

Project Report

On

Hospital Management System

Submitted by:

Soham Bapusaheb Waghmare (2020BCS035)(A17)

Chaitanya Rajendra Talware (2020BCS008)(B04)

Aniket Krishna Sontakke (2020BCS024)(B12)

Under the guidance of:

Dr. Jaishri Waghmare

Mrs. Swapnaja Moralwar



Shri Guru Gobind Singhji Institute of Engineering and Technology

Vishnupuri, Nanded (Maharashtra State) INDIA PIN 431606

Government Aided Autonomous Institute DTE Code: 2020

NAAC Accredited institute GRADE B++, CGPA 2.91 (2020 -2025)

Vision Statement: Education of Human Power for Technological Excellence

Department of Computer of Computer Science and Engineering

CERTIFICATE

Certified that this is bonafide record pf project work titled
Hospital management system

Done By:

Soham B. Waghmare (2020BCS035) (A17)

Chaitanya R. Talware (2020BCS008) (B04)

Aniket K. Sontakke (2020BCS024) (B12)

IN PARTIAL FULFILMENT FOR THE REQUIREMENT OF THIRD YEAR,
PROJECT,

Mrs. Swapnaja Moralwar
(Project Guide)

Dr. Jaishri Waghmare
(Head of Department)

Acknowledgement

Apart from the efforts of team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, co-ordination, and combined efforts of several sources of knowledge.

We are eternally grateful to our teacher Mrs. Swapnaja Moralwar for her even willingness to give us valuable advice and direction under which we executed this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

Table of Content

SR. NO.	CHAPTER	Page No.
0.	ABSTRACT	05
0.	PROBLEM STATEMENT	05
1.	INTRODUCTION	06
	1.1 PURPOSE	06
	1.2 SCOPE	06
	1.3 DEFINATIONS, ACRONYMS & ABBRAVIATIONS	07
	1.4 OVERVIEW	06
	1.5 SOFTWARE TOOLS USED	08
	1.6 TECHNOLOGY STACK	11
2.	SYSTEM DESIGN	12
	2.1 Table design	12
	2.2 ER DIAGRAM	14
3.	PYTHON CODE	15
4.	SYSTEM IMPLEMENTATION	18
	4.1 SCREENSHOTS	18
	4.2 FLOWCHART	22
5.	CONCLUSION	23
6.	FUTURE SCOPE	24
7.	REFERENCES	25

Abstract

Our project Hospital Management system includes registration of patients, storing their details into the system, and booking their appointments with doctors. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. User can search availability of a doctor and the details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

It is having mainly two modules. One is at Administration Level and other one is of user i.e., of patients and doctors. The Application maintains authentication to access the application. Administrator task includes managing doctors' information, patient's information. To achieve this, a database was designed one for the patient and other for the doctors which the admin can access. The complaints which are given by user will be referred by authorities. The Patient modules include checking appointments, prescription. User can also pay doctor's Fee online.

Problem Statement

In this busy world we don't have the time to wait in infamously long hospital queues. The problem is, queuing at hospital is often managed manually by administrative staff, then take a token there and then wait for our turn then ask for the doctor and the most frustrating thing - we went there by traveling a long distance and then we come to know the doctor is on leave or the doctor can't take appointments.

HMS will help us overcome all these problems because now patients can book their appointments at home, they can check whether the doctor they want to meet is available or not. Doctors can also confirm or decline appointments, this helps both patient and the doctor because if the doctor declines' appointment, then patient will know this in advance and patient will visit hospital only when the doctor confirms' the appointment this will save time and money of the patient. Patients can also pay the doctor's consultant fee online to save their time.

HMS is essential for all healthcare establishments, be it hospitals, nursing homes, health clinics, rehabilitation centres, dispensaries, or clinics. The main goal is to computerize all the details regarding the patient and the hospital. The installation of this healthcare software results in improvement in administrative functions and hence better patient care, which is the prime focus of any healthcare unit.

Chapter 1.

