A REPORT ON THE INTERSHIP TRAINING AT

(Raha Industrial and Healthcare products)

Submitted By **NIVETHA R (212302880)**

2023-2026

Dr.MGR JANAKI COLLEGE OF ARTS AND SCIENCE FOR WOMEN

DEPARTMENT OF COMPUTER APPLICATIONS

INTERNSHIP REPORT FOR 2023-2024

NAME :	NIVETHA R
REGISTER NO :	212302880
DEPARTMENT&YEAR :	BCA & 1ST year
BATCH :	2023 to 2026
SHIFT :	I
INTERNSHIP :	WEB DEVELOPEMENT
COMPANY :	RAHA INDUSTRIAL AND HEALTHCARE PRODUCTS

ACKNOWLEDGEMENT

I wish to thank that Dr.MGR JANAKI college of arts and science for women having given me an opportunity to under do training in the company hereby wish to thank our correspondent Dr.LATHA RAJENDRAN and principal Dr.R.MANIMEGALAI of Dr.MGR JANAKI college of arts and science for women for giving me opportunity taken up institutional training at way to web.

Last but at least I own a deep sense of gratitude to my parents and my friends for their support and co-operation.

ABSTRACT

Web development is the process of creating and maintaining websites and web applications. It involves various technologies, programming languages, and frameworks to design and develop user-friendly and interactive websites including front-end development, back-end development, and database management. It also highlights the importance of responsive design, accessibility, and security in web development. Additionally, the abstract explores emerging trends such as progressive web applications (PWAs) and the integration of artificial intelligence (AI) and machine learning (ML) technologies in web development.

Mobile application development refers to the process of creating software applications specifically designed for mobile devices. The fundamental concepts and methodologies involved in mobile app development, including platform selection, user interface design, programming languages, and development frameworks. It also discusses the importance of performance optimization, cross-platform development, and mobile app testing for delivering high-quality mobile applications.

Software testing is a crucial phase in the software development lifecycle that ensures the quality, reliability, and functionality of software applications. It provides an overview of the key concepts and methodologies involved in software testing. It covers different types of testing, including unit testing, integration testing, system testing, and user acceptance testing (UAT). The abstract discusses the importance of test planning, test case development, and test automation in achieving effective software testing. It also highlights the significance of quality assurance (QA) processes, bug tracking, and continuous integration (CI) in software testing. Furthermore, the abstract explores emerging trends in software testing, such as agile testing, DevOps, and the adoption of artificial intelligence (AI) for test automation and analytics.

INTRODUCTION

A web application is a type of software application that runs on web servers and is accessed through a web browser or a mobile app using the Internet. These applications are designed to interact with users, process and store data, and provide various functionalities, such as e-commerce, social networking, online banking, or email. Web applications are typically built using web development technologies such as HTML, CSS, JavaScript, and server-side scripting languages such as PHP, Python, or Ruby. They can be accessed from any device with an internet connection, making them accessible and convenient for users across the globe.

Designing concepts

Designing is the fundamental principles and ideas that guide the creation of visual designs for various mediums such as graphic design, web design, user interface (UI) design, and product design. These concepts serve as a foundation for developing visually appealing, functional, and effective designs. Here are some key designing concepts:

Visual Hierarchy

Visual hierarchy involves arranging elements in a design to create a clear order of importance. It helps guide the viewer's attention and emphasizes key elements or messages. Designers use techniques like size, color, contrast, and placement to establish a visual hierarchy.

Balance

Balance refers to the distribution of visual weight in a design. It ensures that the elements are arranged harmoniously and creates a sense of stability and equilibrium. Balance can be achieved through symmetrical or asymmetrical arrangements of elements.

Typography

Typography deals with the selection, arrangement, and styling of fonts in a design. It includes considerations like font choice, size, spacing, and readability. Typography plays a crucial role in conveying the tone, hierarchy, and overall aesthetic of a design.

Colour Theory

Colour theory involves understanding how colours interact, evoke emotions, and communicate messages. Designers use color palettes, color combinations, and colour psychology to create visually appealing and meaningful designs. Colors can convey different moods, establish brand identity, and guide user interactions.

