**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANASANGAMA, BELAGAVI-590018**

****

**INTERNSHIP REPORT (21INT822)**

**Submitted in Partial fulfillment of the Requirements for the VIII Semester of the Degree of**

**Bachelor of Engineering**

**In**

**Computer Science & Engineering**

**By**

**SHRUTI JAYASWAL (1CR21CS177)**

**Under the Guidance of Prof. Rajeshwari R**

**Assistant Professor, Dept. of CSE**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING CMR INSTITUTE OF TECHNOLOGY**

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

**2024-2025**



**DEPT. OF COMPUTER SCIENCE & ENGINEERING**

CERTIFICATE

This is to certify that **SHRUTI JAYASWAL (1CR21CS177),** student of CMR Institute of Technology have undergone Internship in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2024-2025**. It is certified that all corrections/suggestions indicated for Initial Reviews have been incorporated in the Report. This internship has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

**----------------------------- ------------------------**

**Sign. of Internal Guide Sign. of HOD**

**Prof. Rajeshwari R Dr. Kesavamoorthy**

**Assistant Professor Professor & HOD**

**Department of CSE Department of CSE**

**CMRIT, Bengaluru CMRIT, Bengaluru**

**External Viva**

Name of the examiner **Signature with date**

1.

2

DECLARATION

I, **Ms. Shruti Jayaswal,** USN **1CR21CS177**, bonafide student of CMR Institute of Technology, Bengaluru, affiliated to Visvesvaraya Technological University, Belagavi, hereby declare that the internship work entitled **“Full-Stack Developer Internship at CodeApto”** has been carried out by me during VIII semester of degree of Bachelor of Engineering in Computer Science and Engineering at CMR Institute of Technology, Bengaluru during the academic year 2024-25 under the esteemed guidance of Balwant Singh**.** The report is original, and it has not been submitted in part or full for any other degree in any University.

NAME: SHRUTI JAYASWAL USN: 1CR21CS177 SIGNATURE:

Place: Bengaluru Date:

**OFFER LETTER**

****

**COMPLETION CERTIFICATE**

****

**ABSTRACT**

During my 6-month internship at CodeApto as a Full Stack Developer, I had the opportunity to work extensively with the MERN stack (MongoDB, Express.js, React.js, and Node.js). My internship involved learning core concepts of full-stack development, gaining hands-on experience in building web applications, and understanding the software development life cycle.

I began by strengthening my knowledge of JavaScript, React for frontend development, and Node.js with Express for backend APIs. I also learned how to use MongoDB for database operations and Git for version control. Over the course of the internship, I worked on a task management project where users could post tasks (like increasing followers), and other users could accept and complete them, providing proof for credit-based rewards.

The internship helped me develop skills in REST API development, authentication using JWT, using Prisma as an ORM, role-based access control, and deployment practices. This experience improved my coding, debugging, and problem-solving skills and gave me a better understanding of how professional web applications are developed and maintained.

# ACKNOWLEDGEMENT

Any work of significance requires a great deal of effort and time put into it. But a factor of even greater importance is efficient guidance and encouragement. In spite of all my dedicated work, this internship would not have been possible without continuous help and guidance provided by people who gave their unending support right from when this idea was conceived.

I would like to thank to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant co- operation and support throughout this Internship tenure.

I would like to thank **Dr. Kesavamoorthy, Professor & Head,** Department of Computer Science and Engineering, CMRIT for her constant guidance and support during this Internship period.

I would like to thank my guide, **Prof. Rajeshwari R, Assistant Professor,** Department of Computer Science and Engineering, CMRIT for her support and mentoring that has been a great help with the Internship work.

I would like to thank my manager, **Balwant Singh** for their constant guidance that helped me in completing the Internship work successfully.

Lastly, I would like to thank **My Family** and **Friends** who have always supported me in every step of the Internship work.

**SHRUTI JAYASWAL**

**(1CR21CS177)**

**Table of Contents**

**CHAPTER 1**

1. Introduction 1
   1. [Establishment year and company activities overview 1](#_TOC_250027)
   2. About the Department/Team & Duration 2

[CHAPTER 2](#_TOC_250026)

1. Objectives 3-4

[CHAPTER 3](#_TOC_250025)

1. Scope of the Internship 5

[CHAPTER 4](#_TOC_250024)

1. Responsibilities and Tasks 6
   1. [AptoWork – Freelancing 8](#_TOC_250023)
      1. [Project Background and Relevance 8](#_TOC_250022)
      2. [Overview of the Project 8](#_TOC_250021)
      3. [Tools and Technologies Used 8](#_TOC_250020)
      4. [Work Performed 9](#_TOC_250019)
      5. [Results and Analysis 12](#_TOC_250018)
      6. [Learning Scope and Outcomes 12](#_TOC_250017)
   2. [Patient Enrollment 12](#_TOC_250016)
      1. [Project Background and Relevance 12](#_TOC_250015)
      2. [Overview of the Project 13](#_TOC_250014)
      3. [Tools and Technologies Used 13](#_TOC_250013)
      4. [Work Performed 13](#_TOC_250012)
      5. [Step-by-Step Evolution of Project 15](#_TOC_250011)
      6. [Results and Analysis 15](#_TOC_250010)
      7. [Learning Scope and Outcomes 15](#_TOC_250009)

