. Divya Chaturvedi**, Assistant Professor, Department of Engineering Mathematics & Computing for his expert guidance and constant encouragement. I acknowledge that it is because of his interest that I enjoyed working on this project and express my earnest and heartfelt thanks to him for his time, support, and efforts.

I am also thankful to all the faculties of the **Department of Engineering Mathematics & Computing** for their encouragement, who had invested their valuable time in providing their feedback with a lot of useful suggestions.

I am highly obliged to all my friends for their encouragement and for helping me at the points where I got stuck. I am deeply indebted to all of them for always helping and inspiriting me.

Shrikant Suryawanshi

0901MC201061

CERTIFICATE OF INTERNSHIP

AD Infocom Systems

62, Bajrang Nagar, Manewada Road, Nagpur – 440027
Contact No.: 9860455757 **Email:** adinfocomsystems@gmail.com
Website: www.adinfocomsystems.in

Approved By
All India Council For Technical Edu
AICTE(NEW DELHI)

Ref. No. - ADIS/2024/25

Date - 01/06/2024

Internship Certificate

We are happy to announce that **Shrikant Suryawanshi** has successfully completed the internship at AD Infocom Systems in the domain of **"App Development"** from **01/01/2024** to **01/06/2024**.

During internship, you have successfully completed all the modules and project work.

We wish you all the best for your future endeavors.

Thank you.

Prabhakar Dorge
Managing Director
AD Infocom Systems

AD Infocom Systems
Seal
Nagpur

CONTENT

Table of Contents

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

AD Infocom Systems stands as a pioneering software-based company, with its headquarters nestled in Nagpur, Maharashtra, India. Renowned for its proficiency in a myriad of IT services, AD Infocom Systems excels in web development, app development, software development, and intricate coding tasks such as MATLAB and NS2.

Beyond crafting cutting-edge solutions, the company is also committed to nurturing talent through comprehensive training and internship programs. These initiatives span across diverse domains of Engineering and Technology, empowering aspiring professionals to hone their skills and contribute meaningfully to the industry.

AD Infocom Systems embodies innovation and expertise, consistently delivering bespoke solutions tailored to meet the evolving needs of its clientele. With a fervent dedication to excellence and a penchant for pushing boundaries, the company continues to carve a formidable presence in the ever-expanding landscape of information technology.

Fig1.1: Company Logo

CHAPTER 2: ABOUT ANDROID

2.1 Introduction

Android, developed by Google, stands as a ubiquitous operating system powering billions of smartphones and tablets worldwide since its inception in 2008. Built on the robust foundation of the Linux kernel, Android is tailored for touchscreen devices, offering users an intuitive and customizable experience. Its flexibility is evident in the myriad of options available for personalization, from widgets to themes, allowing users to tailor their devices to suit their preferences. At the heart of Android's appeal lies its expansive ecosystem of applications, accessible through the Google Play Store, offering users access to millions of apps catering to various needs and interests.

2.2 Why Android

Android's broad selection can be ascribed to a few variables that make it a favored choice for both clients and designers alike. Firstly, its colossal client base guarantees that engineers have get to a expansive group of onlookers, making it an alluring stage for app advancement. In addition, Android's open-source nature cultivates a collaborative environment, where designers can contribute to its advancement and advancement. Also, Google's comprehensive biological system gives engineers with a plenty of instruments, assets, and conveyance channels, streamlining the app improvement prepare and encouraging app revelation.

Fig2.1: Android Logo

The diversity of Android devices, spanning various price ranges and specifications, ensures that there is something for everyone, catering to different budgets and demographics. Overall, Android's combination of accessibility, flexibility, and support makes it an indispensable platform for developers seeking to create impactful and engaging mobile experiences.

2.3 Common Languages in Android

When it comes to programming languages, developers can choose from a variety of solutions for Android development, each with unique advantages and applications. Because of its reliability, speed, and extensive library and framework ecosystem, Java has long been the most popular language for creating Android apps. Nonetheless, Kotlin's attributes, such its short syntax and null safety, have contributed to its appeal in more recent years. Because Kotlin and Java are fully compatible, developers can easily transfer old projects or use both languages within the same codebase.

Furthermore, C and C++ are still useful for jobs like game development and system component construction that call for high performance or low-level system access. Lastly, XML is used to create Android layouts for user interface design, offering an organized method of specifying the arrangement and visual style of app screens. With so many languages available, developers may select the one that best fits their needs and style, guaranteeing flexibility and productivity when creating Android apps.

2.4 Kotlin vs Java

Selecting between Kotlin and Java for Android app development has become a hotly debated topic. Both languages offer advantages and disadvantages, and developers wishing to start their Android development journey must be aware of these distinctions.

Java has been the mainstay of Android application development for almost ten years, having been the most popular language for Android development during that time. Java, which is renowned for its durability and reliability, offers a strong platform for creating dependable and effective apps. Because of its well-developed ecosystem and extensive documentation,

Developers especially those with prior Java programming experience—find it to be a popular option.

However, since its 2011 introduction by JetBrains, Kotlin has been increasingly popular among Android developers. With features and syntax aimed to improving code clarity and expressiveness, Kotlin presents a more contemporary take on programming. Because of its compatibility with Java, developers may easily include Kotlin code into already-existing Java projects, which makes it simple for developers who are already familiar with Java to use.

Null safety features, which lessen the possibility of null pointer exceptions, a frequent cause of errors in Java code, are one of Kotlin's main advantages over Java. In order to ensure that null values are handled properly at compile time rather than resulting in runtime problems, Kotlin's compiler statically verifies for nullability.