Introduction

- 1.1 PURPOSE
- 1.2 SCOPE
- 1.3 DEFINITIONS, ACRONYMS, and ABBREVIATIONS
- 1.4 OVERVIEW
- 1.5 SOFTWARE TOOLS USED
- 1.6 TECHNOLOGY STACK

1.1 PURPOSE

This software will help the company to be more efficient in registration of their patients and manage appointments, records of patients. It enables doctors and admin to view and modify appointments schedules if required. The purpose of this project is to computerize all details regarding patient details and hospital details.

1.2 SCOPE

The system will be used as the application that serves hospitals, clinic, dispensaries, or other health institutions. The intention of the system is to increase the number of patients that can be treated and managed properly.

If the hospital management system is file based, management of the hospital must put much effort on securing the files. They can be easily damaged by fire, insects, and natural disasters. Also, could be misplaced by losing data and information.

1.3 DEFINITIONS, ACRONYMS, and ABBREVIATIONS

1. **Cardiologist** - treats heart disease.
2. **Pediatrician** - treats infants, toddlers, children, and teenagers.
3. **Plastic Surgeon** - restores, reconstructs, corrects, or improves in the shape and appearance of damaged body structures, especially the face.
4. **Psychiatrist** - treats patients with mental and emotional disorders.
5. **Ophthalmologist** - treats eye defects, injuries, and diseases.
6. **ENT**- Ear, Nose and Throat Specialist.
7. **Appt** – Appointment.
8. **Sign up** - Creating New User.
9. **Log in** - Logging in Existing User.
10. **Ph No** - Mobile number.

1.4 OVERVIEW

Our application contains two modules – the admin module and the user module. Our application will not only help the admin to preview the monthly and/or yearly data but it will also allow them to edit, add or update records. The software will also help the admin to monitor the transactions made by the patients and generate confirmations for the same. The admin will be able to manage and update information about doctors.

The user module can be accessed by both the doctors and the patients. The doctor can confirm and/or cancel appointments. The doctors can even add prescriptions for their patients using our application. The patients will be able to apply for the appointment and make transaction for the same and can even cancel appointments with the doctors. They can track details about the previous transactions made by them.

Advantages

- The system automates the manual procedure of managing hospital activities.
- Doctors can view their patients' treatment records and details easily.
- It even generates an instant bill.
- The system is convenient and flexible to be used.
- It saves their time, efforts, money and resources.

Disadvantages

- Requires large database.
- The admin has to manually keep updating the information by entering the details in the system.
- Need Internet connection.

1.5 SOFTWARE TOOLS USED

HTML-

HTML or Hyper Text Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example . The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments, and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

CSS-

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind

of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification.

JAVA SCRIPT-

JavaScript (JS) is a dynamic computer programming language. It is most used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from

the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative,

and functional programming styles. The application of JavaScript to use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language, but just-in-time compilation is now performed by recent (post-2012) browsers.

PHP-

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Pre-processor, a recursive backronym. PHP code is interpreted by a webserver with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly

into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used

in standalone graphical applications. PHP is free software released under the PHP License. PHP can be deployed on most web servers and as a standalone shell on almost every operating system and platform, free of charge.

MYSQL-

MySQL ("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety

of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation

MySQL is a popular choice of database for use in web applications and is a central component of the widely used LAMP open-source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open-source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality.