[CHAPTER 5](#_TOC_250008)

1. Challenges and Problem-Solving 14
   1. [Freelancing services App 16](#_TOC_250007)
   2. Patient Enrolment 16
   3. Common Problem-Solving Flow 17

[CHAPTER 6](#_TOC_250006)

1. Skills Developed 18
   1. [Technical Skills 20](#_TOC_250005)

* 1. [Problem Solving and Debugging 21](#_TOC_250004)
  2. [Project Management and Workflow Planning 22](#_TOC_250003)
  3. [Communication and Collaboration 22](#_TOC_250002)

[CHAPTER 7](#_TOC_250001)

1. Outcome of the Internship 21

[CHAPTER 8](#_TOC_250000)

1. Conclusion 22

REFERENCES 23

#### LIST OF FIGURES

Fig 4.1 Flowchart for website Project Workflow 11

Fig 4.2: Flowchart for Patient Enrollment 14

Fig 5.1 Problem solving flowchart 19

#### LIST OF TABLES

* 1. Project Evolution- Patient Enrollment 15
  2. : Challenges vs Solution- Portfolio website 17
  3. Challenges vs Solution- Patient Enrollment 18
  4. Skills acquired 21

**CHAPTER 1**

# INTRODUCTION

CodeApto is a forward-thinking consulting and development firm dedicated to helping startups tackle their most complex challenges with custom-built solutions. With a strong focus on innovation and adaptability, the company offers specialized support that aligns with the unique needs of businesses across diverse industries like tech, e-commerce, and more.

What sets CodeApto apart is its ability to deliver localized strategies that blend deep industry knowledge with an understanding of regional regulations and market trends. Whether it's regulatory guidance or tech integration, CodeApto ensures startups are equipped not just for today’s challenges but also for tomorrow’s opportunities.

### Establishment Year and Company Activities Overview

This internship was conducted at CodeApto, incorporated on February 19, 2023. The company serves as a dynamic technology startup focused on delivering innovative, tailor- made solutions for businesses across various domains. With a strong emphasis on creativity, scalability, and user-centric development, CodeApto acts as a catalyst for digital transformation. The organization fosters collaboration between professionals and clients, aiming to revolutionize how modern digital solutions are designed, developed, and delivered. During my internship at CodeApto, I was part of the web development team focusing on full-stack application development using the MERN stack. The company primarily worked on building customized web applications and platforms for clients, offering services like task automation tools, user management systems, and API integrations.

As an intern, I participated in daily stand-up meetings, collaborated with developers and mentors, and contributed to the development of key features in ongoing projects. The team followed agile development practices, which helped me understand real-world workflows, sprint planning, code reviews, and testing. I also got exposure to working with version control systems (GitHub), third-party APIs, and deployment tools.

### About the Department & Duration

I joined CodeApto as a Full Stack Developer Intern in the Web Development Department, with the internship spanning from November 1, 2024, to May 6, 2025. The internship was focused on giving hands-on experience in modern web technologies, real-world project development, and software development best practices.

During the initial phase of the internship, I was introduced to the MERN stack—MongoDB, Express.js, React.js, and Node.js. I learned the basics of each technology, including how to structure backend APIs with Node and Express, design frontend interfaces with React, and manage data using MongoDB. This training helped me understand how to build full-stack applications efficiently.

After completing the learning phase, I developed a basic project using the MERN stack. This project helped me apply the concepts I learned, including CRUD operations, user authentication, routing, and component-based development in React. It served as a strong foundation for working on more complex real-world applications.

In the later phase of my internship, I contributed to a major project called AptoWork, which was similar to a freelancing task management platform. The platform allowed users to post tasks (like increasing social media followers), and other users could accept and complete them. After submission and approval of proof, credits were awarded. I worked on key features such as user authentication, task assignment, role-based access, and secure API integration.

Overall, the internship at CodeApto enhanced both my technical and practical understanding of full-stack development. It gave me valuable exposure to real-time problem-solving, teamwork, and project delivery under deadlines.

## CHAPTER 2

# OBJECTIVES

1. **To Gain Practical Exposure to Full-Stack Software Development in a Startup Environment** The primary aim of this internship at CodeApto was to gain hands-on experience in full-stack web development using the MERN stack (MongoDB, Express.js, React.js, Node.js). In the initial months, I focused on learning the fundamentals of each technology in the stack. Later, I built a basic MERN project to apply my skills and gradually moved on to contributing to a larger, more complex product—AptoWork, a freelancing service platform. This journey helped me understand the full development lifecycle, from UI design and API creation to backend logic and database management.
2. **To Understand Professional Practices and the Dynamics of Startup Culture**

Working at CodeApto gave me valuable exposure to a collaborative, fast-paced startup environment. I regularly interacted with developers, mentors, and project leads, which enhanced my communication, coordination, and adaptability. Presenting my progress, receiving constructive feedback, and incorporating it into my work helped me become more confident and self-directed. The experience taught me how to work independently and efficiently while also being an effective team player.

##### To Understand Agile Methodologies and Collaborative Development Processes

I During the internship, we followed an agile-like workflow, using tools such as Git and GitHub for version control and task tracking. I participated in sprint planning, code reviews, and milestone-based deliverables, which helped me learn how to break down features into manageable tasks and deliver them on time. Understanding how collaborative development works in a professional setup greatly improved my organizational and time management skills.

