

Web Application Vision Optical

Project Report

KUDSE222F-045

KUDSE222F-039

KUDSE222F-031

Date of Submission: 01.01.2024



Vision Optical Eye Care Center

No : 112/2 Variyapola

Kurunegala

DECLARATION

We, the team members of Inline Workspace, collectively declare that this project report titled "Vision Optical Portfolio Website" is the result of our own original research and work. It has not been submitted in part or full for any other degree or qualification in any university or institution. All sources used for the completion of this report have been duly acknowledged.

We understand the consequences of submitting plagiarized work and hereby affirm that all contributions from external sources have been properly cited and referenced. Any material not our own, which is included in this report, has been identified with the appropriate credits.

We further declare that the work presented in this report reflects our understanding and knowledge of the subject matter and is a true representation of the efforts invested by Inline Workspace in this project.

Date: 11.17.2023

Signatures:

K M U S KONARA

N M S T SARATHCHANDRA

N B U U BALALLA

ABSTRACT

The "Vision Optical Portfolio Website" project, undertaken by Inline Workspace, is a comprehensive web application designed to enhance the online presence, operational efficiency, and user experience of Vision Optical. The primary objectives of this project are to streamline the appointment scheduling process for users, establish a robust user account management system, and lay the foundation for the seamless integration of a billing system.

KEY OBJECTIVES

- **Appointment Management:** The web application allows users to conveniently schedule appointments with Vision Optical. The admin panel enables efficient monitoring and maintenance of appointment statuses.
- **User Account System:** Users can create accounts to manage their appointments, providing a personalized and seamless experience. This feature enhances user engagement and facilitates effective communication.
- **Email Confirmation System:** A robust email confirmation system notifies users upon successful appointment scheduling, ensuring clarity and reducing the likelihood of missed appointments.
- **Future Billing System Integration:** The project is designed with scalability in mind, paving the way for the future integration of a billing system to streamline financial transactions.

METHODOLOGIES

The project adopts an Agile development methodology, utilizing sprints and regular feedback loops to facilitate flexibility and responsiveness to changing requirements. The analysis phase involves fact-finding techniques to understand the existing system and identify areas for improvement. The design phase incorporates process models, use case diagrams, class diagrams, and activity diagrams to provide a blueprint for system development.

OUTCOMES

The Vision Optical Portfolio Website emerges as a user-friendly platform that not only simplifies appointment scheduling but also establishes a foundation for enhanced user interaction and future expansions. The integration of an admin panel, user accounts, and email confirmation contributes to a seamless and efficient user experience.

In conclusion, the "Vision Optical Portfolio Website" project by Inline Workspace successfully addresses key objectives, leveraging modern web technologies to create a dynamic and user-centric platform for Vision Optical and its clientele.

ACKNOWLEDGMENT

The successful completion of the "Vision Optical Portfolio Website" project by Inline Workspace would not have been possible without the invaluable support and contributions of various individuals and organizations. We extend our heartfelt gratitude to:

Vision Optical: We express our sincere appreciation to the Vision Optical team for their cooperation, insights, and continuous support throughout the project. Their commitment to innovation and excellence has been instrumental in shaping the success of this endeavor.

Project Supervisor: Our gratitude goes to Madam Sumudu, our project supervisor, for providing guidance, valuable feedback, and unwavering support. Your expertise and encouragement have been crucial in steering the project in the right direction.

Inline Workspace Team Members: A special thank you to all members of Inline Workspace who worked collaboratively, contributing their skills and dedication to ensure the project's success. Your teamwork and commitment have been key to overcoming challenges and achieving milestones.

Beta Testers and Users: We appreciate the individuals who participated in beta testing and provided valuable feedback, helping us refine and improve the functionality and user experience of the Vision Optical Portfolio Website.

Educational Institution: We acknowledge the support received from NIBM for providing the necessary resources, infrastructure, and conducive learning environment that facilitated the execution of this project.

TABLE OF CONTENTS

1. Introduction

1.1 Introduction

1.2 Motivation for the Project

1.3 Objectives of the Project

1.4 Scope of the Project

1.5 Structure of the Dissertation

2. Analysis

2.1 Introduction

2.2 Fact Finding Techniques

2.3 Current System and Processes

2.4 Drawbacks of the Current System

2.5 Similar Systems and Literature Review

2.6 Functional Requirements

2.7 Non-Functional Requirements

3. Design

3.1 Introduction

3.2 Overview of Process Models

3.3 Introduction to Selected Model

3.4 High-Level Use Case Diagram

3.5 Class Diagram

3.6 Database Design

3.6 User Interface Design

4. Implementation

4.1 Introduction to Implementation

4.2 Implementation Environment

4.3 System Developed Tools and Technologies

4.4 Major Code Segments

5. Evaluation

5.1 Introduction

5.2 System Test Plan

5.3 System Test Cases

5.4 User Evaluation

6. Conclusion

6.1 Introduction

6.2 Problems Encountered

6.3 Future Improvements

6.4 Lessons Learned

7. References

8. Appendix

9. Client Certificate

LIST OF FIGURES

- Figure 1: High-Level Use Case Diagram
- Figure 2: Class Diagram
- Figure 4: Database Schema
- Figure 5: User Interface Screenshots

List of Tables

- Table 1: Functional Requirements Matrix
- Table 2: Non-Functional Requirements Checklist
- Table 3: System Test Cases and Results

List of Acronyms

- API: Application Programming Interface
- UI: User Interface
- UX: User Experience
- SQL: Structured Query Language
- QA: Quality Assurance

1.1 Introduction

The "Vision Optical Portfolio Website" project, spearheaded by Inline Workspace, is a strategic initiative aimed at developing a dynamic web application to bolster the digital presence and operational efficiency of Vision Optical. This introduction serves as the project's foundational cornerstone, providing a comprehensive overview of its purpose, significance, and overarching goals.

1.2 Motivation for the Project

The impetus behind undertaking the "Vision Optical Portfolio Website" project is rooted in the dynamic landscape of modern business and customer engagement. Recognizing the growing importance of online accessibility and streamlined interactions, Vision Optical is motivated to create a centralized digital platform. This platform not only caters to the convenience of its clientele but also aligns with contemporary technological trends. The motivation is to enhance the customer experience, improve operational workflows, and position Vision Optical as a digitally adept service provider.

1.3 Objectives of the Project

1.3.1 Streamline Appointment Scheduling

The primary objective is to develop a user-friendly interface facilitating seamless appointment scheduling for Vision Optical's customers. This feature aims to reduce wait times, enhance customer satisfaction, and optimize the overall appointment management process.

1.3.2 Establish a User Account Management System

Implementation of a robust user account management system is another key objective. This system empowers customers to create accounts, manage appointments, and receive personalized notifications. This not only enhances customer engagement but also fosters a sense of connectivity with the optical services.

1.3.3 Lay the Foundation for Billing System Integration

A forward-thinking approach is adopted to design the architecture and framework of the web application, ensuring readiness for the integration of a billing system in the future. This strategic objective positions the project for scalability and adaptability to future enhancements.

1.4 Scope of the Project

The scope of the "Vision Optical Portfolio Website" project encompasses the development and implementation of key features:

1.4.1 Appointment Scheduling

The project will deliver a feature-rich appointment scheduling system accessible to users, facilitating efficient appointment setting.

1.4.2 User Account Management

The web application will include a comprehensive user account management system, empowering customers to create accounts, manage appointments, and receive relevant notifications.

1.4.3 Future Billing System Integration

While not immediately implemented, the project's scope includes laying the groundwork for the integration of a billing system in future phases, ensuring adaptability to evolving business needs.

1.5 Dissertation Structure

Chapter 1: Introduction

1.1 Sets the Context for the Project:

Industry overview and Vision Optical's position.

1.2 Introducing its Purpose:

The essential purpose of the "Vision Optical Portfolio Website" project.

1.3 Motivation for the Project:

Driving forces and motivations behind the project.

1.4 Objectives of the Project:

Measurable goals, including appointment scheduling and account management.

1.5 Scope of the Project:

Features and functionalities, including future billing system integration.

Chapter 2: Analysis

2.1 Explores the Current System and Processes:

In-depth examination of existing systems and processes.

2.2 Identifies Drawbacks:

Critical analysis of current system limitations.

2.3 Reviews Similar Systems and Literature Review:

Examination of comparable systems and relevant literature.

2.4 Establishes Functional and Non-functional Requirements:

Defining functional and non-functional criteria for the project.

Chapter 3: Design

3.1 Outlines the Design Phase:

Overview of the design phase's importance.

3.2 Includes Process Models:

Consideration and selection of appropriate process models.

3.3 Introduces Use Case Diagrams:

High-level visualization of user interactions.

3.4 Incorporates Class Diagrams:

Blueprint of key classes and their relationships.

3.5 Utilizes Activity Diagrams:

Visual representation of system actions.

3.6 Encompasses Database Design:

Structuring and detailing the database.

3.7 Addresses User Interface Design:

Principles and considerations for a user-friendly interface.

Chapter 4: Implementation

4.1 Describes the Implementation Phase:

Introduction to the practical execution of the project.

4.2 Covers the Environment:

Specification of the development environment.

4.3 Discusses Tools, Technologies, and Code Segments:

Utilized tools, technologies, and key code segments.

Chapter 5: Evaluation

5.1 Discusses the Evaluation Phase:

Significance of evaluating the implemented system.

5.2 Presents the System Test Plan:

Outlining the plan for system testing.

5.3 Details System Test Cases:

Specific cases for evaluating system performance.

5.4 Discusses User Evaluations:

Methodologies for gathering and presenting user feedback.

Chapter 6: Conclusion

6.1 Summarizes the Project:

Concise summary of the entire project.

6.2 Highlights Encountered Problems:

Reflection on challenges faced during the project.

6.3 Proposes Future Improvements:

Suggestions for future enhancements based on lessons learned.

6.4 Shares Lessons Learned:

Key takeaways and insights gained from the project.

References

ANALYSIS

2.1 Introduction

In this section, the analysis phase is introduced, outlining its significance in understanding the current state of Vision Optical's operations and the need for the "Vision Optical Portfolio Website" project. The introduction sets the stage for a comprehensive examination that spans from the existing processes to potential improvements.