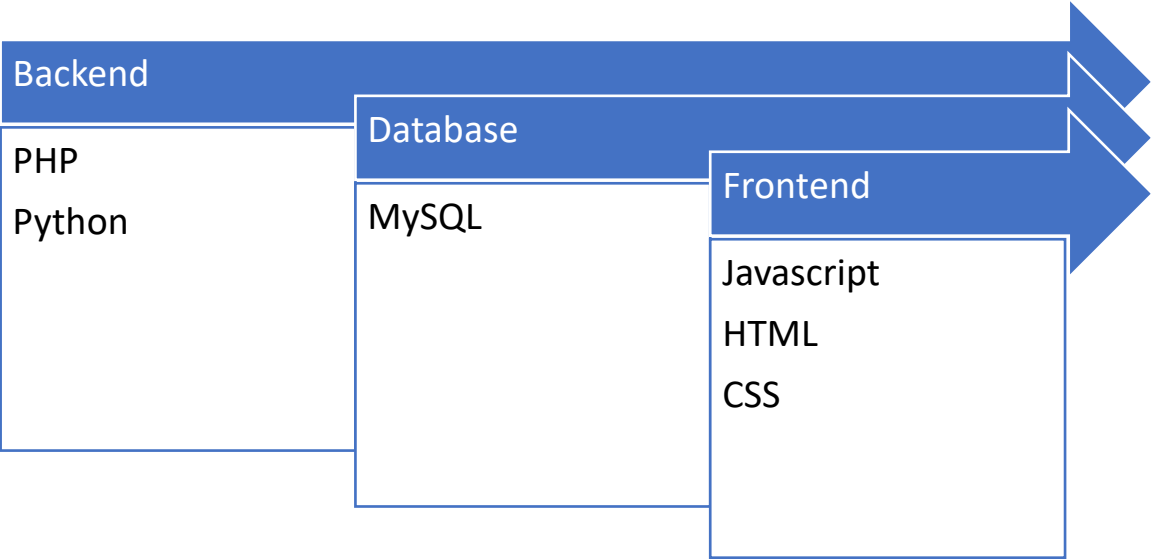
Python-

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

XAMPP-

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

1.6 TECHNOLOGY STACK:

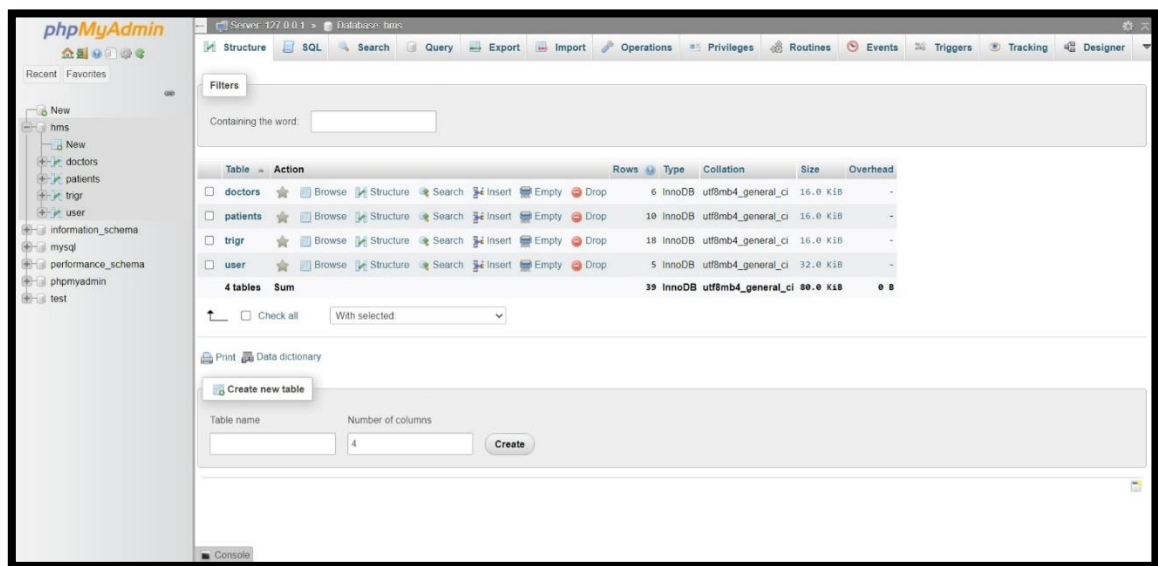


Chapter 2.