##### To Strengthen Problem-Solving and Debugging Skills Through Real-World Challenges

Throughout the development of AptoWork and other internal projects, I encountered real-world challenges involving UI responsiveness, backend errors, and integration issues. With guidance from mentors, I learned to use debugging tools, inspect console errors, and implement fixes using clean and reusable code. Code reviews and pair programming sessions played a key role in helping me understand industry best practices and maintainable code structure.

##### To Contribute Meaningfully to Real-Time Projects and Build a Strong Technical Portfolio

One of the most fulfilling aspects of this internship was contributing to real, production-level projects. Apart from a basic practice project, I actively contributed to AptoWork—a freelancing service web app developed by CodeApto. I worked on features like user onboarding, project listings, and secure authentication. By taking ownership of modules and collaborating with the team, I delivered functional components that were integrated into the live system. These contributions now form a strong part of my technical portfolio and reflect the growth I achieved during this internship.

## CHAPTER 3

# SCOPE OF INTERNSHIP

The internship at The Design Theeta offered a comprehensive and engaging platform for applying The internship at CodeApto offered a comprehensive and practical experience in full-stack web development using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The scope covered both learning and contributing phases, starting with foundational training and gradually moving toward the development of production-ready features.

During the initial phase, I focused on strengthening my technical understanding of the MERN stack and modern web development practices. I explored the basics of frontend design using React, backend API development using Node and Express, and NoSQL database handling with MongoDB.

As I progressed, I took on increasingly complex tasks, including the development of a personal practice project and contributions to a major real-time web application—AptoWork, a freelancing service platform. I gained experience working on features like user authentication, data flow, dashboard UI, and role-based access control using JWT.

In addition to technical tasks, I became familiar with real-world software development processes such as version control with Git, agile task planning, and collaborative coding through regular meetings and code reviews.

The internship provided not only technical exposure but also helped develop soft skills such as time management, communication, and teamwork in a startup environment. The entire scope of the internship was designed to bridge the gap between academic knowledge and industry requirements, ensuring I was job-ready by the end of the program.

## CHAPTER 4

# RESPONSIBILITIES AND TASKS

### AptoWork – Freelancing Platform

#### Project Background and Relevance

AptoWork is a full-stack web application developed to connect freelancers and clients, similar to platforms like Fiverr or Upwork. The platform allows users to register, create projects, send offers, manage gigs, and collaborate effectively. This project was relevant for understanding real-world startup demands and building scalable, production-grade applications.

#### Overview of the Project

The project focused on creating a multi-role system for admins, freelancers, and clients with features like authentication, job posting, offer tracking, and messaging. As part of a team, I contributed to both front-end and back-end modules of the application using the MERN stack.

#### Tools and Technologies Used

* + - * **MongoDB:**

Used as the database to store dynamic data such as contact form submissions or project details. It provided flexibility in handling JSON-like documents and was well-suited for the portfolio's lightweight data needs.

#### Express.js:

Served as the backend framework for building RESTful APIs. It handled routing, form submissions, and communication between the frontend and MongoDB. It also helped in setting up middleware for request parsing and validation.

#### React.js:

The primary library used for building the user interface. React’s component-based structure allowed me to create reusable and modular UI elements such as navigation bars, service cards, project showcases, and the contact form. React Router was used to manage client-side routing between pages.

#### Node.js:

Provided the runtime environment to run JavaScript on the server side. Node powered the Express backend and enabled asynchronous handling of API requests.

* + - * Other Tools: Git & GitHub, Visual Studio Code, Postman (for API testing), Figma (for design reference)

#### Work Performed

The **AptoWork** project involved building a comprehensive freelancing platform using the MERN stack (MongoDB, Express.js, React.js, Node.js) to connect freelancers and clients, allowing them to post projects, send offers, and communicate directly. This section outlines the key stages of development, technical challenges faced, and solutions implemented throughout the project.

1. Requirement Gathering and Wireframing

The project started with a detailed analysis of user requirements, which involved multiple discussions with mentors, clients, and other stakeholders to understand the platform's key features: user registration, project posting, offer management, and messaging. Based on this input, I created user stories and wireframes using Figma, which helped in visualizing the platform’s flow, user journey, and page layout. This was crucial for ensuring that the platform met both functional and user experience goals.

1. Frontend Development with React

I developed the frontend using React.js, creating modular and reusable components such as the navbar, service cards, project showcase sections, and the contact form. Key implementation steps included:

* Using React Router for smooth navigation between pages.
* Applying Bootstrap and custom CSS for responsive layouts.
* Using React hooks (useState, useEffect) for managing component states.

Challenges included ensuring consistent spacing and styling across different screen sizes, which I resolved using a mobile-first approach and media queries. I also implemented smooth scroll behavior and animated transitions to enhance user experience.

1. Backend API with Express.js and Node.js

For backend functionality, I built an Express.js server that handled HTTP POST requests from contact form.

Tasks included:

* Creating an API route to receive and validate form data.
* Storing the form submissions in MongoDB.
* Sending response messages back to the frontend for user confirmation.

The backend was hosted on Render, and I used Postman to test API responses and debug any request issues. Middleware was implemented to handle errors and sanitize inputs for security. The backend was hosted on Render, and I used Postman to test API responses and debug any request issues. Middleware was implemented to handle errors and sanitize inputs for security.