In summary, Kotlin's contemporary features and improved developer experience make it an appealing option for Android app development, even though Java and Kotlin both offer advantages. For experienced Java developers seeking to update their knowledge or for novices wishing to explore Android development, Kotlin provides an adaptable and strong language that is ideal for creating the next wave of Android applications.

Fig2.2: Java

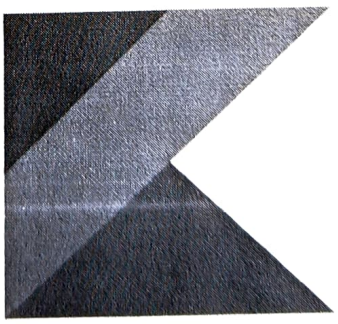


Fig2.3: Kotlin

2.5 Future of Android

Android appears to have a bright future ahead of it, with more developments and innovations planned. With Android powering billions of devices globally, it is anticipated that the platform will continue to dominate the mobile market as technology advances. The incorporation of machine learning and artificial intelligence (AI) into Android smartphones is a significant area of development. Our gadgets will become more efficient and intuitive as a result of these technologies, which will also allow for smarter and more tailored user experiences. Furthermore, because its operating system powers a variety of smart devices inside the Internet of Things (IoT) ecosystem, Android is well-positioned to transcend the realms of smartphones and tablets.

Project Treble and Project Mainline are two examples of Google's continuing development initiatives that are meant to improve device security and optimize the Android upgrade process while guaranteeing customers have access to the newest features and updates. Furthermore, the introduction of 5G technology would completely transform the mobile environment by providing reduced latency and quicker internet rates. As a result, Android device capabilities will be significantly enhanced and new opportunities for mobile applications and services will arise. All things considered, Android has a bright future full of limitless possibilities for development and innovation.

CHAPTER 3: APP DEVELOPMENT TYPES

3.1 Native and Hybrid App Development:

Native App Development: Developing apps specifically for a specific platform, like iOS or Android, is known as native app development. Platform-specific programming languages, such as Java or Kotlin for Android and Swift or Objective-C for iOS, are used to create these apps. Native applications are renowned for their peak performance, smooth device feature integration, and compliance with platform-specific UI/UX standards.

Hybrid App Development: In Hybrid app development, web technologies like HTML, CSS, and JavaScript are used to create applications, which are then wrapped in native containers. These applications are more affordable and have faster development cycles since they use a single codebase to run on several platforms. Simpler applications or projects with restricted deadlines and budgets are good candidates for hybrid applications.

Hybrid App Development can be done using many frameworks and technologies but mostly we use React Native and Flutter.

3.2 Developing Android Apps using React Native, Flutter, or Native:

React Native: A well-liked framework for creating cross-platform mobile applications with JavaScript and React is called React Native, and it was created by Meta. With only one codebase, it lets developers create native-like experiences for both iOS and Android devices. React Native offers access to native modules and APIs, facilitating faster development and easy code sharing between platforms. Its component-based architecture streamlines UI development and supports hot reloading, enhancing developer productivity. React Native is favoured for its extensive community support, robust ecosystem, and flexibility in integrating with existing native codebases.

Flutter: Using a single codebase, developers can create natively built apps for desktop, web, and mobile platforms with Google's Flutter UI toolkit. Developed using the Dart programming language, Flutter offers an extensive collection of modifiable widgets and a reactive architecture, facilitating the development of very efficient and aesthetically pleasing applications. Flutter's "write once, run anywhere" approach eliminates the need for platform-specific UI components, resulting in consistent experiences across devices. Its fast rendering engine, stateful hot reload, and rich developer tools contribute to rapid iteration and efficient development. Flutter's growing popularity stems from its modern architecture, expressive UI, and strong community support.

Native Android Development: Native Android application development involves building apps specifically for the Android platform using Java or Kotlin programming languages and the Android Studio IDE. Developers have full access to Android APIs and platform-specific features, allowing them to create high-performance, native user experiences tailored to Android devices. Native Android development offers seamless integration with device hardware, superior performance optimization, and adherence to Android design principles. It does, however, necessitate independent development work for other platforms, like as iOS, which might lengthen and increase the cost of development.

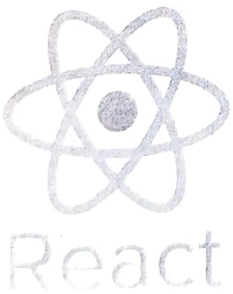


Fig3.1: React Native

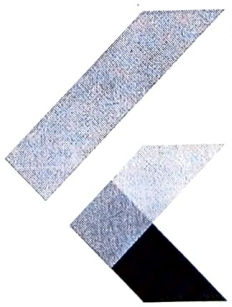


Fig3.2: Flutter

Fig3.3: Android

CHAPTER 4: INTEGRATED DEVELOPMENT ENVIRONMENT

4.1 What's an IDE?

A complete software suite called an Integrated Development Environment (IDE) is made to make the process of developing applications and programs easier. It has a single interface that houses several necessary tools, including a code editor, compiler, debugger, and class browser. This integration makes all required tools easily available, which improves the development process and increases productivity. Intelligent code completion, syntax highlighting, code refactoring tools, integrated version control, breakpoints for debugging, and build automation are typical elements of an IDE. An IDE's main objective is to remove the need for a developer to constantly switch between windows or applications by combining all of the tools they require into one single application..

There are specialist IDEs for a number of programming languages, such as Java, Python, and C++. Eclipse, NetBeans, IntelliJ IDEA, Visual Studio, and CodeBlocks are a few of the well-known IDEs. By offering a consistent and effective environment, these IDEs greatly simplify and accelerate the development process by providing features and tools that are specifically designed for the languages and frameworks they support.