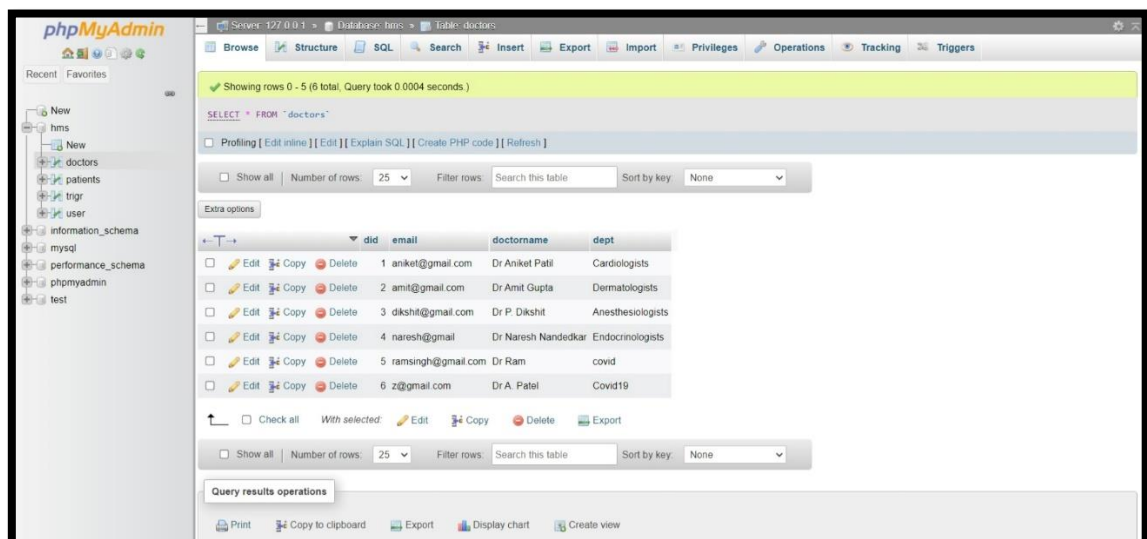
System Design

2.1 Table Design

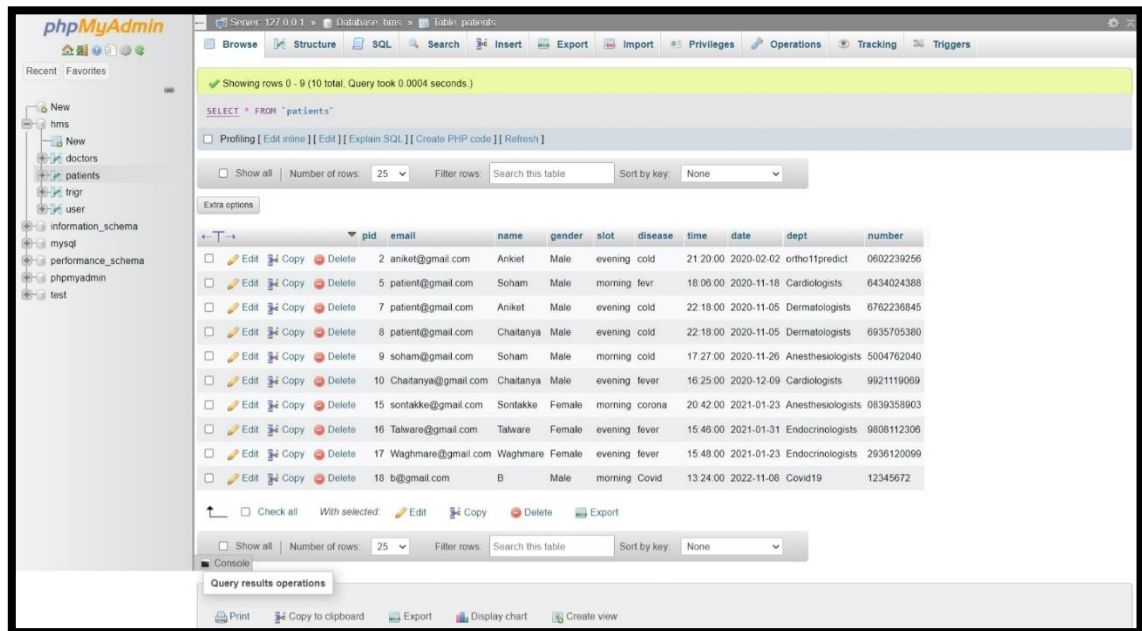
2.1.1 Tables in the database:



2.1.2 Table of doctors:



2.1.3 Table of patients:



Showing rows 0 - 9 (10 total, Query took 0.0004 seconds)