Contrast

Contrast refers to the difference in visual properties, such as colour, size, or shape, between elements in a design. Contrast helps create visual interest, highlight important elements, and improve readability. It enhances the overall impact and legibility of a design.

Unity and Consistency

Unity ensures that all elements in a design work together cohesively and create a unified whole. Consistency involves maintaining a common visual style and design elements throughout a project or across different screens or pages. Unity and consistency enhance the user experience and reinforce the brand identity.

White Space/Negative Space:

White space, also known as negative space, is the empty or unused space in a design. It helps improve readability, clarity, and visual focus. White space provides breathing room for elements and allows them to stand out.

User-Centered Design

User-centered design focuses on designing products, interfaces, and experiences that prioritize the needs, goals, and behaviors of the end users. It involves conducting user research, usability testing, and iterative design processes to create user-friendly and intuitive designs.

These designing concepts provide a framework for designers to make informed decisions and create visually appealing, functional, and user-centric designs. However, it's important to note that design is a creative field, and these concepts can be interpreted and applied in various ways based on the specific design goals and context**.**

Front End

Front-end refers to the part of a software application that the user interacts with directly. It includes the graphical user interface (GUI), the layout, the design, and the overall user experience.

In the context of web development, the front end typically consists of the HTML, CSS, and JavaScript code that is executed by a web browser. HTML (Hypertext Markup Language) is used to structure the content of the web page, while CSS (Cascading Style Sheets) is used to style the page and define its layout. JavaScript is used to add interactivity to the page, such as animations, dynamic effects, and user input validation.

Front-end developers are responsible for creating the visual and interactive aspects of web applications. They work closely with designers and back-end developers to ensure that the user interface is user-friendly, responsive, and visually appealing.

Backend

Backend development refers to the process of building and maintaining the server-side of a web application or software. It involves implementing the logic, algorithms, and database operations that enable the application to function properly. The backend is responsible for handling data processing, business logic, and integration with external services.

Key aspects of backend development include:

Server-side Programming: Backend developers use programming languages such as Python, Java, Ruby, PHP, or Node.js to write server-side code. They develop the application's core functionality, handle user requests, and process data.

Database Management: Backend developers work with databases to store, retrieve, and manipulate data. They use database management systems (DBMS) like MySQL, PostgreSQL, MongoDB, or Oracle to design and implement the database structure and optimize data access.

APIs and Web Services: Backend developers create APIs (Application Programming Interfaces) that allow different systems to communicate with each other. They define endpoints and data formats to enable data exchange between the frontend and backend or between different backend systems.

Security: Backend developers implement security measures to protect the application and its data. This includes authentication and authorization mechanisms, input validation, data encryption, and protection against common security vulnerabilities like SQL injection and cross-site scripting (XSS).

Performance Optimization: Backend developers optimize the application's performance by implementing caching mechanisms, efficient database queries, and load balancing techniques. They ensure the application can handle a large number of simultaneous users and respond quickly to user requests.

Web Designing

Web designing refers to the process of creating the visual and aesthetic aspects of a website. It involves designing the layout, color scheme, typography, graphics, and other visual elements that make up the website's appearance. Web designing is a crucial aspect of web development, as it determines how a website looks and feels to users.