4.2 How to choose an IDE?

An integrated development environment (IDE) selection should take a number of things into account. First and foremost, the language or framework you want to utilize should be your top priority. Make sure the IDE has strong support for the language of your choice. Although many IDEs are multilingual compatible, some are language-specific. Second, assess what features you really need. Think about if you require features like build automation, version control integration, debugging tools, or code completion. Choose an IDE that offers the tools you need for your development requirements. While some IDEs are more user-friendly for novices, others include more sophisticated functionality. Select the one that best suits your needs and skill level.

User interface and usability are equally important considerations. Try out many IDEs to see which one you are most at ease with. Think about things like third-party plugins, themes, shortcuts, and add-ons. A positive user experience may greatly increase your productivity and efficiency.

4.3 Key Features and Components of IDEs

For an android app development agency, integrated development environments (IDEs) provide a plethora of useful features and resources. The main features and elements present in the majority of IDEs are as follows:

a. Compiler: Generates executable code from source code so that it may be created and run right away in the integrated development environment.

b. Debugger: Helps find and fix coding mistakes. Breakpoints, step over/into actions, and changeable inspection are among the features it offers.

c. Intellisense: This feature saves time throughout the coding process by offering intelligent code completion by foreseeing potential code possibilities depending on context.

d. Version Control: By integrating with version control systems like Git and SVN, it makes managing various code versions easier.

e. Testing Tools: This section covers a variety of code testing tools, including unit and integration tests.

4.4 Popular IDEs for Android Development

4.4.1 Visual Studio Code (VS Code)

Microsoft created Visual Studio Code (VS Code), a robust, portable, and highly configurable code editor. Via its vast extension marketplace, Visual Studio Code (VS..Code) supports a multitude of programming languages, including JavaScript, TypeScript, Python, Java, C++, PHP, Go, and many more. It is available for Windows, macOS, and Linux. VS Code's

IntelliSense, which offers intelligent code completion based on function definitions, imported modules, and variable types, is one of its best features. It also comes with a powerful debugging tool, an integrated terminal, and Git version control support. Developers that value a simplified and effective coding environment frequently choose Visual Studio Code (VS Code) for online and mobile app development due to its flexibility and extensibility.

Fig4.1: Visual Studio (VS) Code

4.4.2 Android Studio

Google's official Integrated Development Environment (IDE) for creating Android apps is called Android Studio. It may be accessed on Linux, macOS, and Windows. A whole range of tools designed exclusively for Android development are available with Android Studio, such as an advanced code editor with code completion, refactoring, and code analysis. Some of its most notable features include the Gradle build system for task automation and dependency management, an integrated Android emulator for testing apps on virtual devices, and a robust Layout Editor with drag-and-drop capabilities for building user interfaces. Android..Studio is a vital tool for Android developers as it makes the process of creating, testing, and releasing Android apps easier with the help of these tools.

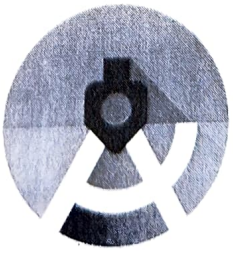


Fig4.2: Android Studio

4.4.3 Xcode

Apple's official IDE for macOS and iOS app development, Xcode is only compatible with macOS. Programming languages including Objective-C, C, C++, and Swift are supported. A wide range of tools are available in Xcode throughout the whole development process, such as the Interface Builder, which allows you to design and create user interfaces using a drag-and-drop interface, as well as extensive testing and debugging capabilities. Additionally, it has an interactive environment called Swift..Playgrounds where users may explore and learn how to write Swift code. Developing apps for Apple's platforms, such as iOS, macOS, watchOS, and tvOS, requires Xcode's interaction with the development environment, which includes comprehensive documentation and sample projects.

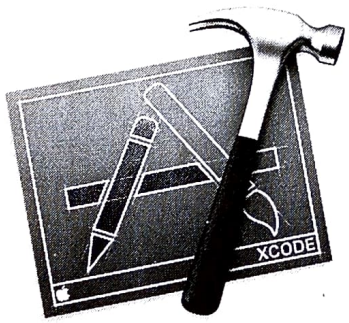


Fig4.3: Xcode

4.4.4 IntelliJ IDEA

Developed by JetBrains, IntelliJ//IDEA is a flexible integrated development environment (IDE) that runs on Windows, macOS, and Linux. It is mostly used for Java development, although it also supports additional languages including Kotlin, Groovy, and Scala. Developers can produce high-quality code more quickly and effectively with the aid of IntelliJ IDEA's strong refactoring tools, on-the-fly code analysis, and enhanced code completion. It has strong built-in tools for testing, debugging, and deploying applications, and it supports a number of

frameworks, including Spring, Java EE, GWT, and Vaadin. Developing large-scale Java apps and Kotlin-based Android projects is a favorite use case for IntelliJ IDEA due to its rich toolkit and intelligent coding support.

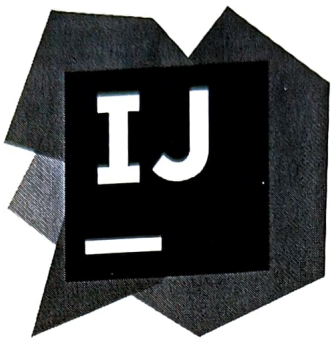


Fig4.4: IntelliJ Idea

4.4.5 Eclipse

Popular open-source IDE Eclipse is accessible on Linux, macOS, and Windows. Thanks to its extensive plugin ecosystem, it supports a broad variety of programming languages, including Java, C++, Python, PHP, and JavaScript. Because of Eclipse's strong extensibility, developers may add different tools and plugins to tailor their development environment. Integrated development and testing tools, support for version control and collaboration, and a modular

architecture that allows for customization to meet various development requirements are some of its key characteristics. Eclipse is especially well-liked among Java developers, although it can be used with a variety of platforms and programming languages because to its adaptability and extensibility.