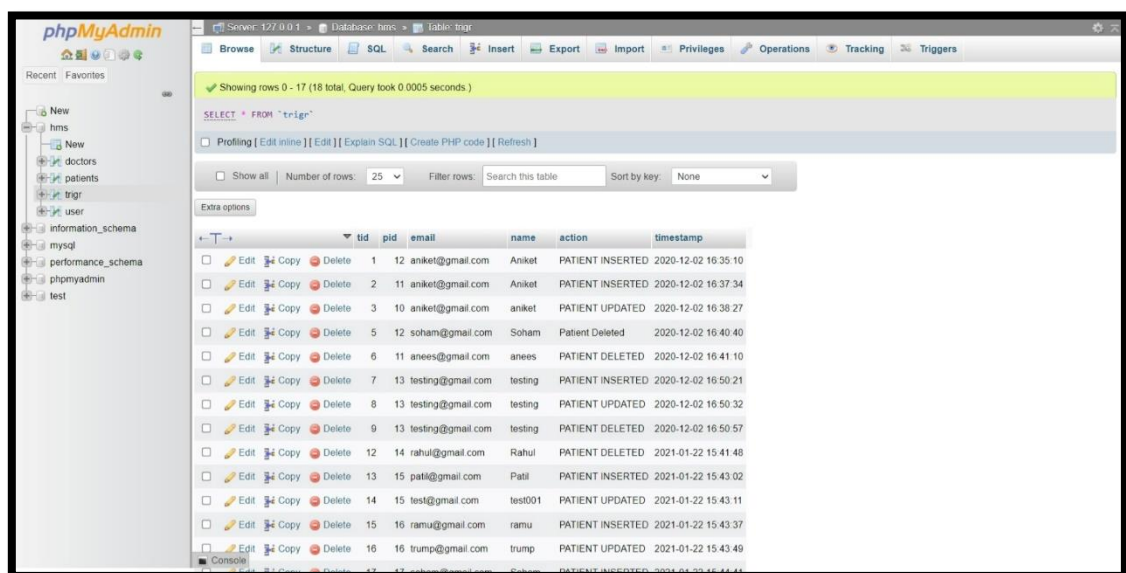
```
SELECT * FROM `patients`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	pid	email	name	gender	slot	disease	time	date	dept	number
<input type="checkbox"/>	2	aniket@gmail.com	Aniket	Male	evening	cold	21.20.00	2020-02-02	ortho11predict	0602239256
<input type="checkbox"/>	5	patient@gmail.com	Soham	Male	morning	fevr	18.06.00	2020-11-18	Cardiologists	6434024388
<input type="checkbox"/>	7	patient@gmail.com	Aniket	Male	evening	cold	22.18.00	2020-11-05	Dermatologists	6762236845
<input type="checkbox"/>	8	patient@gmail.com	Chaitanya	Male	evening	cold	22.18.00	2020-11-05	Dermatologists	6935705380
<input type="checkbox"/>	9	soham@gmail.com	Soham	Male	morning	cold	17.27.00	2020-11-26	Anesthesiologists	5004762040
<input type="checkbox"/>	10	Chaitanya@gmail.com	Chaitanya	Male	evening	fever	16.25.00	2020-12-09	Cardiologists	9921119069
<input type="checkbox"/>	15	sontakke@gmail.com	Sontakke	Female	morning	corona	20.42.00	2021-01-23	Anesthesiologists	0839358903
<input type="checkbox"/>	16	Talware@gmail.com	Talware	Female	evening	fever	15.46.00	2021-01-31	Endocrinologists	9808112306
<input type="checkbox"/>	17	Waghmare@gmail.com	Waghmare	Female	evening	fever	15.48.00	2021-01-23	Endocrinologists	2936120069
<input type="checkbox"/>	18	b@gmail.com	B	Male	morning	Covid	13.24.00	2022-11-08	Covid19	12345672

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

2.1.4 Table showing Record of all activities:



Showing rows 0 - 17 (18 total, Query took 0.0005 seconds)