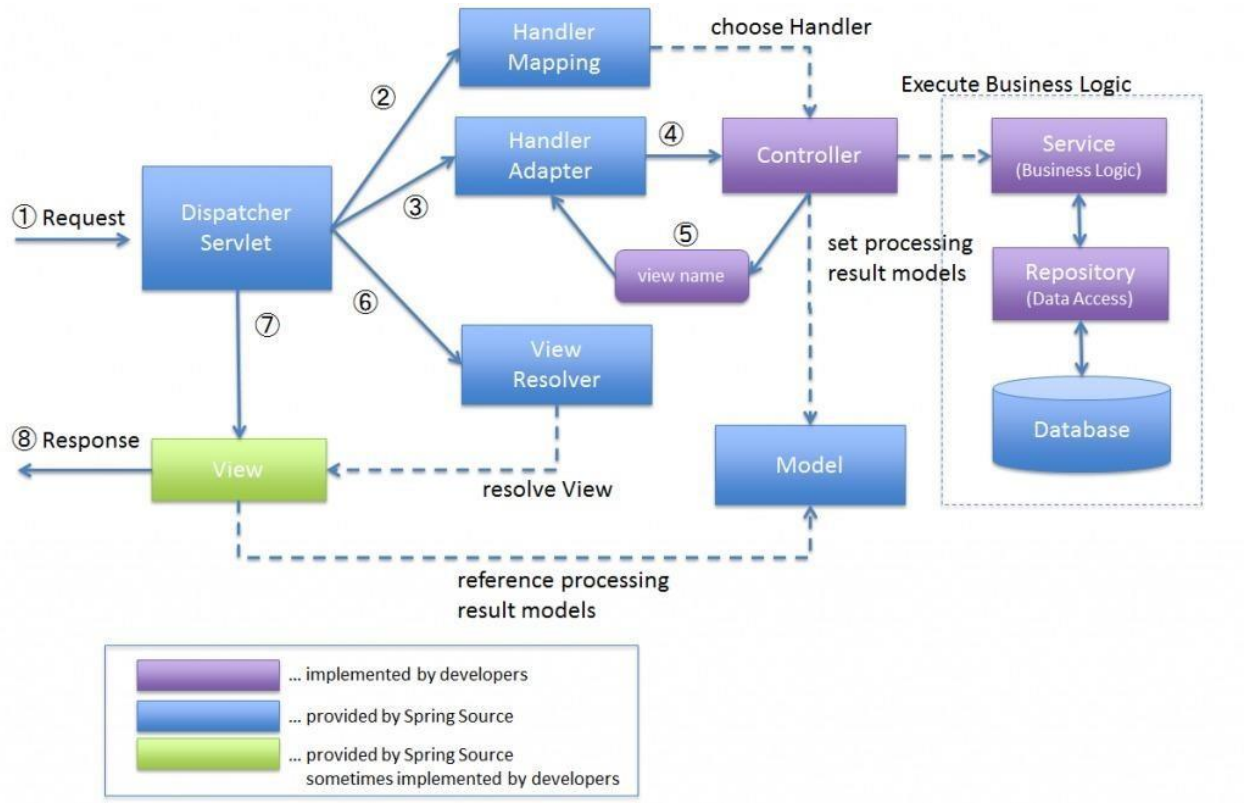
Web designing involves various disciplines, including graphic design, user interface (UI) design, user experience (UX) design, and coding. A web designer may work on designing the website layout, creating custom graphics and illustrations, choosing the appropriate typography, and ensuring that the website is accessible and easy to use for all users.

Web designers must also consider various factors, such as the website's target audience, the purpose of the website, and the branding and marketing goals of the business or organization that owns the website. The goal of web designing is to create an engaging and visually appealing website that reflects the brand's personality and effectively communicates the desired message to its users.

Web Application

A web application is a software application that is accessed and run through a web browser or web-based client. It is typically stored on a remote server and can be accessed from any device with an internet connection. Web applications are designed to perform specific functions or provide specific services to users, such as online shopping, social media, email, and file sharing.

They are built using web technologies such as HTML, CSS, JavaScript, and server-side scripting languages such as PHP or Python. Web applications can be accessed on a variety of devices including desktop computers, laptops, tablets, and smartphones.



MOBLIE ARCHITECTURE

Mobile architecture refers to the structure and organization of software systems and frameworks that enable the development and functioning of mobile applications. It encompasses the design principles, patterns, and technologies used to create robust, scalable, and efficient mobile applications. Here are some key mobile architecture concepts:

Native Architecture: Native mobile architecture involves developing applications specifically for a particular mobile platform, such as iOS or Android. It utilizes platform-specific programming languages (Objective-C or Swift for iOS, Java or Kotlin for Android) and frameworks (UIKit for iOS, Android SDK for Android) to create highly optimized and performant applications that leverage the full capabilities of the device**.**

Cross-Platform Architecture: Cross-platform architecture allows for the development of mobile applications that can run on multiple platforms using a single codebase. It involves frameworks like React Native, Flutter, or Xamarin, which enable developers to write code once and deploy it across different

platforms. Cross-platform architecture offers code reusability, faster development cycles, and cost-effectiveness.