2.2 Fact Finding Techniques

To gather pertinent information for the analysis, various fact-finding techniques will be employed:

Surveys and Questionnaires:

- Utilized to collect structured data from potential users, existing patients, and staff.

- Aims to understand user preferences, needs, and expectations for the web application.

Interviews:

- Conducted with key stakeholders, including administrators and potential users.

- Provides in-depth insights into specific requirements and pain points.

Observation:

- On-site observation of the current appointment scheduling process, patient interactions, and administrative workflows.

- Reveals valuable information for improving user experience and administrative efficiency.

2.3 Current System and Processes

This section involves a detailed exploration of Vision Optical's current system and processes:

Appointment Scheduling:

- Analysis of the existing method of appointment scheduling.

- Identification of the key steps involved and the technologies in use.

User Account Management:

- Evaluation of the current approach to user account management.

- Assessment of how users currently create and manage their accounts.

Administrative Processes:

- Examination of the workflows followed by the administrative staff.

- Identification of pain points and areas for improvement.

2.4 Drawbacks of the Current System

An in-depth analysis of the current system reveals certain drawbacks:

Inefficiencies in Appointment Scheduling:

- Possible delays or inefficiencies in the current appointment scheduling process.

- Identifying areas where the system can be streamlined for better user experience.

Limited User Account Features:

- Evaluation of limitations in the current user account management system.

- Identifying features that can be enhanced or added for improved user engagement.

Administrative Challenges:

Identification of challenges faced by the administrative staff in managing appointments.

Highlighting areas where the system can facilitate administrative tasks.

2.5 Similar Systems and Literature Review

This section involves a thorough examination of similar systems and relevant literature:

Competitor Analysis:

Review of portfolio websites or appointment scheduling systems used by similar businesses.

Identification of best practices and innovative features.

Literature Review:

Exploration of academic and industry literature related to web applications in the healthcare domain.

Insights into technological trends and user expectations in similar contexts.

2.6 Functional Requirements

The functional requirements for the "Vision Optical Portfolio Website" project are established:

Appointment Scheduling:

Users should be able to schedule appointments seamlessly through the web application.

Confirmation notifications to users and administrative staff.

User Account Management:

User account creation and management functionalities.

Personalized user profiles with appointment history.

Administrative Dashboard:

Dashboard for administrators to view and manage appointments.

Real-time status updates on appointments.

Email Notification System:

Automated email notifications to users upon appointment scheduling.

Reminder emails for upcoming appointments.

User Authentication:

Secure user authentication for account creation and access.

2.7 Non-Functional Requirements

The non-functional requirements for the project include:

Performance:

The web application should handle a specified number of simultaneous users without performance degradation.

Security:

Implementation of robust security measures to protect user data and maintain confidentiality.

Scalability:

The system should be designed to accommodate future enhancements, such as the integration of a billing system.

Usability:

The user interface should be intuitive and user-friendly, catering to users with varying levels of technical proficiency.

Reliability:

The system should be reliable, with minimal downtime and disruptions to service.

DESIGN

3.1 Introduction

This section introduces the design phase, emphasizing its pivotal role in translating requirements into a tangible and user-friendly system. Design is crucial to ensure that the "Vision Optical Portfolio Website" aligns with user expectations, operational efficiency, and future scalability.

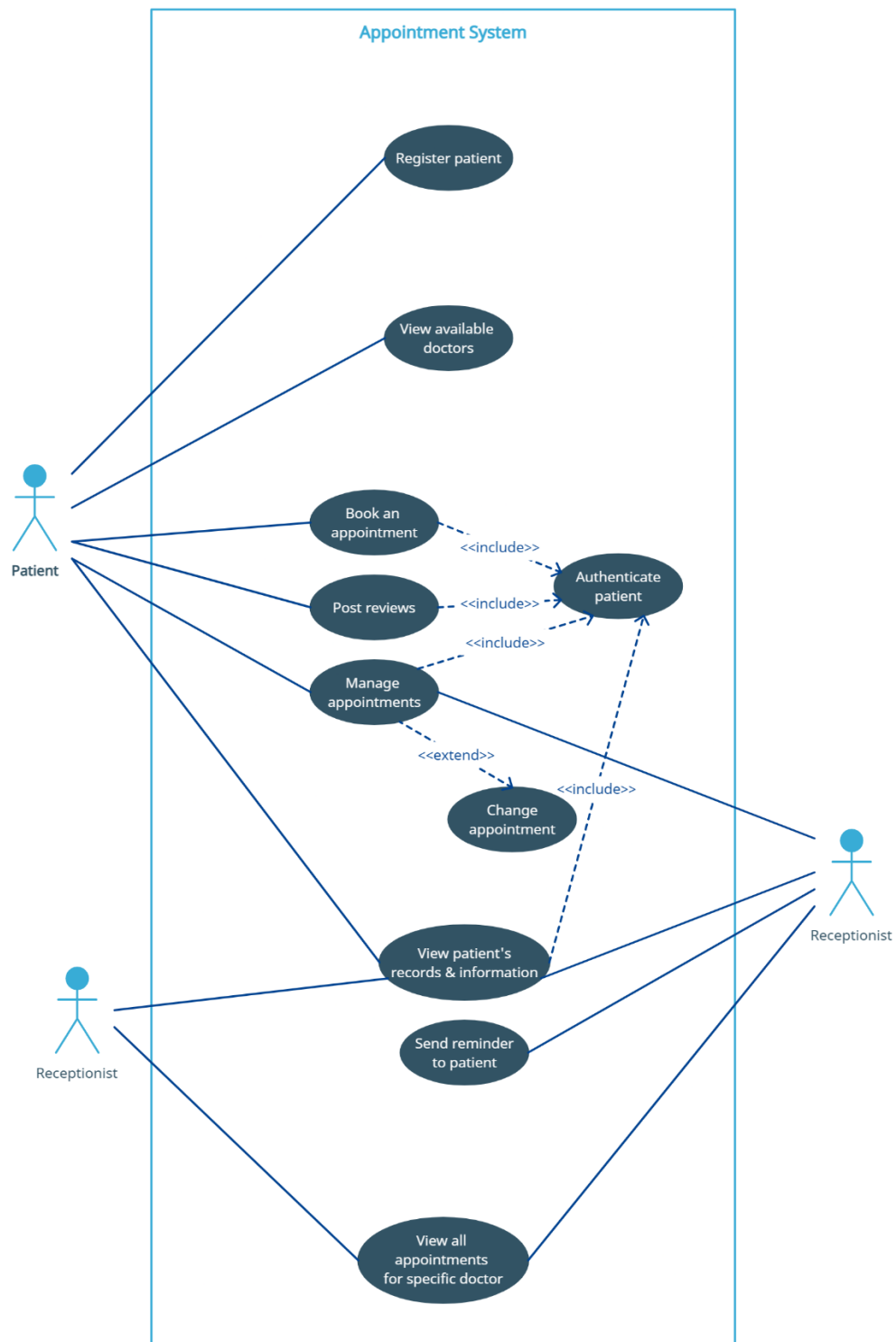
3.2 Overview of Process Models

Various process models will be considered for the project. Given the dynamic nature of the web application development, an Agile methodology is deemed suitable. Agile allows for iterative development, accommodating changes based on user feedback, which aligns with the project's objectives.

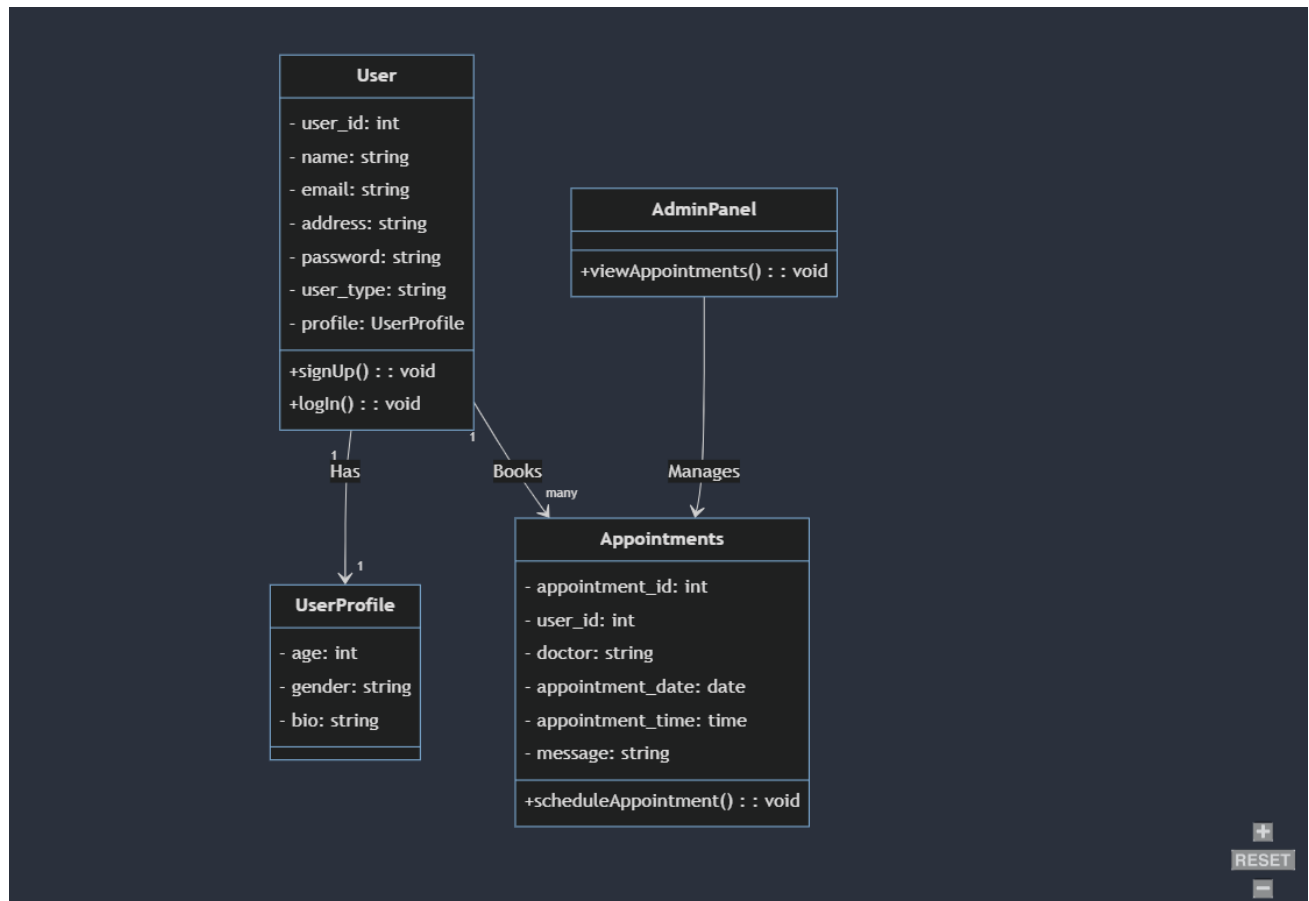
3.3 Introduction to Selected Model

The Agile methodology is selected for its adaptability and responsiveness to evolving user needs. This model involves iterative cycles, each focusing on specific functionalities. This iterative approach ensures that the project remains flexible to changes and improvements throughout its development.