```
SELECT * FROM `trig`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	tid	pid	email	name	action	timestamp
<input type="checkbox"/>	1	12	aniket@gmail.com	Aniket	PATIENT INSERTED	2020-12-02 16:35:10
<input type="checkbox"/>	2	11	aniket@gmail.com	Aniket	PATIENT INSERTED	2020-12-02 16:37:34
<input type="checkbox"/>	3	10	aniket@gmail.com	aniket	PATIENT UPDATED	2020-12-02 16:38:27
<input type="checkbox"/>	5	12	soham@gmail.com	Soham	Patient Deleted	2020-12-02 16:40:40
<input type="checkbox"/>	6	11	anees@gmail.com	anees	PATIENT DELETED	2020-12-02 16:41:10
<input type="checkbox"/>	7	13	testing@gmail.com	testing	PATIENT INSERTED	2020-12-02 16:50:21
<input type="checkbox"/>	8	13	testing@gmail.com	testing	PATIENT UPDATED	2020-12-02 16:50:32
<input type="checkbox"/>	9	13	testing@gmail.com	testing	PATIENT DELETED	2020-12-02 16:50:57
<input type="checkbox"/>	12	14	rahul@gmail.com	Rahul	PATIENT DELETED	2021-01-22 15:41:48
<input type="checkbox"/>	13	15	patil@gmail.com	Patil	PATIENT INSERTED	2021-01-22 15:43:02
<input type="checkbox"/>	14	15	test@gmail.com	test001	PATIENT UPDATED	2021-01-22 15:43:11
<input type="checkbox"/>	15	16	ramu@gmail.com	ramu	PATIENT INSERTED	2021-01-22 15:43:37
<input type="checkbox"/>	16	16	trump@gmail.com	trump	PATIENT UPDATED	2021-01-22 15:43:49

2.1.5 Table showing users in the system:

Server: 127.0.0.1 - Database: hms - Table: user

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds)

SELECT * FROM `user`

Extra options

		id	username	usertype	email	password
<input type="checkbox"/>	Edit	13	Aniket	Doctor	aniket@gmail.com	pbkdf2:sha256:1500005xAKZCJ0S4:7a7e704708f6659d7...
<input type="checkbox"/>	Edit	14	Soham	Patient	soham@gmail.com	pbkdf2:sha256:1500005Yf51iDC\$028c7B1a536ed9d4771...
<input type="checkbox"/>	Edit	15	Soham	Patient	chatanya@gmail.com	pbkdf2:sha256:15000058eSHerKVSa8b27379ce9b2499d4ca...
<input type="checkbox"/>	Edit	16	a	Doctor	a@gmail.com	pbkdf2:sha256:2600005QJ0wq2tSB6UYQPE\$713db91ed48...
<input type="checkbox"/>	Edit	17	b	Patient	b@gmail.com	pbkdf2:sha256:26000058nyCHN\$NvavWZC\$C0275bfb5901...

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

Console

2.2 ER Diagram

Chapter 3.