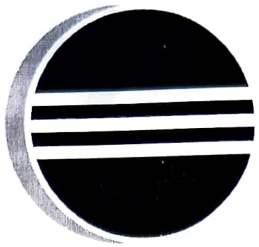


Fig4.5: Eclipse

CHAPTER 5: INTERNSHIP PROJECTS

5.1 Simple Calculator App

5.1.1 Introduction

Let's talk about a handy tool we all use every day: the Simple Calculator app. It's built using a cool language called Kotlin, and it's all about making math easy and quick.

This app keeps things simple and easy to understand. You can add, subtract, multiply, or divide numbers without any fuss. Plus, it's designed to be super easy to use, even if you're not a math whiz.

In this part, we'll take a closer look at how this app works, what makes it special, and how Kotlin helps make it all happen. Whether you're just adding up some numbers or need to do more complex calculations, this Simple Calculator app has got your back. So, let's dive in and explore the magic behind this handy little tool.

5.1.2 Requirements

To develop this app, we needed a few key things:

a. Understanding of Kotlin: Since we chose Kotlin as our programming language, we needed to be familiar with its basics and how to write code in it.

b. User Interface Design: We needed to create an easy-to-use interface for the calculator so that users could input numbers and operators effortlessly.

c. Mathematical Logic: Behind the scenes, the app needed to perform basic arithmetic operations like addition, subtraction, multiplication, and division accurately.

5.1.3 Explanation:

We started by designing a simple user interface (UI) with buttons for numbers (0-9) and operators (+, -, *, /). This UI was created using Kotlin's XML layout files, which define how the app looks on the screen.

Next, we wrote Kotlin code to handle user interactions and perform calculations. For example, when a user taps a number button, the corresponding number is displayed on the screen. When an operator button is tapped, the app remembers the current number and the selected operator. Finally, when the user taps the "=" button, the app calculates the result based on the stored numbers and operator.

Throughout the development process, we tested the app extensively to ensure it worked smoothly and accurately. We fixed any bugs or issues that arose and made improvements to enhance the user experience.

5.1.4 What We Learned:

Building the Simple Calculator app taught us several valuable lessons:

a. **Kotlin Fundamentals:** We deepened our understanding of Kotlin's syntax, features, and best practices while writing code for the app.

b. **UI/UX Design:** We learned how to create a user-friendly interface that makes it easy for people to interact with the app and perform calculations effortlessly.

c. **Problem-Solving:** Developing the app challenged us to think critically and creatively to solve problems, such as handling user input and performing calculations accurately.

d. **Testing and Debugging:** We gained experience in testing the app thoroughly and debugging issues to ensure it functions correctly across different devices and scenarios.

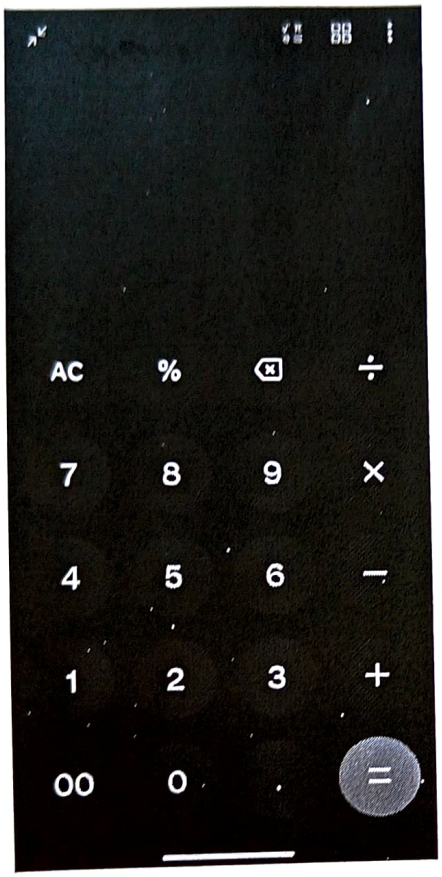


Fig5.1: Calculator

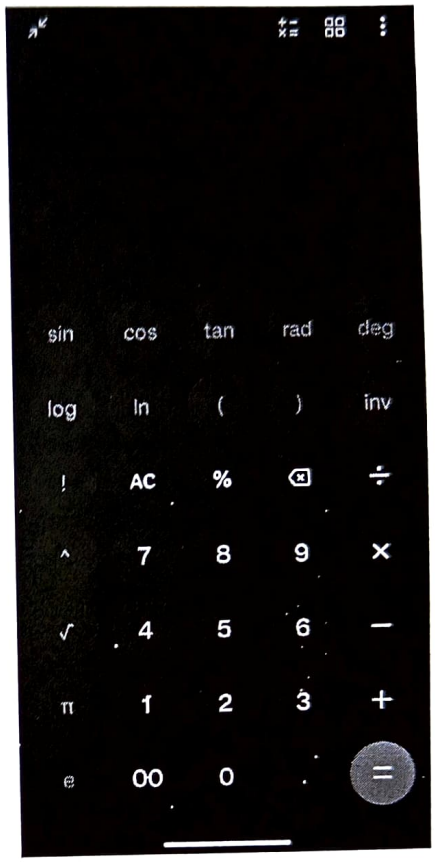


Fig 5.2: Calculator

5.2 Meme Sharing App

5.2.1 Introduction

In the age of social media, sharing memes has become a ubiquitous form of communication and entertainment. Enter the Meme Sharing app, a fun and user-friendly platform designed to delight users with a plethora of humorous and relatable content. Built with Kotlin, this app offers a seamless experience for discovering, creating, and sharing memes with friends and followers.