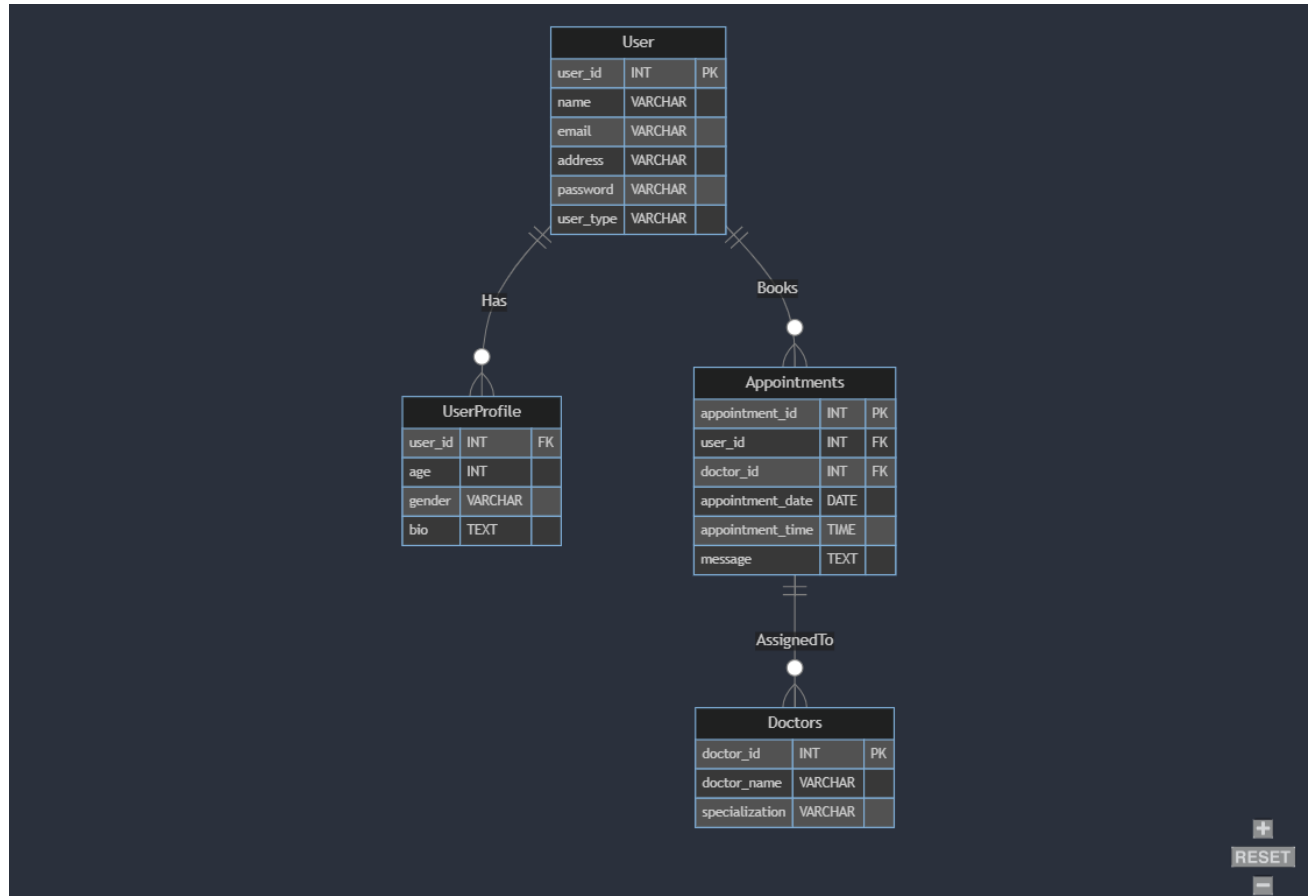
3.4 High-Level Use Case Diagram



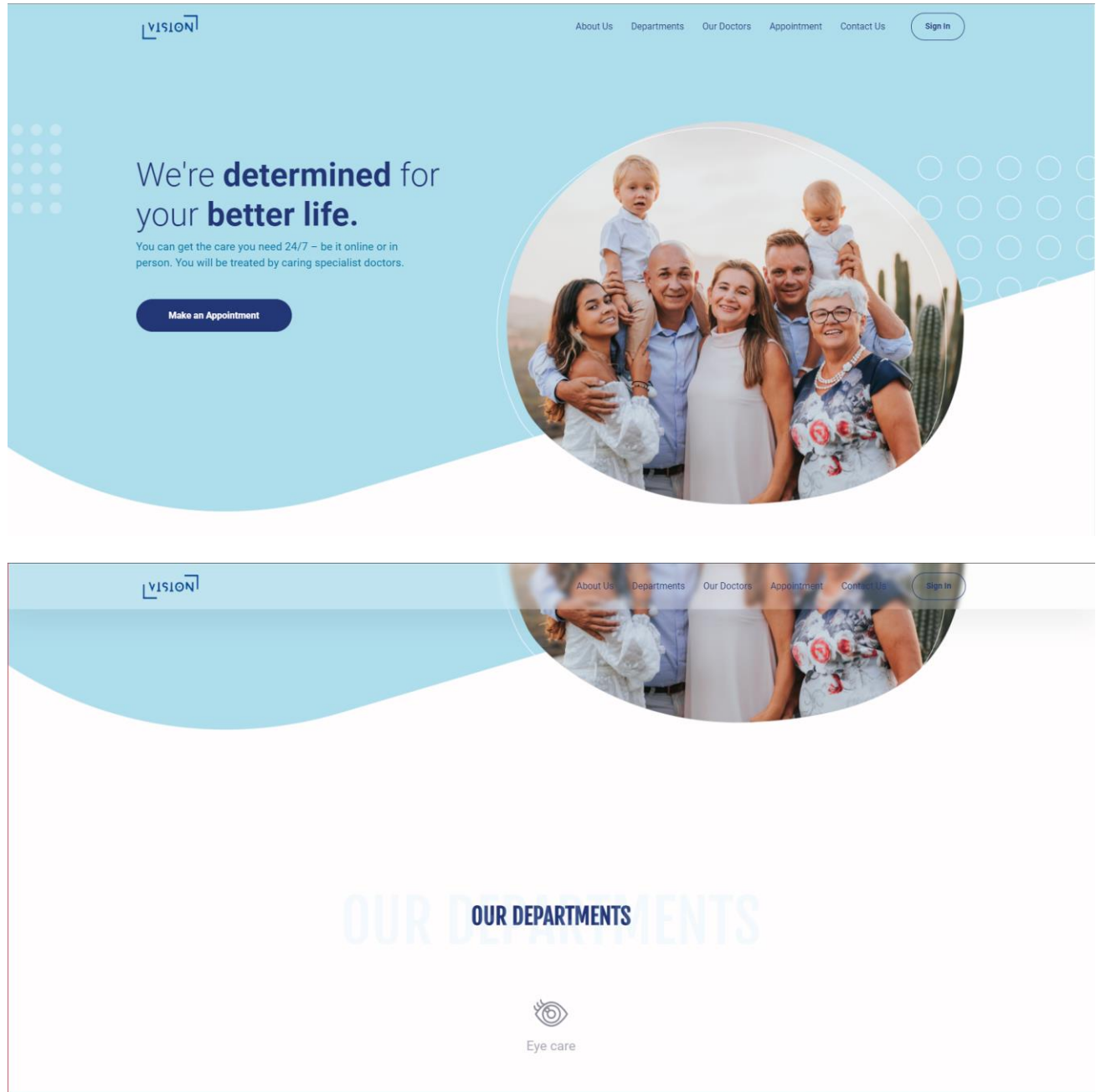
3.5 Class Diagram

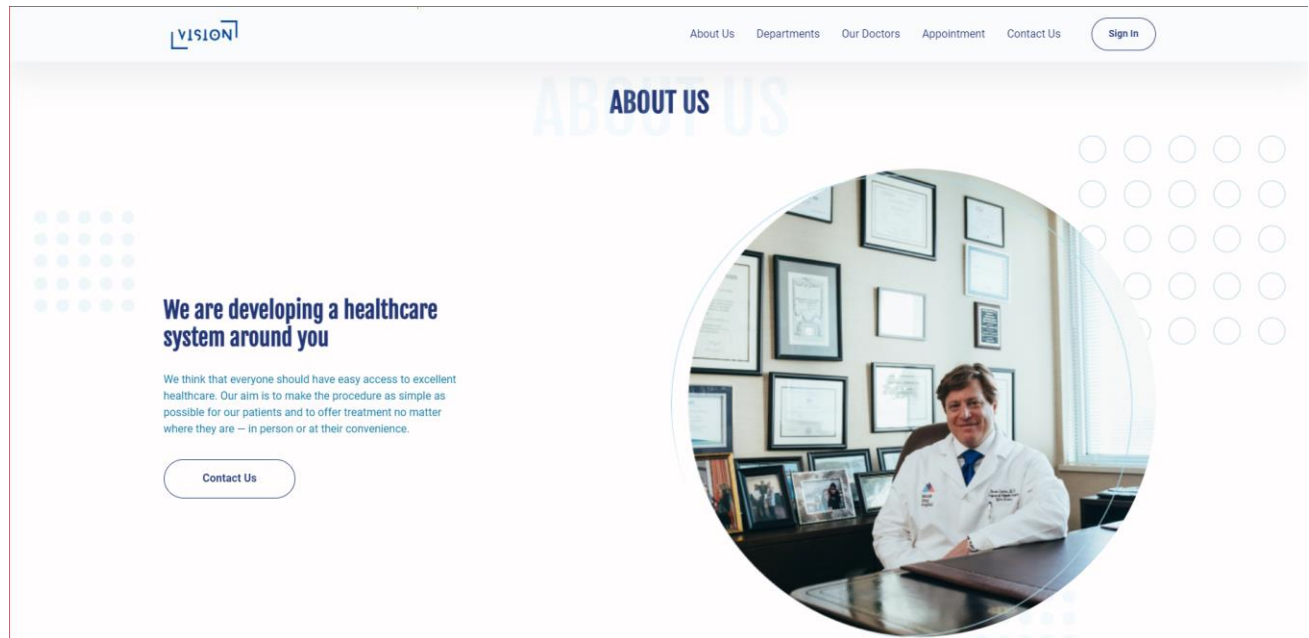
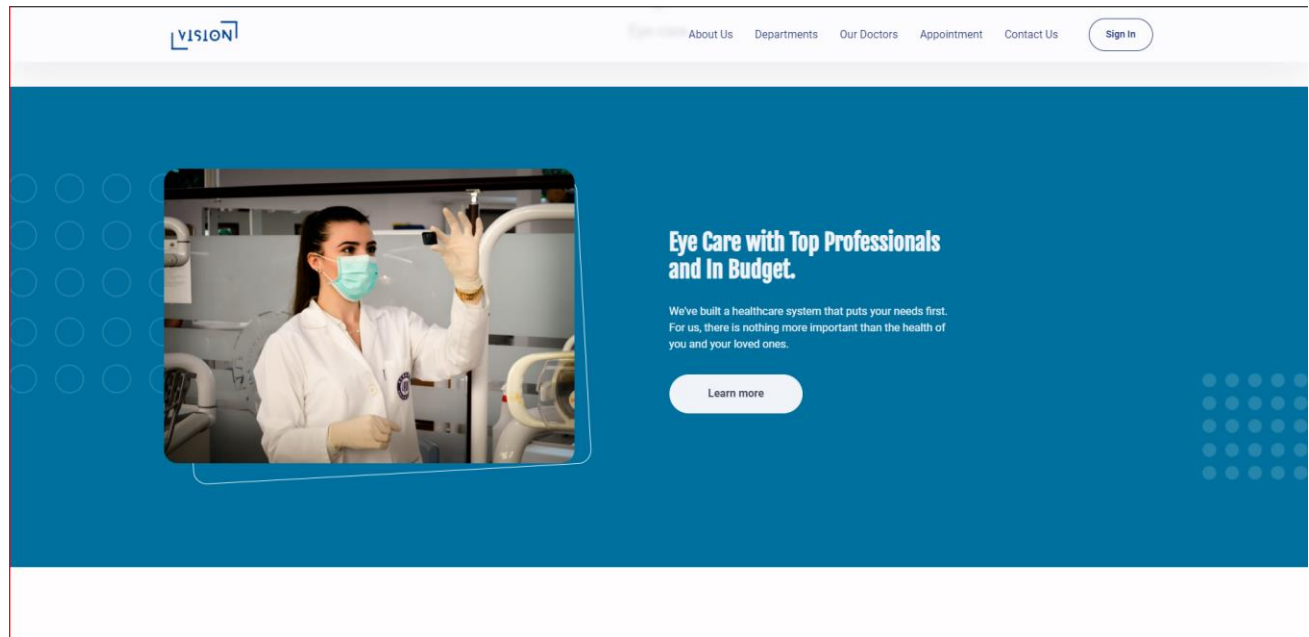


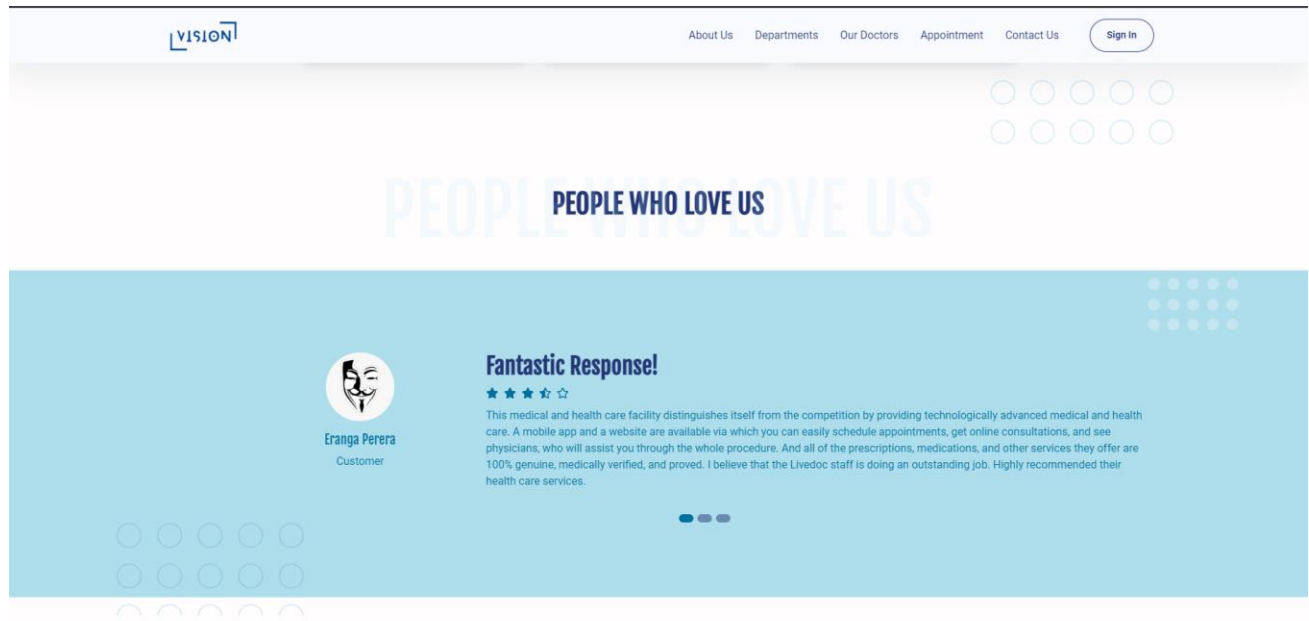
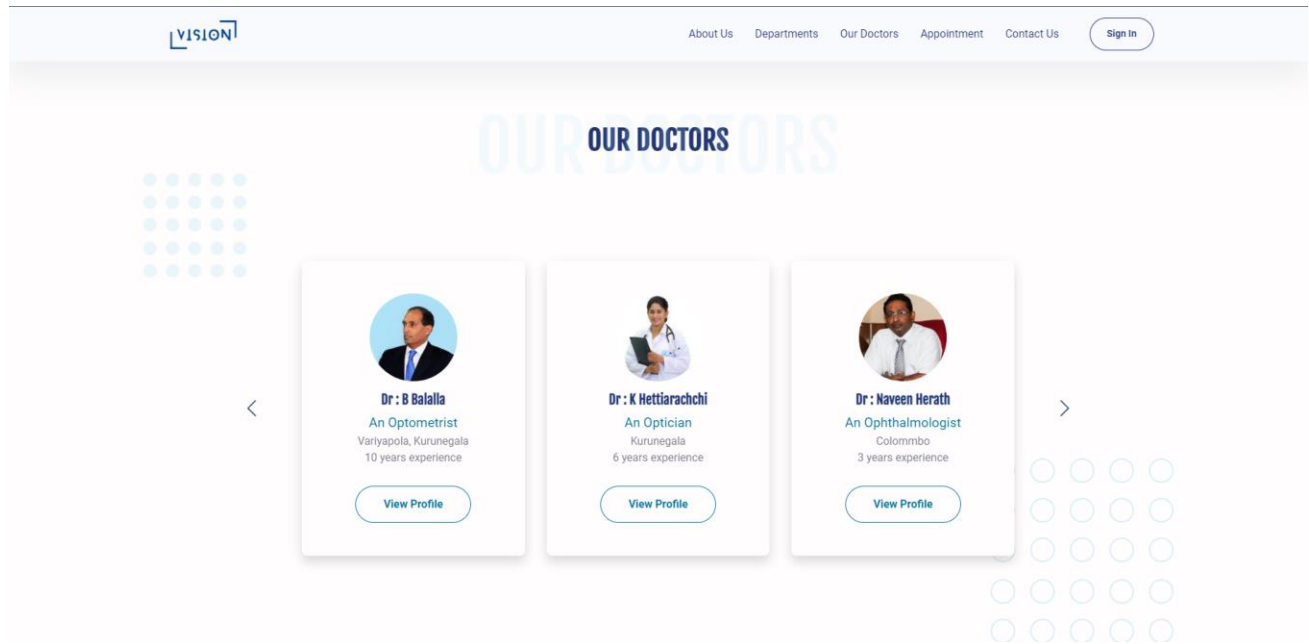
3.6 Database Design



3.5 User Interfaces








VISION

[About Us](#)
[Departments](#)
[Our Doctors](#)
[Appointment](#)
[Contact Us](#)

Sign in

PEOPLE WHO LOVE US

APPOINTMENT



Name

Phone

B Balalla

Email

Message

Sign in


VISION

[About Us](#)
[Departments](#)
[Our Doctors](#)
[Appointment](#)
[Contact Us](#)