1. MongoDB Integration and Data Handling

I set up a MongoDB database (hosted on MongoDB Atlas) to store contact form entries. The schema defined fields such as name, email, subject, and message. Mongoose was used to define the schema and interact with the database. Proper validation checks and error handling ensured that malformed data was filtered out before storage.

1. Deployment and Hosting

The React frontend was deployed using Netlify, and the backend was deployed on Render. Environment variables were used for sensitive data like database URIs. I ensured CORS compatibility between frontend and backend and tested both independently and as a complete unit post-deployment.

1. Version Control and Repository Management

I maintained the entire project on GitHub, using Git for version control. Each stage of development— UI creation, backend integration, bug fixes—was committed with clear messages. The repository was structured into client and server folders to separate frontend and backend code.

Learning Highlights

* Gained hands-on experience with React component structuring and hooks.
* Learned how to build and connect a Node-Express API with a MongoDB database.
* Understood deployment strategies using Netlify and Render.
* Improved debugging, responsive design practices, and clean code documentation.
* Learned real-world practices in API testing, error handling, and project collaboration.

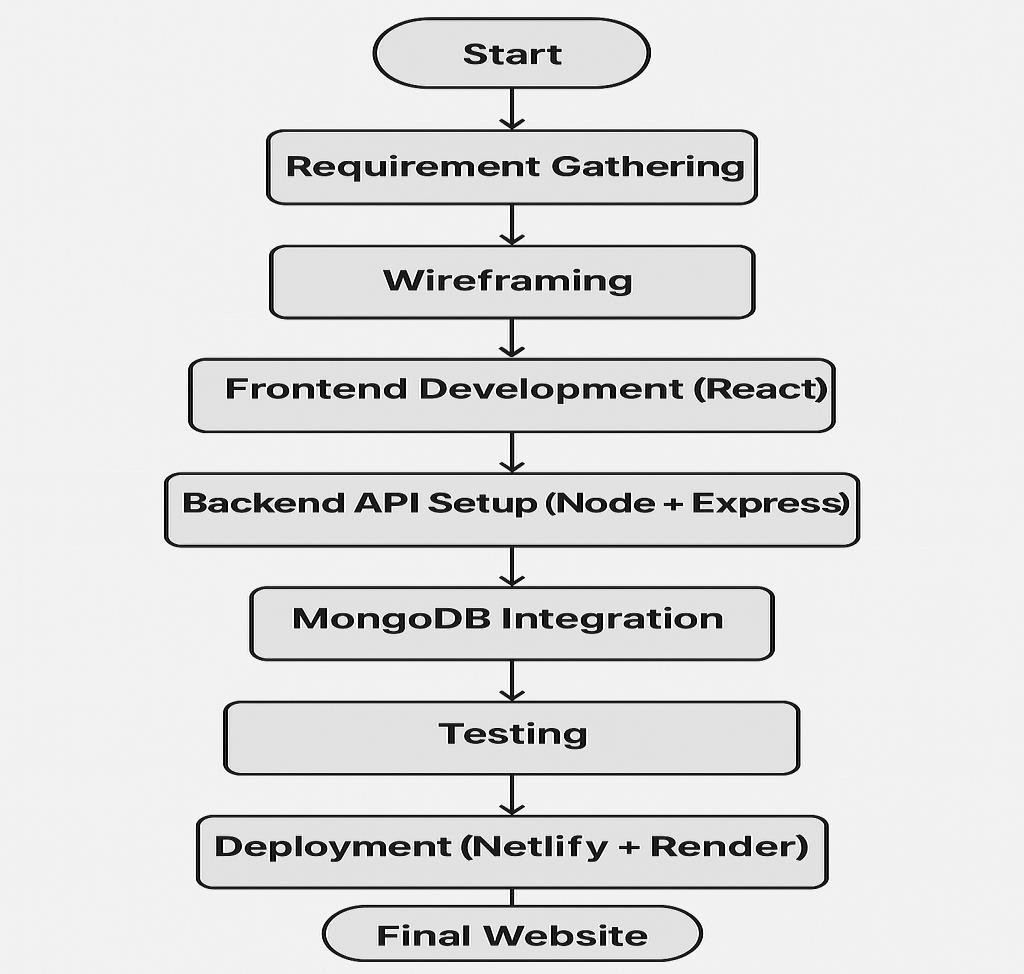


Fig 4.1: Flowchart of Project Workflow

#### Results and Analysis

The successful development and deployment of the AptoWork platform marked a major milestone in delivering a functional and user-friendly freelancing system. The project achieved its core goals of allowing users to register, post projects, send offers, and manage communications between clients and freelancers. The platform’s clean UI, responsiveness, and logical user flow received positive feedback from mentors and peers. React components were effectively used to maintain consistency and modularity, while backend APIs ensured smooth data handling and secure operations. Hosting the frontend on Netlify and the backend on Render showcased my practical understanding of deployment pipelines. The seamless integration of MongoDB for storing user and project data further confirmed the robustness and scalability of the system. Overall, AptoWork not only demonstrated the practical implementation of a full-stack project but also added a valuable, production-ready application to my portfolio.