5.2.2 Requirements

To develop the Meme Sharing app, we needed to address several key requirements:

a. **Content Aggregation:** The app should aggregate a diverse range of memes from various sources, including popular meme websites, social media platforms, and user submissions.

b. **User Authentication:** Users should be able to create accounts, log in securely, and personalize their meme browsing experience.

c. **Meme Creation Tools:** The app should provide users with intuitive tools for creating and customizing memes, such as adding captions, emojis, and filters to images.

d. **Social Sharing Features:** Users should have the ability to share memes seamlessly with their friends and followers on social media platforms like Facebook, Instagram, and Twitter.

5.2.3 Explanation

We began by designing the user interface (UI) of the app, focusing on creating a visually appealing and easy-to-navigate layout. The UI includes features like a meme feed, search functionality, user profiles, and sharing options.

Next, we implemented functionality for content aggregation, leveraging APIs and web scraping techniques to fetch memes from popular sources. We also incorporated user-generated content, allowing users to submit their own memes for community viewing.

For user authentication and account management, we integrated secure login and registration processes using Kotlin's authentication libraries and encrypted storage techniques to safeguard user data.

In terms of meme creation tools, we developed a user-friendly interface that enables users to select images, add text overlays, apply filters, and customize memes to their liking. We also implemented features for meme editing, such as cropping, resizing, and rotating images.

To facilitate social sharing, we integrated APIs for seamless sharing of memes on popular social media platforms. Users can easily share memes with friends and followers with just a few taps, fostering virality and engagement within the app's community.

Throughout the development process, we prioritized user feedback and conducted usability testing to ensure that the app met the needs and expectations of its target audience. We addressed any bugs or performance issues promptly and iterated on the app's design and functionality based on user input.

5.2.4 What We Learned

Building the Meme Sharing app provided valuable insights and learnings in several areas:

1. **Content Aggregation:** We gained experience in sourcing and aggregating content from multiple sources, including APIs and web scraping techniques.
2. **User Authentication and Security:** We deepened our understanding of user authentication mechanisms and implemented secure practices to protect user data and privacy.
3. **UI/UX Design:** We honed our skills in designing intuitive user interfaces that prioritize ease of navigation and visual appeal, enhancing the overall user experience.

4 **Community Engagement:** We learned the importance of fostering community engagement and virality through social sharing features and user-generated content, driving user retention and growth.

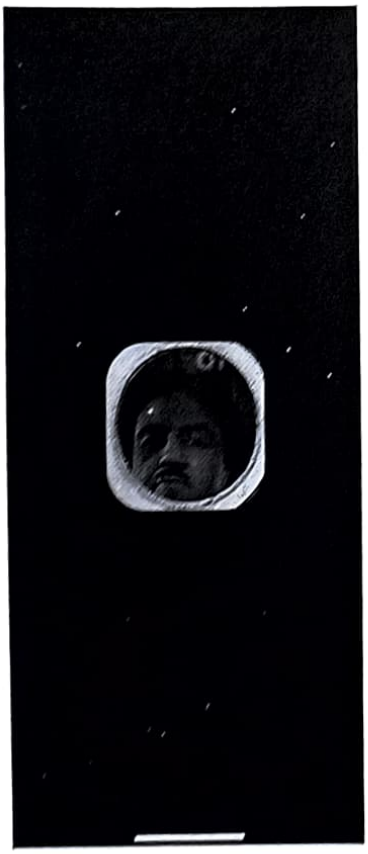


Fig5.3: Meme App(Splash Screen)

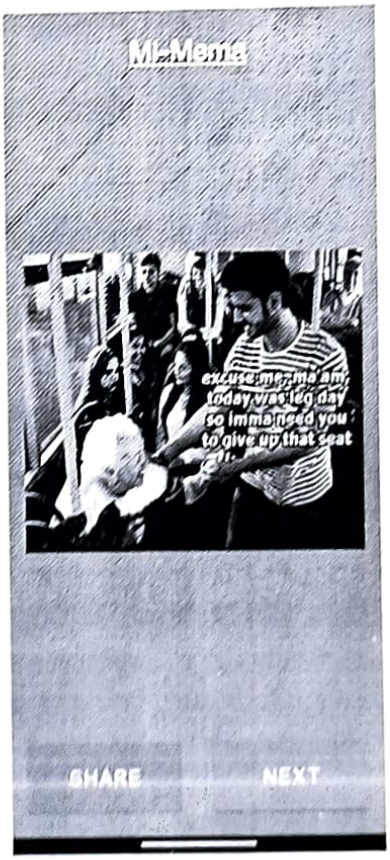


Fig5.4: Meme App(Home Page)

4. **Community Engagement:** We learned the importance of fostering community engagement and virality through social sharing features and user-generated content, driving user retention and growth.