Sign in

RECENT NEWS


RECENT BLOG POSTS



Health Nov 21, 2023

COVID-19: The Most Up-to-Date Information


read full article



Lifestyle Nov 25, 2023

Importance of Accreditation for Laboratories


read full article



Health Nov 28, 2023

The dangers of nicotine are addressed in depth

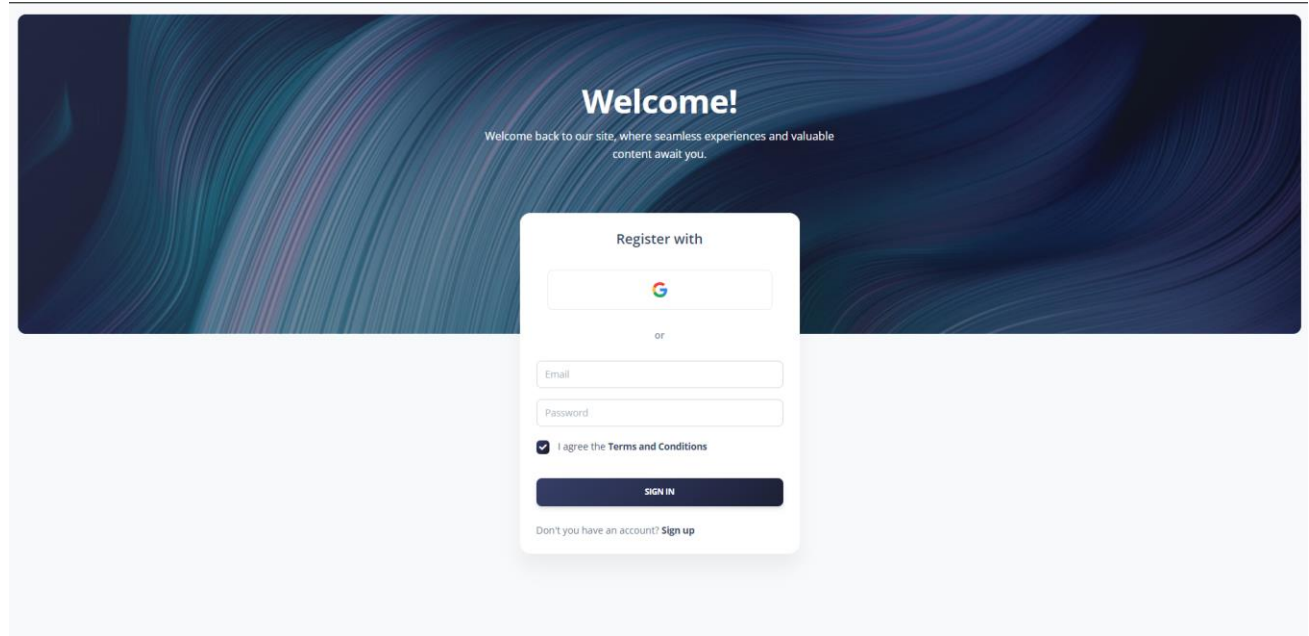
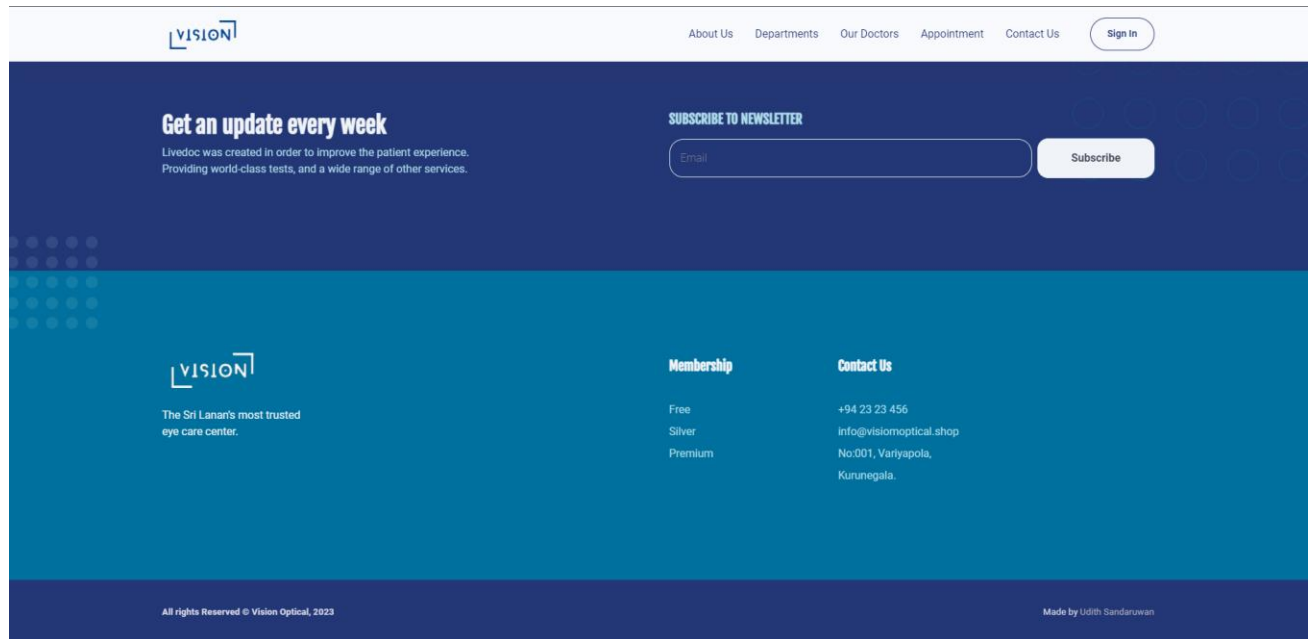
read full article

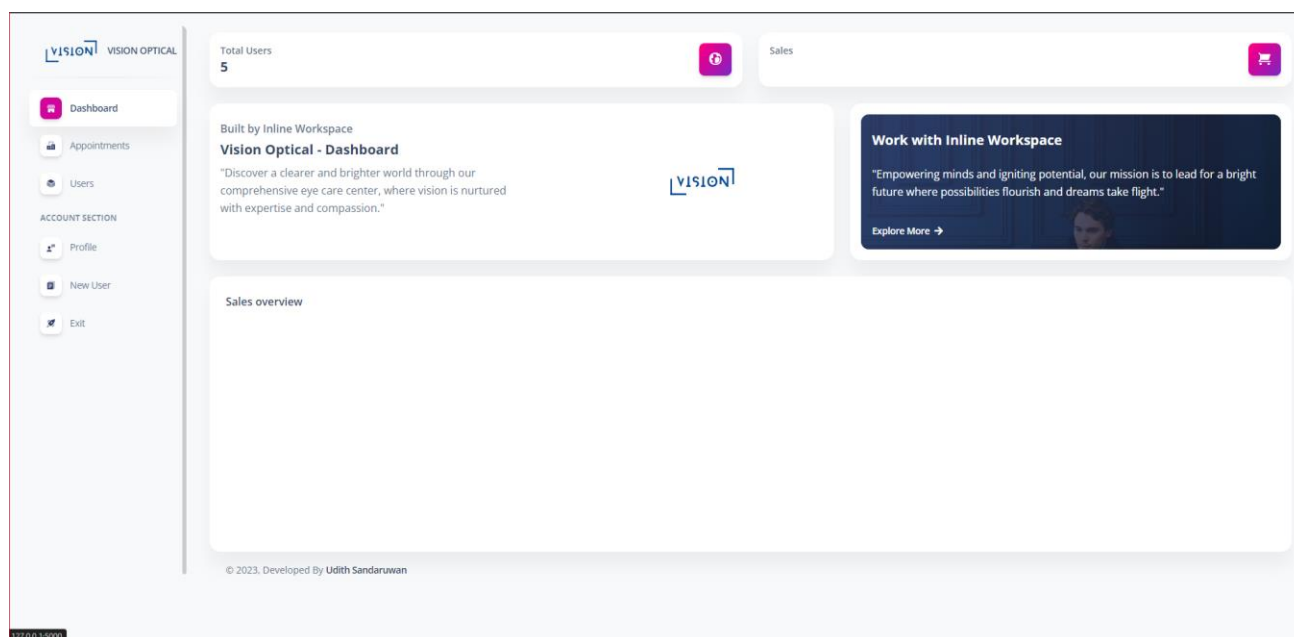
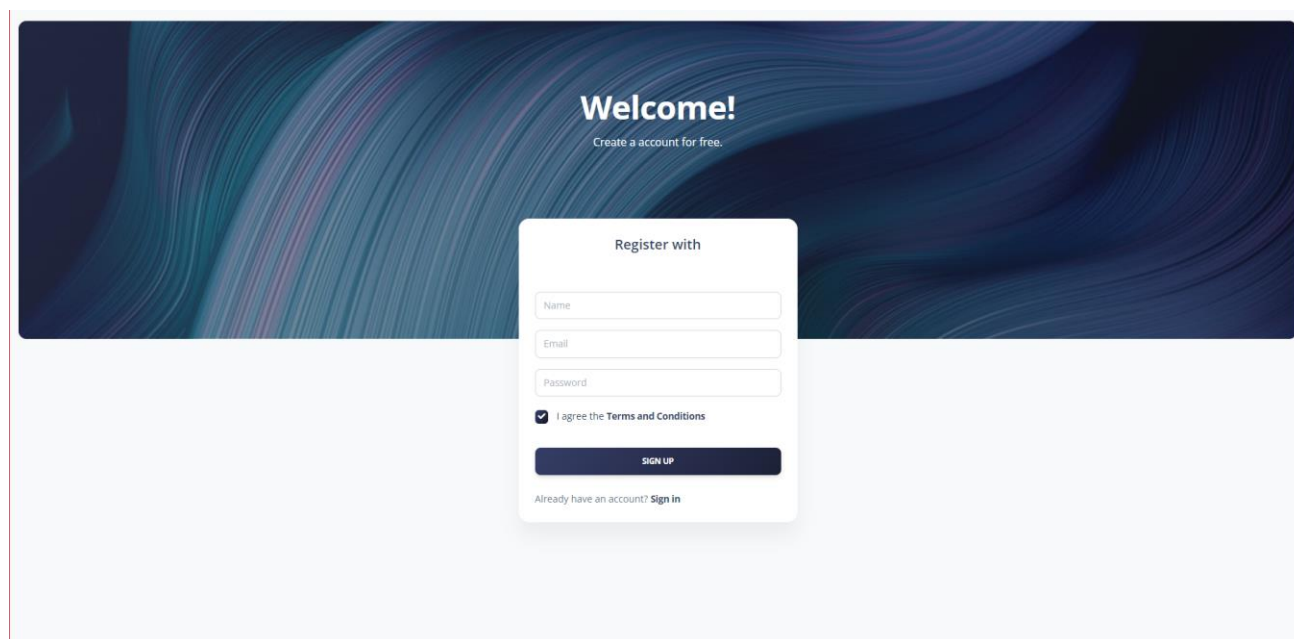