#### Learning Scope and Outcomes

This project offered a rich learning experience by covering the full software development lifecycle using the MERN stack. I gained practical skills in building responsive and interactive UIs with React, implementing backend logic using Node.js and Express.js, and handling data storage and retrieval through MongoDB. I also learned essential concepts in RESTful API design, user authentication using JWT, and secure data handling. Through the process of debugging and testing (using Postman and manual test cases), I strengthened my problem-solving abilities. Hosting the application taught me how to handle environment variables, CORS configuration, and deployment workflows. Additionally, managing the codebase on GitHub and practicing version control helped improve my collaboration readiness and code organization. Overall, this internship project significantly boosted my confidence and competence as a full-stack web developer.

### Patient Enrollment

#### Project Background and Relevance

The company offers a solution called AccelerateRx, which facilitates batch enrollment of patients into its messaging and engagement programs. This is a key feature during onboarding and for recurring mass enrollment events. However, current operations have shown that a significant number

This results in decreased patient engagement, frustrated clients, and unnecessary support overhead. The Patient Enrollment project was launched to investigate and resolve these inefficiencies by ensuring more accurate and inclusive batch enrollments.

#### Overview of the Project

This project aims to identify where and why eligible patients are being left out during the batch enrollment process and offer solutions to optimize speed, scale, and data accuracy. The initiative involves analyzing the existing process for gaps, enhancing diagnostic scripts, storing enriched enrollment data for analysis, and building reports that clients can act upon.

By doing so, the project seeks to provide more inclusive, efficient, and actionable enrollment operations across all client databases—ensuring that all truly eligible patients have a chance to participate.

#### Tools and Technologies Used

* + - * **Python** – Core development environment for scripting and logic.
      * **Pandas** – Data handling and CSV report generation.
      * **SQLAlchemy** – Database engine and connection pooling.
      * **MySQL** – Source database for audit and patient tables.
      * **json\_repair** – Repairing malformed encrypted JSON payloads.
      * **dotenv** – Secure handling of environment variables and credentials.
      * **ThreadPoolExecutor (Python)** – Multithreading for performance optimization.
      * **Pickle/CSV** – Exporting final data and reports.
      * **Shell/CLI** – Script execution and automation environment.

#### Work Performed

* + - * Set up environment configurations using .env for date ranges, credentials, and host mappings.
      * Loaded and decrypted encrypted audit payloads using AES logic and json\_repair for robust JSON handling.
      * Extracted globalPatientID values from nested JSON responses, even those malformed or corrupted.
      * Built queries to verify if these patients existed in the customer table and checked their current status.
      * Classified each patient ID into:
        + Found but unenrolled (customer\_status = 0)
        + Found but inactive (customer\_status = 1)
        + Or truly missing
      * Generated detailed client-wise reports with metrics such as enrollment accuracy and execution duration.