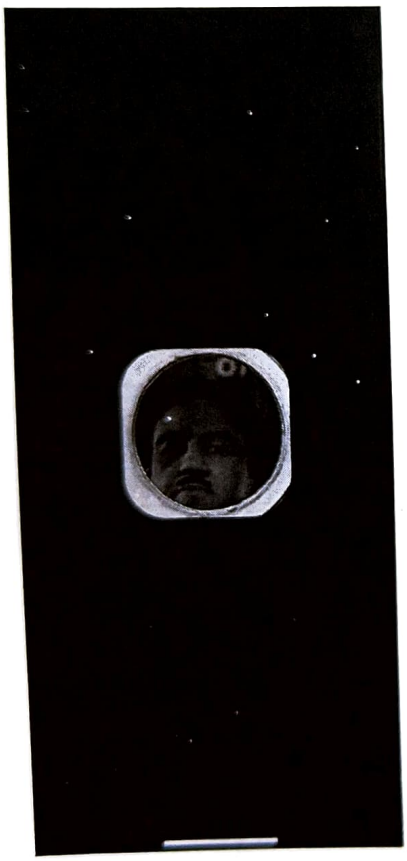


Fig5.3: Meme App(Splash Screen)

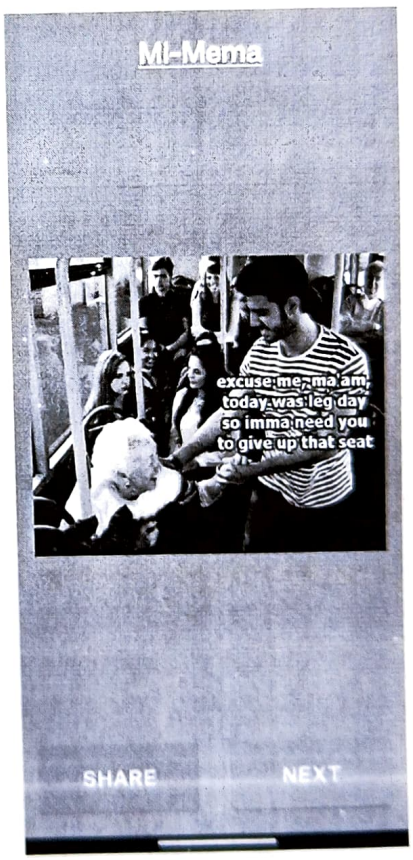


Fig5.4: Meme App(Home Page)

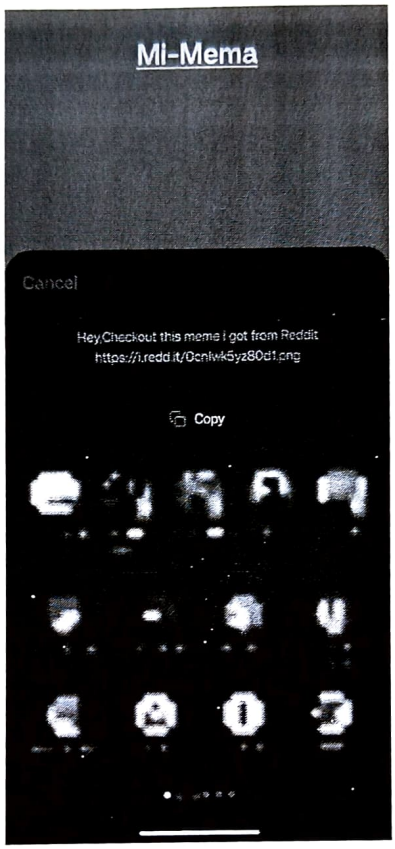


Fig5.5: Meme App (Share Page)

5.3 News App

5.3.1 Introduction

In today's fast-paced world, staying informed is more important than ever. Enter the News App, a comprehensive platform built with Kotlin, designed to keep users up-to-date with the latest news and events from around the globe. With a sleek and intuitive interface, this app provides a seamless experience for accessing diverse content, personalized to individual interests and preferences.

5.3.2 Requirements

Developing the News App involved addressing a range of critical requirements:

1. **Content Aggregation:** The app needed to aggregate news articles from a variety of reputable sources, including newspapers, magazines, online publications, and news agencies.
2. **Categorization and Filtering:** Users should be able to browse news articles based on categories such as politics, technology, sports, entertainment, and more. Additionally, the app should offer filtering options to refine search results based on keywords, topics, or specific sources.
3. **Personalization:** To enhance user engagement, the app should allow users to personalize their news feed by selecting favorite topics, sources, or regions. Personalized recommendations based on browsing history and user preferences should also be provided.
4. **Offline Access and Saving:** Users should have the ability to save articles for offline reading and access previously viewed content even when offline. This feature ensures uninterrupted access to news content, especially in areas with limited internet connectivity.
5. **Accessibility and Multilingual Support:** The app should prioritize accessibility features such as text-to-speech capabilities and support multiple languages to cater to a diverse user base.

5.3.3 Explanation

The development process of the News App began with conceptualizing and designing the app's user interface (UI). We aimed to create a visually appealing and user-friendly interface that would facilitate easy navigation and content discovery.

Next, we implemented functionality for content aggregation by integrating APIs from various news sources. These APIs enabled the app to fetch news articles in real-time and display them to users in a structured and organized manner.

For categorization and filtering, we developed algorithms to classify news articles into relevant categories and provide filtering options based on user preferences. Advanced search capabilities were implemented to enable users to find specific articles quickly and efficiently.

Personalization features were a key focus of the app, allowing users to customize their news feed according to their interests. Machine learning algorithms were employed to analyze user behavior and provide personalized recommendations tailored to individual preferences.

To address offline access and saving, we implemented local caching mechanisms to store news articles for offline viewing. This ensured that users could access their saved articles even without an internet connection, enhancing the app's usability and reliability.

Accessibility features such as text-to-speech conversion and support for multiple languages were integrated to make the app more inclusive and accessible to a wider audience.