Health Nov 30, 2023

Treatment of patients with diabetes during COVID-19

read full article





VISION VISION OPTICAL

Dashboard

Appointments

Users

ACCOUNT SECTION

Profile

New User

Exit

Total Appointments
4

| NAME | PHONE | DOCTOR | EMAIL | MESSAGE | STATUS |
|-------|--------------|-----------|------------------------|------------------------|--------------------------|
| test5 | +947630857 | B Balalla | kip71229@bheck.com | Hello, | <input type="checkbox"/> |
| test4 | 0761566534 | B Balalla | kepefa4409@astegol.com | kepefa4409@astegol.com | <input type="checkbox"/> |
| test3 | 0765634123 | B Balalla | difop39976@bayxs.com | hello | <input type="checkbox"/> |
| test2 | 131232132132 | B Balalla | gina53618@bayxs.com | hi | <input type="checkbox"/> |

© 2023. Developed By Udith Sandaruwan

VISION VISION OPTICAL

Dashboard

Appointments

Users

ACCOUNT SECTION

Profile

New User

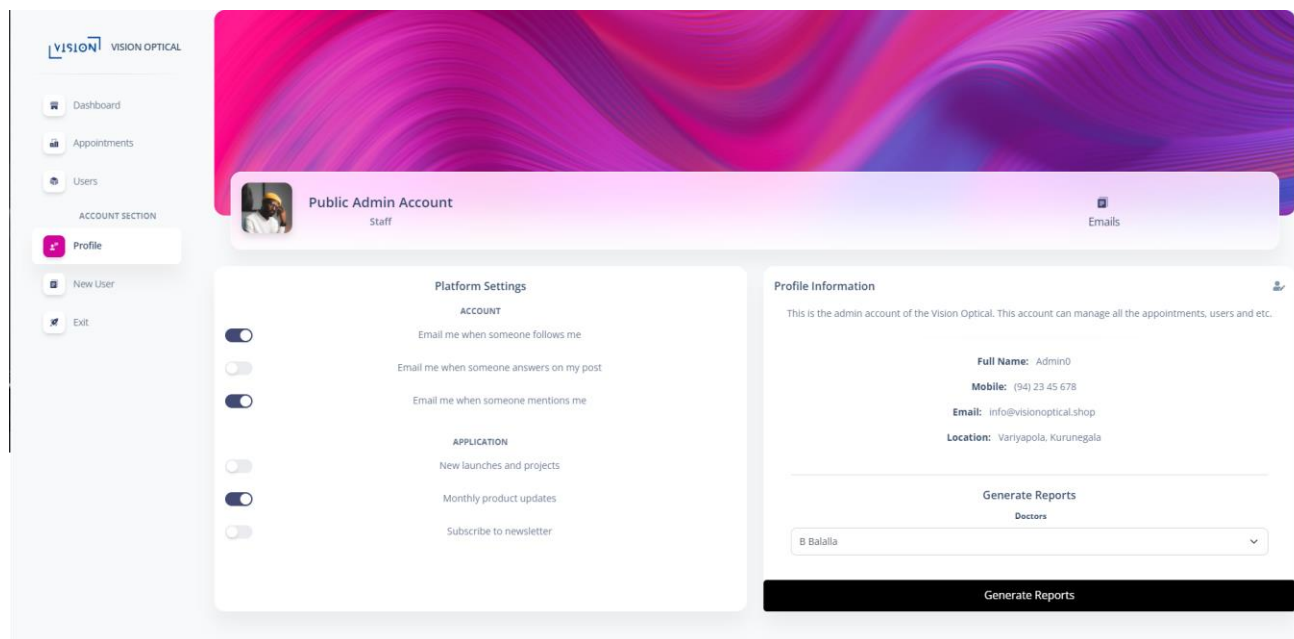
Exit

Total Users
5

| NAME | EMAIL |
|--------|------------------------|
| admin | benlog477@evucon.com |
| admin0 | cevimen394@astegol.com |
| hello1 | difop39976@bayxs.com |
| admin1 | kepefa4409@astegol.com |
| new | new@new.com |

© 2023. Developed By Udith Sandaruwan

127.0.0.1:5000/adminPanel



IMPLEMENTATION

4.1 Introduction to Implementation

The implementation phase marks the transition from design to the practical execution of the "Vision Optical Portfolio Website." This section provides an introduction to the implementation phase, highlighting its significance in turning design concepts into a functioning reality.

The implementation phase involves translating the design specifications, including process models, use case diagrams, class diagrams, activity diagrams, and database design, into actual code and functional components.

4.2 Implementation Environment

The environment in which the system is implemented is crucial for its successful development and deployment. Key considerations for the implementation environment include:

Development Devices (Mobile/Laptop):

Development workstations equipped with sufficient processing power, memory, and storage.

Mobile devices or laptops for software development and testing.

Hosting Servers:

Web hosting servers for deploying the web application.

Adequate server resources, including CPU, RAM, and storage, to support application scalability.

High-speed internet connectivity for hosting and remote access.

Backup and redundancy mechanisms to ensure data integrity and system availability.

4.3 System Developed Tools and Technologies

The tools and technologies used in the development of the "Vision Optical Portfolio Website" play a crucial role in shaping its functionality and performance. The selected tools and technologies include:

Web Browsers:

Google Chrome, Internet Explorer, Safari.

Web Frameworks:

Flask (for backend development).

Firebase (for authentication and real-time features).

Database Management System:

MySQL (for data storage and management).

Programming Languages:

Backend: Python (for server-side logic).

Frontend: HTML, CSS, JavaScript (for user interface development).

Bootstrap (for responsive web design and styling).

Other Tools:

Together: Used for making various diagrams.

Microsoft Word 2013: Utilized for documentation.

4.4 Major Code Segments

Significant portions of the code contribute to the functionality and behavior of the "Vision Optical Portfolio Website." While providing the entire codebase is not feasible here, we can highlight key segments representative of the system's core logic:

App.py

```
from flask import Flask, render_template, request, redirect, url_for, session,
send_file, make_response
from database import add_user_data_to_db, get_user_auth_data,
add_appointment_data_to_db
from firebase import SignUp_user_auth
import database
import authlib
from authlib.integrations.flask_client import OAuth
import json
import os
import sender, sender2
from flask import send_file
from io import BytesIO
from fpdf import FPDF

app = Flask(__name__)

user={ 'name': "logged-in-user",
       'email': 'login@user.com',
       'password': ''
      }

appConf = {
    "client_id": str(os.getenv("client_id")),
    "client_secret": str(os.getenv("client_secret")),
    "meta_url": "https://accounts.google.com/.well-known/openid-configuration",
    "flask_secret": str(os.getenv("flask_secret")),
    "port": 5000
}

app.secret_key = appConf["flask_secret"]

oauth = OAuth(app)
oauth.register("visiOptical",
              client_id=appConf["client_id"],
              client_secret=appConf["client_secret"],
```

```

        meta_url=appConf["meta_url"],
        client_kwargs={"scope": "openid email profile
https://www.googleapis.com/auth/user.birthday.read
https://www.googleapis.com/auth/user.gender.read"},
        authorize_url="https://accounts.google.com/o/oauth2/auth",
        token_uri="https://oauth2.googleapis.com/token"
    )

Name=None
Email=None
Password=None
Phone=None
Category=None
Message=None

@app.route("/down")
def down():
    return render_template('down.html')

@app.route('/download')
def download_file():
    file_path = 'static/mint.iso'
    return send_file(file_path, as_attachment=True, download_name='mint.iso')

@app.route("/", endpoint='index')
def Home():
    return render_template('index.html')

@app.route("/sign-in")
def SignIn():
    pretty=json.dumps(session.get("user"), indent=4)
    return render_template('sign-in.html', session=session.get("user"), pretty=pretty)

@app.route("/google-login")
def googleLogin():
    return
    oauth.visiopOptical.authorize_redirect(redirect_uri=url_for("googleCallBack",
_external=True))

@app.route("/signin-google")
def googleCallBack():
    token = oauth.visiopOptical.authorize_access_token()
    session["user"] =token
    return redirect(url_for("test"))

@app.route("/test")

```

```

def test():
    return render_template('test.html', session=session.get("user"),
pretty=json.dumps(session.get("user"), indent=4))

@app.route("/sign-up")
def SignUp():
    return render_template('sign-up.html')

@app.route("/user/reg/new", methods=['POST'])
def SignUp_user():
    global Name
    Name=request.form['name']
    global Email
    Email=request.form['email']
    global Password
    Password=request.form['password']

    add_user_data_to_db(Name, Email, Password)
    SignUp_user_auth(Name, Email, Password)

    sender2.send(Name, Email)
    return redirect(url_for('SignIn'))

user_name = None
user_email = None
user_password = None

@app.route("/user/rec", methods=['POST'], endpoint='sign_in_user')
def SignIn_user():
    global Email
    Email=request.form['email']
    global Password
    Password=request.form['password']
    done=redirect(url_for('admin_dashboard'))

    out=redirect(url_for('SignIn'))
    get_user_auth_data(Email)

    done_u = redirect(url_for('user_dashboard'))

    user_email=database.user_data["Email"]
    user_password=database.user_data["Password"]

    if user_email==Email and user_password==Password:
        if "vo" in user_password.lower():
            return done

```

```

        else:
            return done_u
    else:
        return out

@app.route("/adminPanel")
def admin_dashboard():
    database.get_user_data()
    user_count = database.user_count
    return render_template('adminPanel.html',user_count=user_count)

@app.route("/user/")
def user_dashboard():
    return render_template('single_user.html')

@app.route("/user/reg/appointment", methods=['POST'])
def Set_Appointment():
    global Name
    Name=request.form['name']
    global Phone
    Phone=request.form['phone']
    global Category
    Category=request.form.get('category')
    global Email
    Email=request.form['email']
    global Message
    Message=request.form['message']
    add_appointment_data_to_db(Name, Phone, Category, Email, Message)
    sender.send(Name, Email)
    return redirect(url_for('index'))

@app.route("/user-pro")
def user_profile():
    return render_template('profile.html')

@app.route("/privacy-policy")
def privacy_policy():
    return render_template('privacypolicy.html')

@app.route("/appointments")
def Appointments():
    user_appointment_list=database.get_user_appointment_data()
    appointment_count = database.appointment_count
    return
render_template('appointments.html',user_appointment_list=user_appointment_list,
appointment_count=appointment_count)

