A Project Report

On

**“DOCTOR APPOINTMENT SYSTEM”**

Submitted by:

GAVIN LIVERO

232CSC02

Under the Guidance of:

Dr. B.G Prashanthi

Department of Computer Science



**Department of Computer Science**

**St Joseph’s University**

**Bengaluru**

Individual/Self copy

**Certificate**



This is to certify that GAVIN LIVERO (Reg No : 232CSC02) has successfully completed the project titled **“DOCTOR APPOINTEMENT”** at St Joseph’s University under the supervision and guidance in the fulfilment of requirements of Third semester, Masters of Science(Computer Science), Bengaluru, India.

1. Dr. B.G Prashanthi

Dr. B.G Prashanthi

Head of Department

Computer Science

**ACKNOWLEDGEMENT**

We recognize that the magnitude of this project necessitated the cooperation, guidance, and support of many individuals, and we are immensely grateful for their assistance throughout the project's completion.

We extend our sincere appreciation to our Principal, Fr. Dr. Victor Lobo, S.J., for providing us with the necessary facilities and resources to carry out this project.

We are indebted to Dr. B.G Prashanthi, Department of Computer Science at St. Joseph's University, for granting us permission to undertake this project.

Our heartfelt thanks go to our project guide, Prof. Mrinmoyee Bhattacharya, Dr. Shashikala D, Dr. Nithya B from the Department of Computer Science at St. Joseph's University, for their invaluable guidance and continuous encouragement, which played a pivotal role in shaping this project.

We also want to express our gratitude to all the teaching and non-teaching staff members of the Computer Science department at St. Joseph's University for their unwavering support and encouragement, which greatly contributed to the successful completion of our project.

Last but not least, we are deeply appreciative of everyone who, whether directly or indirectly, contributed to the success of this project. Your support was instrumental in making this endeavour a reality

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **CONTENTS** | **PAGE NO** |
| 1 | ABSTRACT |  |
| 2 | INTRODUCTION |  |
| 3 | CHAPTER 1: |  |
|  | * Problem definition |  |
|  | * Objectives |  |
|  | * System requirements |  |
| 4 | CHAPTER 2: |  |
|  | * Development Tools/Technologies |  |
| 5 | CHAPTER 3: |  |
|  | * Requirements |  |
| 6 | CHAPTER 4: |  |
|  | * Content Diagram |  |
| 7 | CHAPTER 5: |  |
|  | * Implementation |  |
| 8 | CHAPTER 6: |  |
|  | * Coding |  |
| 9 | REFERENCES |  |

**ABSTRACT**

The healthcare industry is experiencing a digital transformation, with a growing reliance on technology to enhance patient care and streamline services. As telemedicine and online healthcare become more prevalent, there is an increasing need for efficient systems that allow patients to easily access medical appointments while enabling doctors to manage their schedules effectively.

The Doctor Appointment Web Application **MEDIKARE** addresses these needs by offering a user-friendly platform for scheduling appointments. Patients can seamlessly book, reschedule, or cancel appointments in real-time, while doctors benefit from tools to optimize their availability and interactions with patients.

Developed using the **MERN Stack** (MongoDB, Express, React, Node.js), the application utilizes this technology’s scalability and performance. This stack enables high-performance user interfaces and robust backend functionality, allowing for real-time updates and secure data handling. Overall, this solution enhances patient-provider interactions and highlights the crucial role of technology in modern healthcare.

**INTRODUCTION**

Due to the increasing demand for more efficient service delivery and better patient care, the healthcare sector is changing quickly. Systems that let patients easily make appointments and help medical professionals run their offices efficiently are in high demand given the increasing popularity of telemedicine and online healthcare solutions. This need is met by the Doctor Appointment Web Application **MEDIKARE**, which provides a user-friendly interface for users to quickly plan, reschedule, or cancel appointments in real time. With the help of this software, patients and physicians may communicate better and organise their schedules more efficiently.

For this project, the **MERN Stack** (MongoDB, Express, React, Node.js) is crucial, as it integrates the most effective features of modern web development technologies. **MongoDB** offers a flexible and scalable database capable of handling various data structures, while **Express** simplifies server-side operations, making it easier to develop robust APIs. **React** delivers an interactive and dynamic user interface, ensuring an outstanding user experience, and **Node.js** allows for fast and efficient data processing, enabling the application to handle multiple requests effectively.

By leveraging the **MERN Stack**, the Doctor Appointment Web Application provides a secure, adaptable, and scalable solution that meets the evolving demands of the healthcare industry, ultimately enhancing patient care and accessibility.

**CHAPTER 1**

**PROBLEM DEFINITION**

The healthcare industry struggles with inefficient appointment management, leading to frustrated patients and increased administrative workloads for providers. Traditional booking methods often result in communication errors and missed appointments. As telemedicine rises, there is growing demand for a digital solution that streamlines scheduling and enhances patient-provider interaction. **MEDIKARE** aims to address these challenges by offering a user-friendly platform for patients to manage appointments efficiently, while enabling healthcare providers to optimize their schedules and improve patient care.

**OBJECTIVES**

With the **MEDIKARE** project, we aim to develop an intuitive platform that transforms how patients schedule, manage, and track their medical appointments. Our goal is to create a seamless system where patients can easily book, reschedule, or cancel appointments, ensuring convenience and accessibility. By the end of the project, we aim to enhance communication between patients and healthcare providers, facilitating better interaction and improving overall healthcare experiences through real-time updates and secure access to important medical information.

**SYSTEM REQUIREMENTS:**

* Operating System: 64-bit Windows 10 or higher.
* Coding Platform: Visual Studio Code.
* Server: MongoDB.
* Web Browser: Latest versions of Chrome, Firefox, Brave.
* Processor: Dual-core processor or better (e.g., Intel i3/i5 or equivalent).
* RAM: Minimum of 4 GB (8 GB recommended for smoother multitasking).
* Network Speed: 2Mbps or higher.

**MODULES USED IN THIS PROJECT:**