Throughout the development process, we conducted thorough testing to ensure the app's functionality, performance, and reliability across different devices and network conditions. User feedback was also collected and incorporated to iteratively improve the app's features and user experience.

5.3.4 What We Learned:

Developing the News App provided valuable insights and learnings in several areas:

1. **Content Aggregation:** We gained experience in integrating and managing data from multiple sources, ensuring the reliability and accuracy of news content displayed in the app.

2. **Offline Access and Caching:** We developed expertise in implementing local caching mechanisms to enable offline access to news articles, improving the app's usability and reliability in diverse scenarios.

3. **Accessibility and Inclusivity:** We recognized the importance of incorporating accessibility features and multilingual support to ensure that the app is inclusive and accessible to users from diverse backgrounds and with varying needs.

Fig5.6: News App(Splash Screen)

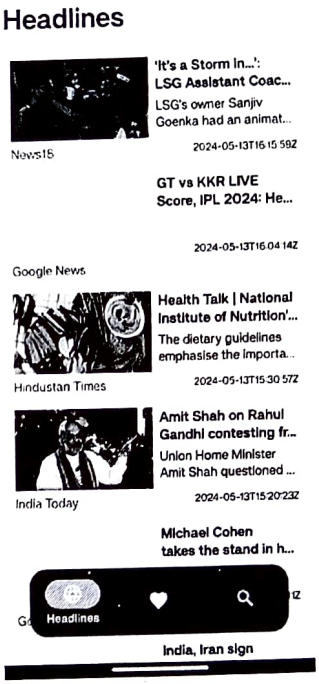


Fig5.7: News App(Home Page)

Fig5.8: News App (Favourites Page)

Fig5.9: News App (Search Page)

TURNITIN PLAGIARISM REPORT

Please Insert a Scanned Copy of the Front pages duly signed by the Candidate, Supervisor, Departmental Turnitrin Coordinator, and HoD with Seal

Similarity Report

PAPER NAME

Internship_Report.pdf

WORD COUNT	CHARACTER COUNT
5146 Words	**29319 Characters**
PAGE COUNT	FILE SIZE
32 Pages	**1.0MB**
SUBMISSION DATE	REPORT DATE
May 15, 2024 3:44 PM GMT+5:30	**May 15, 2024 3:51 PM GMT+5:30**

● **10% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- 1% Publications database
- Crossref database
- Crossref Posted Content database
- 9% Submitted Works database

● **Excluded from Similarity Report**

- Bibliographic material

ANNEXURE-1

LEARNING OUTCOMES

Learning outcomes in Android app development focus on the skills and knowledge that individuals can gain from learning this technology. In Android app development, I aim to understand fundamental concepts such as user interface design, application lifecycle, and data storage. They also learn to use various tools and frameworks to build functional and visually appealing applications. Moreover, students develop problem-solving skills as they troubleshoot errors and debug their code. Through Android app development, learners can enhance their creativity by designing innovative solutions to real-world problems.

Starting from Android basics like what and why Android is important to Why Android is a leader in current market. Concepts like Coroutines, Flows helped me in achieving Asynchronous programming thus improving performance of the apps. Learnt a lot about UI building using XML, layouts like RelativeLayout, GridLayout, StaggeredLayout, ConstraintLayout helped me in building apps more user friendly, but after learning Jetpack Compose, things become easier in terms of UI building. While designing any app, Architecture patterns like MVVM is widely used by most of the Developers, so using this, helped me in managing code and in deploying it to GitHub or any other Version Control System.

Transitioning to Kotlin, an increasingly popular programming language for Android development, expands these learning outcomes. With Kotlin, learners gain proficiency in writing concise and readable code, improving their productivity and efficiency. Additionally, Kotlin's features such as null safety and extension functions enable developers to write more robust and maintainable code. By mastering Kotlin for Android development, learners enhance their employability and open doors to exciting career opportunities in the ever-growing field of mobile app development.

ANNEXURE-2

DAILY DIARY

Week	Duration Start date – End date (DD/MM/YY) - (DD/MM/YY)	Progress of Internship
Week - 1	01/01/24 – 07/01/24	Initial Introduction to Android
Week - 2	08/01/24 – 14/01/24	Basics of Kotlin with Android Studio setup
Week - 3	15/01/24 – 21/01/24	Basics of OOPS and working of Gradle
Week - 4	22/01/24 - 28/01/24	Version Control System (BitBucket and Github)
Week - 5	29/01/24 – 04/02/24	First Project (Simple Calculator app)
Week - 6	05/02/24 – 11/02/24	App Components (Activity, Intent, etc) and Layouts
Week - 7	12/02/24 – 18/02/24	UI Designing using Compose and XML
Week - 8	19/02/24 – 25/02/24	Architecture Patterns (MVVM, MVP, MVC, etc)
Week - 9	26/02/24 – 03/03/24	Second Project (Meme Sharing App)
Week - 10	04/03/24 – 10/03/24	Network Call using Retrofit, OkHttp
Week - 11	11/03/24 – 17/03/24	Local and Cloud Database Introduction
Week - 12	18/03/24 – 24/03/24	Firebase and Room Database working with small tasks
Week - 13	25/03/24 – 31/03/24	Asynchronous Programming using Coroutines,Flows
Week - 14	01/04/24 – 07/04/24	Third Project (News App)
Week - 15	08/04/24 – 14/04/24	Linting using Ktlint and Project Completion
Week - 16	15/04/24 – 21/04/24	Debugging using various tools
Week - 17	22/04/24 – 28/04/24	Testing app using Espresso and JUnit
Week - 18	29/04/24 – 05/05/24	Common Practices while Deploying an app, Jetpack Compose optimization practices
Week - 19	06/05/24 – 12/05/24	App Deployment in Google Play Store
Week - 20	13/05/24 – 20/05/24	Internship Ending sessions with Doubt Clearance