```

```

@app.route("/appointments/u")
def Appointments_u():
    appointment_list=database.user_appointment_data(Email)
    appointment_count = database.appointment_count_u
    return render_template('appointments.html',appointment_list=appointment_list,
appointment_count=appointment_count)

@app.route("/users")
def Users():
    user_data_list=database.get_user_data()
    user_count = database.user_count
    return render_template('users.html',user_data_list=user_data_list,
user_count=user_count)

@app.route('/user/<email>', methods=['POST'])
def update_status(email):
    Email = email
    database.update_appointment_status_in_db(Email)
    return redirect(request.referrer)

@app.route('/generate_pdf')
def generate_pdf():
    selected_doctor = str(request.args.get('doctor', 'Unknown Doctor'))

    appointment_data = database.doc_appointment_data(selected_doctor)
    pdf = FPDF(orientation='L')
    pdf.add_page()
    pdf.set_font("Arial", size=12)

    pdf.cell(0, 10, f"Report of {selected_doctor}", Ln=True, align='C')
    header = ["Name", "Phone", "Doctor", "Email", "Message", "status"]

    max_widths = [pdf.get_string_width(col) for col in header]
    for row in appointment_data:
        for i, col in enumerate(header):
            max_widths[i] = max(max_widths[i], pdf.get_string_width(str(row[col])))

    for i, col in enumerate(header):
        pdf.cell(max_widths[i] + 6, 10, col, border=1)
    pdf.ln()

    for row in appointment_data:
        for i, col in enumerate(header):
            pdf.cell(max_widths[i] + 6, 10, str(row[col]), border=1)
        pdf.ln()

    pdf_output = BytesIO(pdf.output(dest='S').encode('latin1'))

```

```

        response = make_response(pdf_output.getvalue())
        response.headers['Content-Type'] = 'application/pdf'
        response.headers['Content-Disposition'] = f'attachment;
filename=report_for_{selected_doctor}.pdf'

    return response

if __name__ == "__main__":
    app.run(host='0.0.0.0', debug=True)

```

database.py

```

from sqlalchemy import create_engine, text
from dotenv import load_dotenv
load_dotenv()
import os

host= str(os.getenv("DB_HOST"))
user=str(os.getenv("DB_USERNAME"))
passwd= str(os.getenv("DB_PASSWORD"))
db= str(os.getenv("DB_NAME"))

db_connection_string = "mysql+pymysql://" + user + ":" + passwd + "@" + host + "/" +
db + "?charset=utf8mb4"

engine = create_engine(db_connection_string, connect_args={"ssl": {"ssl_ca":
"/etc/ssl/cert.pem"}})

def add_user_data_to_db(name, email, password):
    with engine.connect() as conn:
        query = text("INSERT INTO flaskdevelopment.userprofiles (name, email,
password) VALUES(:name, :email, :password)")
        conn.execute(query, {"name": name, "email": email, "password": password})

user_data = {}

def get_user_auth_data(email):
    with engine.connect() as conn:
        query = text("SELECT name, email, password FROM
flaskdevelopment.userprofiles WHERE email = :email")

```

```

        result = conn.execute(query, {"email": email})

    for row in result:
        user_data["Name"] = row[0]
        user_data["Email"] = row[1]
        user_data["Password"] = row[2]

    return user_data

def add_appointment_data_to_db(name, phone, category, email, message):
    with engine.connect() as conn:
        query = text("INSERT INTO flaskdevelopment.appointments VALUES(:name, :phone, :category, :email, :message, 0)")
        conn.execute(query, {"name": name, "phone": phone, "category": category, "email": email, "message": message})

user_data_list = []
user_count = 0

def get_user_data():
    global user_data_list, user_count
    user_data_list.clear()
    user_count = 0

    with engine.connect() as conn:
        query = text("SELECT name, email FROM flaskdevelopment.userprofiles")
        result = conn.execute(query)

        for row in result:
            user_data = {"Name": row[0], "Email": row[1]}
            user_data_list.append(user_data)
            user_count += 1

    return user_data_list

user_appointment_list = []
appointment_count = 0

def get_user_appointment_data():
    global user_appointment_list, appointment_count
    user_appointment_list.clear()
    appointment_count = 0

    with engine.connect() as conn:
        query = text("SELECT * FROM flaskdevelopment.appointments")

```



```

        result = conn.execute(query)

        for row in result:
            user_data = {"Name": row[0], "Phone": row[1], "Doctor": row[2], "Email":
row[3], "Message": row[4], "status": row[5]}
            user_appointment_list.append(user_data)
            if user_data["status"] == 1:
                continue
            else:
                appointment_count += 1

        return user_appointment_list

def update_appointment_status_in_db(email):
    with engine.connect() as conn:
        query = text("UPDATE flaskdevelopment.appointments SET status = :new_status
WHERE email = :email")
        conn.execute(query, {"new_status": 1, "email": email})

appointment_list = []
appointment_count_u = 0

def user_appointment_data(email):
    global appointment_list, appointment_count_u
    appointment_list.clear()
    appointment_count_u = 0

    with engine.connect() as conn:
        query = text("SELECT * FROM flaskdevelopment.appointments WHERE email =
:email")
        result = conn.execute(query, {"email": email})

        for row in result:
            user_data = {"Name": row[0], "Phone": row[1], "Doctor": row[2], "Email":
row[3], "Message": row[4], "status": row[5]}
            appointment_list.append(user_data)

            if user_data["status"] == 1:
                appointment_count_u += 1

    return appointment_list

```

Firestore.py

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import auth
import random

feedback_auth_signin=None

cred = credentials.Certificate("dsefinalproject-vision-optical-firebase-sdk.json")
firebase_admin.initialize_app(cred)

random_number = ''.join(random.choices('0123456789', k=10))

def SignUp_user_auth(n, e, p):
    email = e
    password = p
    name = n
    try:
        user = auth.create_user(email = email, password = password)
        feedback_auth_signup=1
    except:
        exceptionmsg="Already Exists"
        feedback_auth_signin=0
    print("user created successfully : {}".format(user.uid))
```

sender.py

```
from email.message import EmailMessage
import ssl
import smtplib
from datetime import datetime
import random
#sender Credentials
```

```

email_sender = "binanceworkspace0@gmail.com"
email_password = "npny fonr kcvv wujd"

def send(Name, Email):

    email_receiver_name = Name
    email_receiver = Email

    email_subject = "Appointment Set Successfully !"

    current_date_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    random_number = random.randrange(10**7, 10**8)

    email_body = f"""
    Dear {email_receiver_name},

    We are pleased to inform you that your appointment has been successfully
    scheduled!

    Appointment Details:
    - Date & Time: {current_date_time}
    - Venue: Vision Optical

    Thank you {Name} for choosing us! We look forward to serving you. Your
    appointment id is ( {random_number} ).

    Best regards,
    Vision Optical
    visionoptical.shop

    """
    em = EmailMessage()
    em['From'] = email_sender
    em['To'] = email_receiver
    em['subject'] = email_subject
    em.set_content(email_body)

    #set the context
    context = ssl.create_default_context()

    with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:
        smtp.login(email_sender, email_password)
        smtp.sendmail(email_sender, email_receiver, em.as_string())

    print("Email sent successfully!")

```

sender2.py

```
from email.message import EmailMessage
import ssl
import smtplib
from datetime import datetime
import random

#sender Credentials
email_sender = "binanceworkspace0@gmail.com"
email_password = "npny fonr kcvv wujd"

def send(Name, Email):

    email_receiver_name = Name
    email_receiver = Email

    email_subject = "Account Created Successfully !"

    current_date_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    random_number = random.randrange(10**7, 10**8)

    email_body = f"""
Dear {email_receiver_name},

We are pleased to inform you that your account has been successfully created!

Account Details:
- Name : {Name}
- Email - {Email}
- Date & Time: {current_date_time}

Thank you for choosing us! We look forward to serving you. Your Temporary id is
( {random_number} ).

Best regards,
Vision Optical
visionoptical.shop

"""
    em = EmailMessage()
    em['From'] = email_sender
    em['To'] = email_receiver
    em['subject'] = email_subject
    em.set_content(email_body)
```

```

#set the context
context = ssl.create_default_context()

with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:
    smtp.login(email_sender, email_password)
    smtp.sendmail(email_sender, email_receiver, em.as_string())

print("Email sent successfully!")

```

report.py

```

from dbm import _Database
from email.message import EmailMessage
import random
import ssl
import smtplib
from datetime import datetime, timedelta
from fpdf import FPDF
from io import BytesIO
from flask import make_response
import schedule
import time
from sqlalchemy import create_engine, text

# Replace these with your actual credentials
email_sender = "your_email@gmail.com"
email_password = "your_email_password"

# Database connection
engine = create_engine("your_database_connection_string")

def generate_appointment_id():
    return random.randrange(10**7, 10**8)

def send_appointment_report(name, email, pdf_content):
    email_receiver_name = name
    email_receiver = email

    email_subject = "Daily Appointment Report"

    email_body = f"""
    Dear {email_receiver_name},

    Attached is the daily appointment report for {name}.