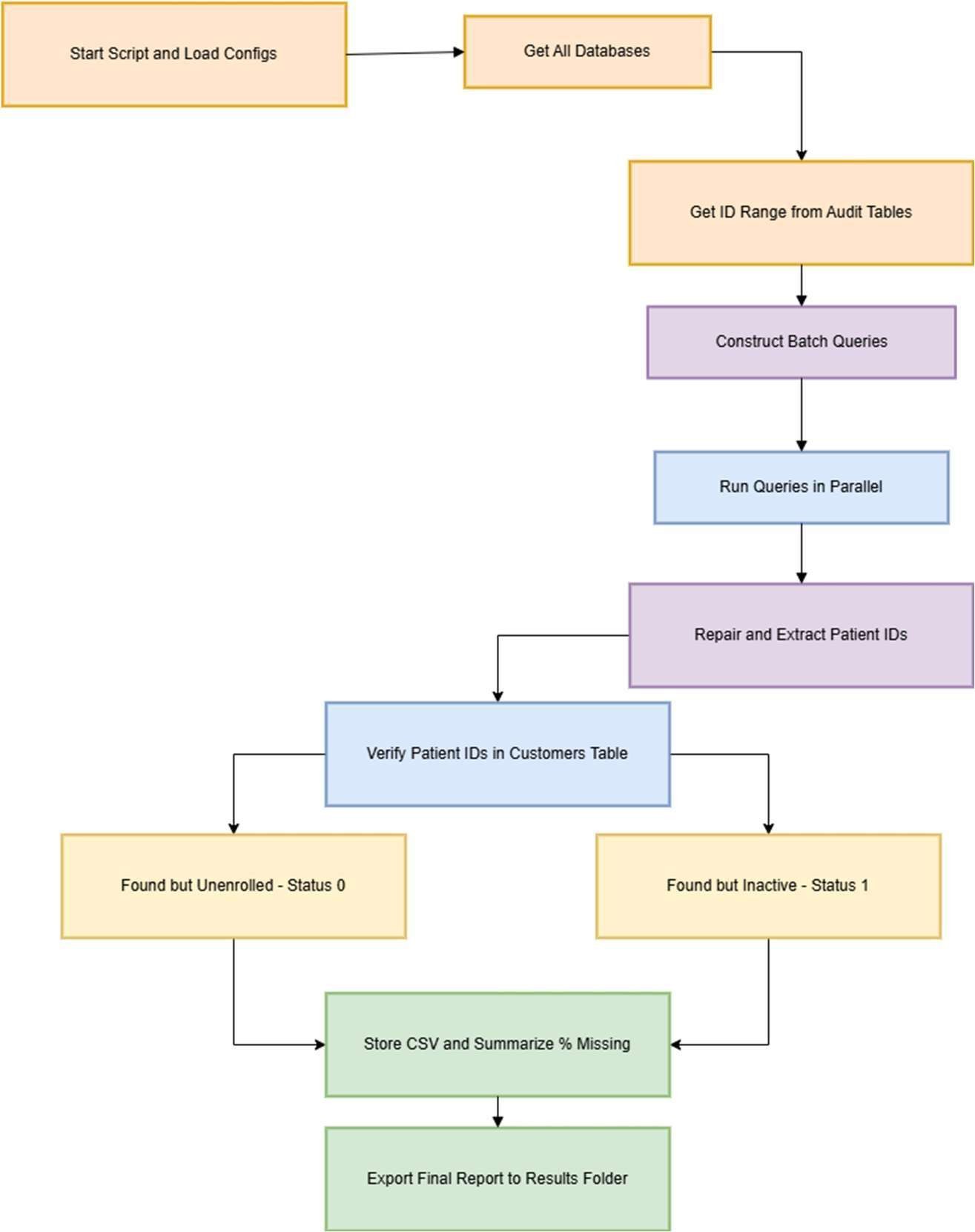


Fig 4.2 Flowchart for Patient Enrollment

#### Step-by-Step Evolution of Project

|  |  |  |
| --- | --- | --- |
| **Phase** | **Task** | **Outcome** |
| Phase 1 | Extract audit responses | Identified enrollment error patterns |
| Phase 2 | JSON repair and decryption | Enabled reading malformed logs |
| Phase 3 | Verification of patient records | Determined true missing vs filtered records |
| Phase 4 | Customer classification by status | Structured report based on actionable insight |
| Phase 5 | Threaded processing and batching | Reduced processing time significantly |
| Phase 6 | Report generation | Delivered CSV outputs for client-side analysis |

Table 4.1 Project Evolution- Patient Enrollment

#### Results and Analysis

The optimized patient enrollment script revealed that many flagged “missing” patients were actually enrolled but remained inactive due to status issues. Enhancements such as multithreading, improved filtering logic, and client-specific batch reporting significantly boosted classification accuracy and reduced processing time. These improvements enhanced visibility for clients and enabled internal teams to take proactive measures, reducing manual effort and support overhead.

#### Learning Scope and Outcomes

This project provided hands-on experience with:

* + - * Analyzing real-world data issues in patient enrollment workflows
      * Secure handling and decryption of sensitive healthcare data
      * Identifying gaps in batch processing logic and enrollment pipelines
      * Building automated diagnostics and reporting tools using Python and SQL

It deepened my understanding of how accurate, scalable, and secure data handling is essential for improving patient engagement, reducing client escalations, and ensuring operational excellence in the healthcare technology domain.

## CHAPTER 5

# CHALLENGES AND PROBLEM SOLVING

During my internship at Apto, I encountered several technical and logical challenges while working on the Patient Enrollment system. These challenges sharpened my problem-solving skills and exposed me to real-world issues in healthcare data processing. They also helped me build confidence in handling production-level systems. Below is a categorized overview of the key challenges I faced and the solutions I implemented.

### Freelancing Services App

* + - **Responsive Layout Issues:** The UI became distorted on mobile devices due to improper column alignment and fixed widths. I fixed this by leveraging Bootstrap’s responsive grid system and writing custom media queries to ensure consistent rendering across different screen sizes.
    - **User Authentication Bugs:** Users were occasionally redirected incorrectly or logged out abruptly. This was caused by issues in token verification and session handling. I resolved it by implementing JWT-based authentication and storing tokens securely in cookies/localStorage.
    - **Service Form Submission Errors:** The service request form initially failed to connect with the backend due to CORS errors and mismatched request formats. I configured proper CORS policies on the backend and standardized the request payload structure.
    - **MongoDB Document Mismatches:** Submitted data didn’t reflect properly in MongoDB because of mismatches between frontend form fields and Mongoose schema. I reviewed and realigned the schema fields, added validation, and sanitized input data.
    - **Payment Integration Issues:** Integration with a third-party payment gateway initially failed due to incorrect callback URLs and environment misconfiguration. I fixed this by securely storing API keys in .env files and validating all transaction endpoints.

|  |  |  |
| --- | --- | --- |
| Challenge | Problem | Solution |
| Responsive layout issues | Site looked broken on smaller screens | Used Bootstrap grid and media queries for responsiveness |
| Form submission errors | API requests blocked due to CORS policy | Configured CORS middleware on Express backend |
| MongoDB integration failure | Contact form data not storing in database | Fixed schema mapping and added input validation |
| Deployment disconnects | Frontend-backend communication failed post-deployment | Corrected base URLs and environment variable setup |

Table 5.1 Challenges vs Solution- Portfolio Website

### Patient Enrollment

* + - **Decryption Failures:** Some audit payloads failed to decrypt correctly, often due to encoding issues or partial data loss. I resolved this by adding exception handling around AES decryption and using fallback logic to attempt partial recovery with json\_repair.
    - **Inaccurate Enrollment Statuses:** Many patients were incorrectly flagged as missing. After deeper investigation, I introduced status-based classification logic that accurately distinguished between inactive, unenrolled, and truly missing patients.
    - **Slow Script Execution:** The original scripts ran sequentially and took a long time over large datasets. I improved performance by implementing ThreadPoolExecutor to run queries concurrently in batches
    - **Database Mapping Errors:** Data mismatches occurred when fetching records from the customer table. I corrected this by validating column mappings and adding consistency checks for globalPatientID values across different databases.
    - **Static Configuration Challenges:** Hard-coded configurations made it difficult to adapt scripts for different environments. I resolved this by externalizing parameters using .env files and dynamic runtime options.