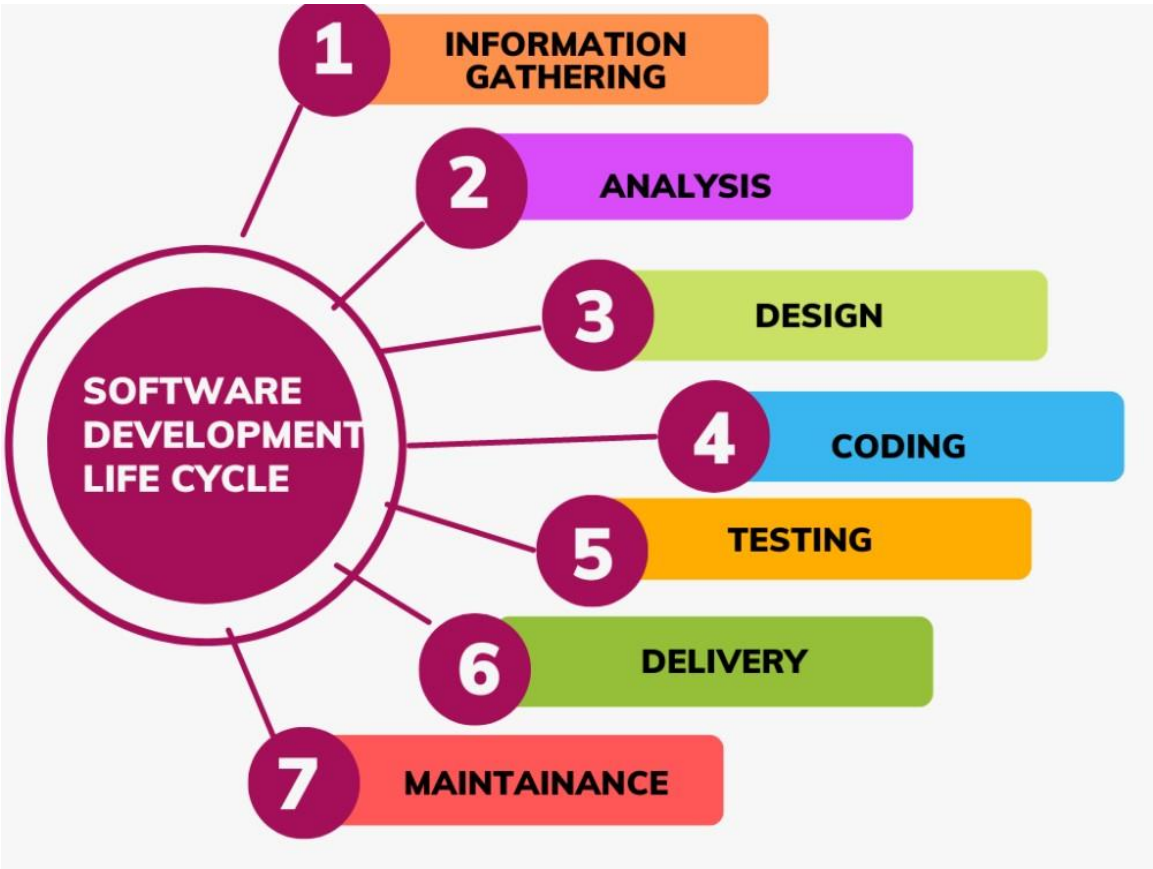
Model-View-Controller (MVC): MVC is a software architectural pattern commonly used in mobile development. It separates the application into three components: the model (data and business logic), the view (user interface), and the controller (handles user input and coordinates interactions between the model and view). MVC promotes code modularity, reusability, and easier maintenance.