ANNEXURE-3A

MPR-1

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV Bhopal)
NAAC Accredited with A++ Grade

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY/COMPANY MENTOR

Name of Student	Shrikant Suryawanshi		Enrollment No.	0901MC201061	
Department	Mathematics & Computing		Sem.	8(session: Jan-June 24)	
Industry/Organization	AD Infocom Systems		Date/Duration	01/01/24 – 01/06/24	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work			✓		
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work			✓		
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Shrikant completes all the tasks on time, being a fresher he has quite good knowledge of Android and currently he has completed all 8 tasks/ projects that we gave him. Hope he will keep working like this in the future.				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry/Company Mentor	**Mr. Prabhakar Dorge**				
Signature of Industry/Company Mentor					

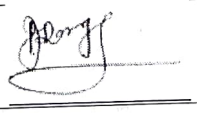
Receiving Date	**15/02/24**	**Name of Faculty Mentor**	**Dr. Divya Chaturvedi**	**Sign**	

ANNEXURE-3B

MPR-2

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV Bhopal)
NAAC Accredited with A++ Grade

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY/COMPANY MENTOR

Name of Student	Shrikant Suryawanshi		Enrollment No.	0901MC201061	
Department	Mathematics & Computing		Sem.	8(session: Jan-June 24)	
Industry/Organization	AD Infocom Systems		Date/Duration	01/01/24 – 01/06/24	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work			✓		
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	This month's report of Shrikant is quite impressive. The growth is quite significant as well.				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry/Company Mentor	**Mr. Prabhakar Dorge**				
Signature of Industry/Company Mentor					

Receiving Date	**15/03/24**	**Name of Faculty Mentor**	**Dr. Divya Chaturvedi**	**Sign**	

ANNEXURE-3C

MPR-3

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV Bhopal)
NAAC Accredited with A++ Grade

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY/COMPANY MENTOR

Name of Student	Shrikant Suryawanshi		Enrollment No.	0901MC201061	
Department	Mathematics & Computing		Sem.	8(session: Jan-June 24)	
Industry/Organization	AD Infocom Systems		Date/Duration	01/01/24 – 01/06/24	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work			✓		
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work				✓	
Behaviour/Discipline/Team work					✓
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	Shrikant's dedication and proficiency are truly commendable. As a fresher, his adaptness with Android development is impressive, evident from the successful completion of all work assigned to him. His commitment to meeting deadlines reflects not only his skills but also his reliability as a team member.				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry/Company Mentor	**Mr. Prabhakar Dorge**				
Signature of Industry/Company Mentor					

Receiving Date	**15/04/24**	**Name of Faculty Mentor**	**Dr. Divya Chaturvedi**	**Sign**	Divya

OFFER LETTER (AD INFOCOM SYSTEMS)

AD Infocom Systems

02, Bajrang Nagar, Manewada Road, Nagpur – 440027
Contact No.: 9860455757 Email: adinfocomsystems@gmail.com Website: www.adinfocomsystems.in

Ref. No. - ADIS/2023/319 **Date - 30/12/2023**

Internship Offer Letter

To,
Shrikant Suryawanshi
Madhav Institute of Technology and Science

Subject – Internship

Dear Shrikant Suryawanshi,

In reference to your registration we would like to congratulate you on being selected for internship in AD Infocom Systems. Your internship duration will be from 01/01/2024 to 01/06/2024. The topic of your internship is **"App Development"**.

You will undergo a learning curriculum as per the learning track assigned to you. The learning path will include in-depth sessions, hands on exercise and project work. The outcome during internship would be monitored through formal evaluations.

Again, congratulations and we look forward to working with you.

Thank you.

Prabhakar Dorge
Managing Director
AD Infocom Systems

AD Infocom Systems Seal Nagpur

OFFER LETTER (12TH WONDER)

12th Wonder Research India Pvt. Ltd.
Unit 312, Tower 1, World Trade Center,
Kharadi, Pune 411014, India

12TH WONDER
Work Smart

Date: 30-04-2024
Name: Shrikant Suryawanshi
Mobile: +91 9713482230

SUBJECT: INTERNSHIP APPOINTMENT LETTER

Dear Shrikant,
We are pleased to appoint your internship, in our Company, 12th Wonder Research India Pvt Ltd, effective on the following terms:

1. Internship Offer

Designation: Intern
Internship Start Date: 21-05-2024
Internship End Date: 21 Nov, 2024
Job Location: Pune, Maharashtra- India

2. Confirmation & Probation

On Successful completion of this Internship you shall be eligible for fulltime employment with 12th wonder subject to organisation's requirement at the time.
You shall be on a probation period of Three (3) months after conversion as a Full-time employee. This period may be further extended at the discretion of the Company based on your performance during probation period.

3. Remuneration

During the term of your Internship with the Company, your stipend will be **INR 12000** per Month. Post successful conversation as fulltime employee your remuneration will be revised to INR **300000-400000** per annum.

4. Confidentiality / Non-Disclosure Agreement.

a) You agree not to disclose any confidential information learned in the course of your internship about the business of the Company or its clients to anybody outside the firm both during and after your term of internship

b) You will be required to maintain utmost confidentiality in respect of project documents, commercial offers, design documents, project cost & estimation, technologies, software packages license, Company policies, Company patterns & trademarks and Company's human assets profile.