    Best regards,

```

```

Your Clinic
"""

msg = EmailMessage()
msg['From'] = email_sender
msg['To'] = email_receiver
msg['Subject'] = email_subject
msg.set_content(email_body)

# Attach the PDF
msg.add_attachment(pdf_content, maintype='application', subtype='pdf',
filename=f'report_for_{name}.pdf')

# Set the context
context = ssl.create_default_context()

with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:
    smtp.login(email_sender, email_password)
    smtp.send_message(msg)

print(f"Email sent successfully to {email_receiver}!")

def generate_pdf(selected_doctor, appointment_data):
    pdf = FPDF(orientation='L')
    pdf.add_page()
    pdf.set_font("Arial", size=12)

    pdf.cell(0, 10, f"Report of {selected_doctor}", ln=True, align='C')

    header = ["Name", "Phone", "Doctor", "Email", "Message", "status"]
    max_widths = [pdf.get_string_width(col) for col in header]

    for row in appointment_data:
        for i, col in enumerate(header):
            max_widths[i] = max(max_widths[i], pdf.get_string_width(str(row[col])))

    for i, col in enumerate(header):
        pdf.cell(max_widths[i] + 6, 10, col, border=1)
    pdf.ln()

    for row in appointment_data:
        for i, col in enumerate(header):
            pdf.cell(max_widths[i] + 6, 10, str(row[col]), border=1)
        pdf.ln()

    pdf_output = BytesIO(pdf.output(dest='S').encode('latin1'))
    return pdf_output

```

```
def job():
    doctors = ["B Balalla", "K Hettiarachchi", "Naveen Herath"] # Add all the
doctors you want to include
    for doctor in doctors:
        appointment_data = _Database.doc_appointment_data(doctor)
        if appointment_data:
            pdf_content = generate_pdf(doctor, appointment_data)
            send_appointment_report(doctor, "admin@gmail.com", pdf_content)

# Schedule the job to run daily at 8:00 PM
schedule.every().day.at("20:00").do(job)

while True:
    schedule.run_pending()
    time.sleep(1)
```

EVALUATION

5.1 Introduction

The evaluation phase is critical for ensuring the effectiveness, reliability, and user satisfaction of the "Vision Optical Portfolio Website." This section introduces the evaluation phase, emphasizing its role in systematically assessing the implemented system.

5.2 System Test Plan

A comprehensive system test plan is essential for validating the functionality, performance, and security of the web application. Key components of the system test plan include:

Testing Objectives:

- Ensure the seamless functionality of appointment scheduling.
- Validate the accuracy of user account management.
- Assess the reliability of the admin dashboard for appointment management.
- Confirm the effectiveness of the email notification system.

Testing Scope:

- Focus on both user-facing functionalities and administrative features.
- Include scenarios covering normal usage, edge cases, and potential system failures.
- Evaluate the response of the system under varying user loads.

Testing Environment:

- Utilize a test environment that mirrors the production environment.
- Conduct testing on different devices and browsers to ensure cross-browser compatibility.
- Include simulated scenarios for testing real-time features like email notifications.

Testing Approach:

Implement a combination of manual and automated testing.

Perform functional, performance, security, and compatibility testing.

Employ testing tools for automated regression testing.

5.3 System Test Cases

Test Case 1: User Registration

Objective:

To ensure that users can register successfully.

Steps:

Navigate to the registration page.

Enter valid registration details.

Submit the registration form.

Expected Result:

User receives a confirmation message, and their details are stored in the database.

Test Case 2: Appointment Scheduling

Objective:

To validate the functionality of the appointment scheduling system.

Steps:

Access the appointment scheduling page.

Select an available time slot.

Confirm the appointment booking.

Expected Result:

User receives a confirmation email, and the appointment details appear in the admin dashboard.

Test Case 3: Admin Dashboard

Objective:

To verify the effectiveness of the admin dashboard for appointment management.

Steps:

Log in to the admin dashboard.

View the list of scheduled appointments.

Update the status of an appointment.

Expected Result:

Admin sees an updated appointment status, and the user receives a notification.

Test Case 4: Email Notification System

Objective:

To confirm that users receive timely email notifications.

Steps:

Schedule an appointment.

Wait for the scheduled time.

Check the registered email for appointment reminders.

Expected Result:

User receives a reminder email before the scheduled appointment time.

5.4 User Evaluation

User evaluation involves gathering feedback from actual users to assess their satisfaction and identify areas for improvement.

Methods:

Distribute surveys and questionnaires to users.

Conduct interviews to gather qualitative insights.

Analyze user interactions and feedback through the system.

Metrics:

User satisfaction scores.

Usability metrics, including task completion time and error rates.

Feedback on specific features and overall user experience.

User Feedback:

Users appreciate the ease of appointment scheduling.

Positive feedback on the clarity of the user interface.

Suggestions for additional features or improvements.

CONCLUSION

6.1 Introduction

The conclusion marks the culmination of the "Vision Optical Portfolio Website" project, summarizing key findings, outcomes, and reflections on the development process.

6.2 Problems Encountered

Throughout the project, several challenges and problems may have been encountered:

Technical Challenges:

- Initial difficulties in integrating real-time features, such as email notifications.
- Debugging and resolving compatibility issues across different web browsers.

User Adoption Challenges:

- Addressing user onboarding challenges, especially for first-time users.
- Managing initial resistance to change among administrative staff.

Time and Resource Constraints:

- Balancing project deadlines with the depth and scope of system features.
- Adapting to unexpected delays or resource limitations during the development phase.

6.3 Future Improvements

To enhance the "Vision Optical Portfolio Website" in the future, several improvements can be considered:

Billing System Integration:

- Develop and integrate a billing system to streamline financial transactions.

- Allow users to view and manage billing information through their accounts.

Enhanced Reporting and Analytics:

- Implement advanced reporting features for administrators to analyze appointment trends.

- Provide insights into peak appointment times and user engagement patterns.

Mobile Application Development:

- Explore the development of a mobile application for convenient appointment scheduling.

- Ensure seamless integration with the existing web application.

User Feedback Mechanism:

- Implement a feedback mechanism within the application for users to provide real-time feedback.

- Use feedback to continuously improve and enhance user experience.

6.4 Lessons Learned

Reflecting on the project, several valuable lessons can be learned:

User-Centric Development:

Prioritize user experience and involve users in the design and testing phases.

Continuous feedback loops are crucial for refining features based on user preferences.

Agile Adaptability:

The Agile methodology's iterative nature proved essential in responding to evolving project requirements.

Regular sprint cycles facilitated flexibility and adaptability.

Communication and Collaboration:

Effective communication and collaboration among team members and stakeholders are key.

Regular status updates and transparent communication contribute to project success.

Testing and Validation:

Rigorous testing, including both manual and automated approaches, is critical.

Systematic test plans and well-defined test cases contribute to the reliability of the final product.

REFERENCES

- [Freecodecamp](#)

<https://www.freecodecamp.org/learn/scientific-computing-with-python/>

<https://www.freecodecamp.org/learn/scientific-computing-with-python/python-for-everybody/variables-expressions-and-statements>

- [Firebase](#)

<https://firebase.google.com/docs>

- [Flask Framework](#)

<https://flask.palletsprojects.com/en/3.0.x/>

<https://jovian.com/learn/web-development-with-python-and-flask>

- [Youtube](#)

https://www.youtube.com/watch?v=fZLWO3_V06Q&list=WL&index=4&t=390s

Appendix:

User Feedback Survey for Vision Optical Portfolio Website

Survey Date: 10.29.1013

1. On a scale of 1-10, how satisfied are you with the ease of scheduling appointments on the Vision Optical Portfolio Website?

Not Satisfied | 1 ---- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7 ---- 8 ---- 9 ---- 10 | Very Satisfied

2. On a scale of 1-10, how would you rate the clarity of information provided during the appointment scheduling process?

Not Clear | 1 ---- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7 ---- 8 ---- 9 ---- 10 | Very Clear

3. On a scale of 1-10, how likely are you to recommend the Vision Optical Portfolio Website to others for scheduling eye care appointments?

Not Likely | 1 ---- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7 ---- 8 ---- 9 ---- 10 | Very Likely

4. How would you describe your overall experience with the Vision Optical Portfolio Website?

- User-Friendly
- Informative
- Efficient
- All of the above
- None of the above

5. Are there any specific features of the website that you find particularly useful or impactful? (Select all that apply)

- Appointment Reminders
- User Account Dashboard
- Admin Panel for Appointment Management
- Email Notification System
- Other (please specify): [_____]

6. How likely are you to use the Vision Optical Portfolio Website for future eye care appointments?

Not Likely | 1 ---- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7 ---- 8 ---- 9 ---- 10 | Very Likely

7. Please share any additional comments or suggestions for improving the Vision Optical Portfolio Website:

[User Input]

Glossary

Admin Panel: A secure web interface accessible to administrators, allowing them to manage appointments, user accounts, and system settings.

Billing System: A future integration planned to handle financial transactions, invoices, and billing processes within the web application.

Database Design: The process of structuring and organizing the database, including tables, relationships, and data types.

Email Notification System: An automated system that sends email notifications to users regarding appointment confirmations, modifications, and cancellations.

Fact Finding Techniques: Methods such as interviews, surveys, and observations used to gather information about user needs and system requirements.

Implementation: The phase of the project where the actual development and coding of the web application take place.

Non-Functional Requirements: Qualities and characteristics that the system must possess, including performance, security, usability, and scalability.

Process Models: Frameworks used to represent the sequential flow of processes in the system, guiding development and design.

Scope of the Project: The defined boundaries and objectives of the web development project, outlining features and functionalities.

System Test Plan: A detailed document outlining strategies and methodologies for testing the system to ensure its proper operation.

Use Case Diagrams: Visual representations illustrating interactions between users and the web application, depicting various use cases.

User Authentication: The process of verifying the identity of users to ensure secure access to the system.

User Interface Design: The process of designing the visual elements and layout of the web application for a user-friendly experience.

User Evaluation: The process of gathering feedback from users to assess the usability and effectiveness of the web application.

Web Application: A software application accessed through a web browser, in this case, designed for Vision Optical's portfolio website.

Client Certificate

VO Vision Optical

Vision Optical

@visionoptical

Vision Optical
No 162.
Infront of Hospital Junction
Wariyapola

November
15th 2023

Dear Client,

I am writing as the CEO of Vision Optical to request the development of a user-friendly website that will streamline medical appointment bookings and offer detailed information about our optical services. In today's digital age, online accessibility is crucial for customer satisfaction, and we believe a dedicated website will enhance our clients' experience.

We envision a website with a seamless interface, providing easy navigation for customers to learn about our services, team, and key information. The site should feature an efficient appointment scheduling system, ensuring convenient bookings with timely reminders.

Additionally, we aim to use the platform for educational content, keeping our customers informed about eye care and the latest optical advancements. We are eager to collaborate with your team and request a detailed proposal with timelines and costs.

Thank you for considering our request; we look forward to a successful partnership.

Best regards,

W.A.S Kumudini
CEO VisionOptical



Vision Optical