Application Programming Interfaces (APIs): Mobile architecture often involves integrating with external APIs to access services and data from third-party providers. APIs enable functionalities such as retrieving data from web services, social media integration, payment gateways, and more. Architectural considerations include implementing proper API handling, authentication, error handling, and data caching.

Security and Privacy: Mobile architecture should address security concerns, including data encryption, secure communication protocols, secure storage, user authentication, and authorization mechanisms. It is essential to protect sensitive user information and ensure compliance with data privacy regulations.
Mobile architecture considerations vary depending on the specific platform, development approach (native or cross-platform), and project requirements. It is crucial to select an appropriate architecture that aligns with the goals of the mobile application, provides a good user experience, and supports scalability and maintainability in the long term.

SOFTWARE TESTING

Software Developement Life cycle



Software Development Life Cycle (SDLC) is a structured approach to software development that defines the various stages involved in creating and maintaining a software product. There are typically six phases in the SDLC:

Requirements gathering and analysis: In this phase, the requirements of the software are gathered from stakeholders, and analyzed to identify any potential issues and conflicts.

Design: Once the requirements have been identified, a software design is created to define the architecture, components, interfaces, and data of the software.

Development: The actual coding and implementation of the software is done in this phase, based on the design specifications.

Testing: Once the software has been developed, it is rigorously tested to ensure that it meets the requirements and specifications.

Deployment: The software is then deployed or released to the end-users.

Maintenance: After deployment, the software is regularly maintained and updated to fix any bugs or issues that arise, and to add new features or functionality.

The SDLC is a continuous process, with each phase building on the previous one. It helps ensure that the software development process is systematic and well-organized, reducing the risk of errors and issues, and increasing the chances of producing a high-quality software product that meets the needs of the stakeholders.

Software testing is the process of evaluating a software application or system to ensure that it meets specified requirements and functions as intended. It involves executing the software with the intention of finding defects, errors, or gaps in its functionality, usability, performance, security, and other quality attributes.

The main objectives of software testing are:

Finding Defects: The primary goal of testing is to identify defects or bugs in the software. This includes functionality errors, usability issues, performance bottlenecks, security vulnerabilities, and any other discrepancies between the expected and actual behavior of the software.

Ensuring Quality: Testing helps ensure that the software meets quality standards and fulfills user requirements. By detecting and resolving defects, testing helps improve the overall reliability, stability, and user satisfaction of the software. Validating Requirements: Testing verifies that the software functions according to the specified requirements. It ensures that all the features, functionalities, and interactions described in the software requirements are implemented correctly.

Enhancing Usability: Testing assesses the user-friendliness and ease of use of the software. It helps identify any issues related to the user interface, navigation, accessibility, and overall user experience, allowing for improvements to be made.

Assessing Performance: Testing evaluates the performance characteristics of the software, such as response time, scalability, resource usage, and reliability under various load conditions. It ensures that the software performs efficiently and meets performance expectations.

Ensuring Security: Testing helps identify vulnerabilities and weaknesses in the software that could be exploited by attackers. It includes testing for potential security breaches, data leaks, authentication issues, and other security-related risks.

Software testing is typically conducted through various techniques and methodologies, including manual testing and automated testing. It involves the creation of test plans, test cases, and test scripts to systematically evaluate the software's behavior. Testing can be performed at different stages of the software development lifecycle, such as unit testing, integration testing, system testing, and user acceptance testing (UAT).

Effective software testing requires a combination of technical skills, domain knowledge, attention to detail, and a structured approach to ensure comprehensive test coverage and accurate defect detection. It plays a critical role in delivering high-quality software products that meet user expectations and business requirements.

CONCLUSION

The Internship program gives us an opportunity to gain hands-on experience using real-life web development tools.

During this internship, I gained a lot of experience. It helped me develop my technical skills as well as gain personal growth.

FUTURE SCOPE

If someone does not have experience in the field, it can be a challenge to find work. It is possible for an individual to turn a successful internship into a career opportunity by gaining experience.

After an internship successfully completed, some future career possibilities include working for an IT company, becoming a software engineer, becoming a web designer, becoming a web developer, and becoming a quality assurance tester.