|  |  |  |
| --- | --- | --- |
| Challenge | Problem | Solution |
| Corrupted audit data | Encrypted JSONs were malformed | Used json\_repair to recover usable data |
| Query speed | Slow ID-range queries in large audit logs | Parallelized query execution using ThreadPoolExecutor |
| Misleading patient status reports | All missing patients were counted equally | Verified against customer\_status to categorize status = 0 and 1 |
| Sensitive data handling | Needed secure encryption/decryption | Used AES decryption in secure, parameterized queries |
| Multi-client DB automation | Needed to run for multiple client schemas | Automated detection & filtering of valid schemas for safe batch processing |

Table 5.2 Challenges vs Solution- Patient Enrollment

### 5.4 Common Problem-Solving Flow

These challenges helped me understand the importance of secure, scalable, and accurate data processing, especially in domains like healthcare where mistakes can have a direct impact on patient experience and client trust. Debugging real-world data, handling encryption securely, and building automation for multi-database environments all played a key role in improving my technical proficiency and professional preparedness.

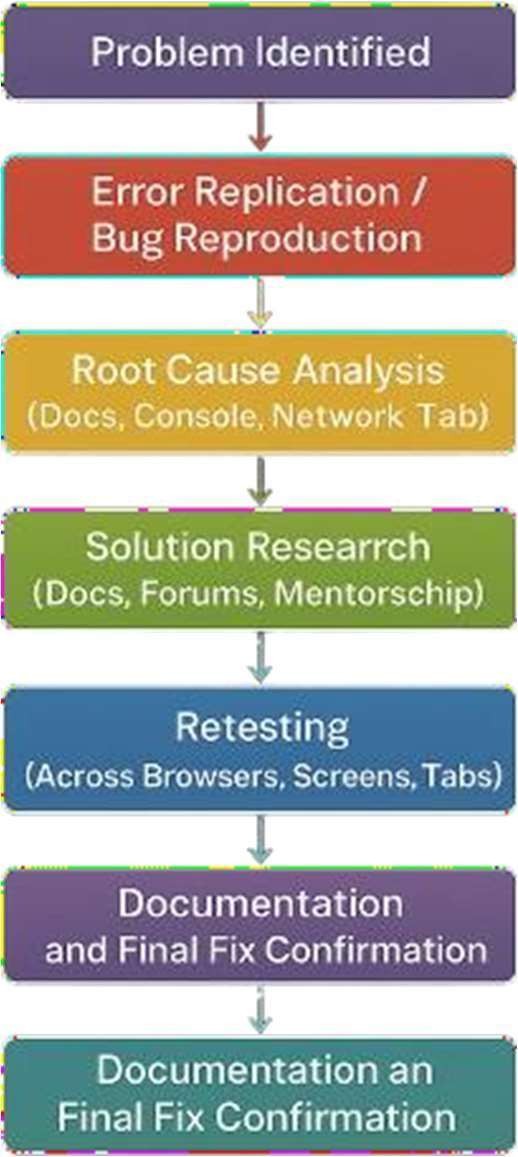


Fig 5.1 Problem solving flowchart

## CHAPTER 6

# SKILLS DEVELOPED

During my work on the Freelancing Services App, I gained valuable experience across various stages of full-stack application development. This project helped me strengthen both my technical and soft skills, as I tackled real-world requirements involving dynamic user interactions, secure data handling, and scalable backend logic. In addition to coding, I also developed project planning, requirement analysis, and collaborative communication skills through iterative development and testing Technical Skills

This project helped me strengthen both my technical and soft skills, as I tackled real-world requirements involving dynamic user interactions, secure data handling, and scalable backend logic. In addition to coding, I also developed project planning, requirement analysis, and collaborative communication skills through iterative development and testing.

|  |  |
| --- | --- |
| **Skill** | **Context of Use** |
| **React.js** | Developed dynamic UI components and handled routing |
| **Node.js & Express.js** | Built backend APIs to handle form submissions and route data |
| **MongoDB** | Stored and managed contact form data and enrollment records |
| **HTML5, CSS3,**  **Bootstrap** | Designed responsive layouts for multiple screen sizes |
| **JavaScript (ES6)** | Added interactivity, form logic, and state handling |
| **API Integration** | Connected frontend with backend for real-time data flow |
| **Debugging (DevTools, Console)** | Traced bugs in layout, network requests, and data issues |

|  |  |
| --- | --- |
| **Batch Processing Logic** | Identified enrollment inefficiencies and improved script performance |
| **Conditional Data Filtering** | Optimized enrollment by filtering misclassified or inactive entries |
| **Version Control (Git & GitHub)** | Managed code versions, tracked issues, and organized project repositories |

Table 6.1 Skills Acquired

### Problem Solving and Debugging

Working on the Freelancing Services App exposed me to a variety of real-world development challenges that significantly strengthened my problem-solving and debugging skills. From broken page layouts to data not saving correctly in the database, I learned how to analyze issues logically and implement targeted fixes.

One major issue was form data not being submitted due to improper backend routing and missing validations. I debugged this by tracing the request-response flow, adjusting form field handling in PHP, and ensuring correct query execution in MySQL. I also faced layout inconsistencies across devices, which I resolved using Bootstrap’s responsive grid and browser dev tools to fine-tune spacing and alignment.

Database-related bugs, like duplicate records and mismatched user-role data, taught me the importance of input validation and session tracking. I used structured error logging and test accounts to simulate user behaviors and identify corner cases. These experiences taught me that debugging isn’t just about fixing code—it’s about understanding how the system behaves, learning from mistakes, and improving the overall application architecture.

### Project Management and Workflow Planning

Building the Freelancing Services App required organized planning, modular execution, and consistent version control to manage multiple components like user registration, job posting, and profile management.

I started by outlining the app’s key modules, then implemented features in logical phases—such as authentication, dashboard creation, and service listing. I tracked my progress using Git and GitHub, with clearly named commits to maintain version history and allow easy rollback when needed.

Managing multiple user types (freelancers and clients) involved handling different workflows, so I created separate scripts and views for each role. I used .env files to separate environment-specific configurations like database credentials, enhancing both security and deployment flexibility. This structured workflow helped me stay organized and ensured that each feature was testable, scalable, and easy to maintain.

### Communication and Collaboration

While the development work was mostly individual, effective communication and documentation were essential, especially when sharing progress and ensuring the project was client-ready.

I regularly documented system workflows, database schema relationships, and test procedures so that the client could understand how the app functioned. When clarifying project requirements, I asked targeted questions to avoid miscommunication and stayed aligned with the client’s expectations.

This experience taught me how to explain technical decisions to non-technical stakeholders, respond constructively to feedback, and ensure that deliverables met real-world usability needs. It also highlighted the value of clear and proactive communication in freelance and client-based projects.

## CHAPTER 7

# OUTCOME OF THE INTERNSHIP

My six-month internship, focused on developing a Full-Stack MERN application, was an enriching and hands-on experience that allowed me to bridge the gap between theoretical knowledge and real-world application. During this period, I worked extensively on a Freelancing Services App project, which was a dynamic and multifaceted platform. This experience helped me deepen my understanding of full-stack web development and sharpen my skills in building end-to-end applications.

Throughout the internship, I strengthened my technical expertise in MongoDB, Express.js, React.js, and Node.js (MERN), while learning how to design, develop, and deploy a functional web application. I gained practical experience in handling real-world development challenges such as optimizing API performance, managing user authentication and authorization, improving database queries, and implementing responsive UI designs.

Apart from technical growth, I developed strong project management skills. This included planning feature releases, coordinating tasks using tools like Jira and Trello, and maintaining version control with GitHub. Managing tasks independently while working collaboratively in a team helped me gain valuable insights into professional software development workflows.

One of the most rewarding aspects of this internship was seeing how my work contributed directly to the project’s success. I was able to observe users interacting with the app, experiencing firsthand how thoughtful development leads to smoother workflows and better user experiences. This gave me a sense of ownership and reinforced the impact of delivering quality, user-centered solutions.

Ultimately, the internship has provided me with a well-rounded skill set that I can confidently apply to future full-stack development roles. It has also fueled my passion for building scalable, efficient, and user-friendly web applications that can solve real-world problems.

## CHAPTER 8

# CONCLUSION

My six-month internship at CodeApto was an invaluable experience that provided me with practical exposure to the world of full-stack development. Working on the Freelancing Services App helped me apply theoretical concepts in a real-world setting, strengthening my technical skills in both front-end and back-end technologies, including React.js, Node.js, Express, and MongoDB.

Throughout this internship, I learned the importance of collaboration, effective communication, and time management within an agile development environment. I faced challenges that required creative problem-solving and debugging, which ultimately enhanced my ability to work efficiently and make sound technical decisions under pressure.

Contributing to the development of the app not only boosted my confidence in handling complex coding tasks but also deepened my passion for building user-centered web applications. The hands-on experience, combined with guidance from my mentors, has given me a clear understanding of how quality software can significantly improve user experience and operational efficiency.

Overall, this internship solidified my career aspirations in software development. I now feel well-prepared to take on future projects with the skills, confidence, and experience necessary to succeed in the fast-evolving tech industry.

# REFERNCES

1. CodeApto – Official Instagram Page https://[www.instagram.com/designtheeta/](http://www.instagram.com/designtheeta/)
2. Udyam Registration Portal. (n.d.). Government of India MSME Registration Lookup. https://udyamregistration.gov.in/
3. React Documentation – A JavaScript library for building user interfaces https://react.dev/
4. Node.js Documentation – JavaScript runtime built on Chrome's V8 engine https://nodejs.org/
5. Express.js – Web framework for Node.js https://expressjs.com/
6. MongoDB Documentation – NoSQL Database https://[www.mongodb.com/docs/](http://www.mongodb.com/docs/)
7. Bootstrap – Frontend Component Library https://getbootstrap.com/
8. Youtube https://[www.youtube.com/](http://www.youtube.com/)
9. VS Code https://code.visualstudio.com/
10. GitHub https://github.com/