Python Code

```
1 from flask import Flask, render_template, request, session, redirect, url_for, flash
2 from flask_sqlalchemy import SQLAlchemy
3 from flask_login import UserMixin
4 from werkzeug.security import generate_password_hash, check_password_hash
5 from flask_login import login_user, logout_user, login_manager, LoginManager
6 from flask_login import login_required, current_user
7 from flask_mail import Mail
8 import json
9
10 with open('config.json', 'r') as c:
11     params = json.load(c)["params"]
12
13 # MY db connection
14 local_server = True
15 app = Flask(__name__)
16 app.secret_key = 'chaitanya'
17
18
19 # this is for getting unique user access
20 login_manager = LoginManager(app)
21 login_manager.login_view = 'login'
22
23 # SMTP MAIL SERVER SETTINGS
24
25 # app.config.update(
26 #     MAIL_SERVER='smtp.gmail.com',
27 #     MAIL_PORT='465',
28 #     MAIL_USE_SSL=True,
29 #     MAIL_USERNAME=params['gmail-user'],
30 #     MAIL_PASSWORD=params['gmail-password']
31 # )
32 # mail = Mail(app)
33
34
35 @login_manager.user_loader
36 def load_user(user_id):
37     return User.query.get(int(user_id))
38
39
40 # app.config['SQLALCHEMY_DATABASE_URL']='mysql://username:password@localhost/databas_table_name'
41 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/hms'
42 db = SQLAlchemy(app)
43
44
45 # here we will create db models that is tables
46 class Test(db.Model):
47     id = db.Column(db.Integer, primary_key=True)
48     name = db.Column(db.String(100))
49     email = db.Column(db.String(100))
50
51
52 class User(UserMixin, db.Model):
53     id = db.Column(db.Integer, primary_key=True)
54     username = db.Column(db.String(50))
55     usertype = db.Column(db.String(50))
56     email = db.Column(db.String(50), unique=True)
57     password = db.Column(db.String(1000))
58
```

```

59
60 class Patients(db.Model):
61     pid = db.Column(db.Integer, primary_key=True)
62     email = db.Column(db.String(50))
63     name = db.Column(db.String(50))
64     gender = db.Column(db.String(50))
65     slot = db.Column(db.String(50))
66     disease = db.Column(db.String(50))
67     time = db.Column(db.String(50), nullable=False)
68     date = db.Column(db.String(50), nullable=False)
69     dept = db.Column(db.String(50))
70     number = db.Column(db.String(50))
71
72
73 class Doctors(db.Model):
74     did = db.Column(db.Integer, primary_key=True)
75     email = db.Column(db.String(50))
76     doctorname = db.Column(db.String(50))
77     dept = db.Column(db.String(50))
78
79
80 class Trigr(db.Model):
81     tid = db.Column(db.Integer, primary_key=True)
82     pid = db.Column(db.Integer)
83     email = db.Column(db.String(50))
84     name = db.Column(db.String(50))
85     action = db.Column(db.String(50))
86     timestamp = db.Column(db.String(50))
87
88
89 # here we will pass endpoints and run the fuction
90 @app.route('/')
91 def index():
92     a = params['gmail-user']
93     return render_template('index.html')
94
95
96 @app.route('/doctors', methods=['POST', 'GET'])
97 def doctors():
98
99     if request.method == "POST":
100
101         email = request.form.get('email')
102         doctorname = request.form.get('doctorname')
103         dept = request.form.get('dept')
104
105         query = db.engine.execute(
106             f"INSERT INTO 'doctors' ('email','doctorname','dept') VALUES ('{email}','{doctorname}','{dept}')"
107         )
108         flash("Information is Stored", "primary")
109
110         return render_template('doctor.html')
111
112 @app.route('/patients', methods=['POST', 'GET'])
113 @login_required
114 def patient():
115     doct = db.engine.execute("SELECT * FROM 'doctors'")
116
117     if request.method == "POST":
118         email = request.form.get('email')
119         name = request.form.get('name')
120         gender = request.form.get('gender')
121         slot = request.form.get('slot')
122         disease = request.form.get('disease')
123         time = request.form.get('time')
124         date = request.form.get('date')
125         dept = request.form.get('dept')
126         number = request.form.get('number')
127         subject = "HOSPITAL MANAGEMENT SYSTEM"
128         query = db.engine.execute(
129             f"INSERT INTO 'patients' ('email','name','gender','slot','disease','time','date','dept','number') VALUES ('{email}','{name}','{gender}','{slot}','{disease}','{t"
130
131 # mail starts from here
132
133 # mail.send_message(subject, sender=params['gmail-user'], recipients=[email],body=f"YOUR BOOKING IS CONFIRMED THANKS FOR CHOOSING US \nYour Entered Details are :{name"
134
135 flash("Booking Confirmed", "info")
136
137 return render_template('patient.html', doct=doct)
138
139
140 @app.route('/bookings')
141 @login_required
142 def bookings():
143     em = current_user.email
144     if current_user.usertype == "Doctor":
145         query = db.engine.execute(f"SELECT * FROM 'patients'")
146         return render_template('booking.html', query=query)
147     else:
148         query = db.engine.execute(
149             f"SELECT * FROM 'patients' WHERE email='{em}'")
150         return render_template('booking.html', query=query)
151
152
153 @app.route("/edit/<string:pid>", methods=['POST', 'GET'])
154 @login_required
155 def edit(pid):
156     posts = Patients.query.filter_by(pid=pid).first()
157     if request.method == "POST":
158         email = request.form.get('email')
159         name = request.form.get('name')
160         gender = request.form.get('gender')
161         slot = request.form.get('slot')

```



```

162     disease = request.form.get('disease')
163     time = request.form.get('time')
164     date = request.form.get('date')
165     dept = request.form.get('dept')
166     number = request.form.get('number')
167     db.engine.execute(
168         f'UPDATE `patients` SET `email` = '{email}', `name` = '{name}', `gender` = '{gender}', `slot` = '{slot}', `disease` = '{disease}', `time` = '{time}', `date` = '{date}'
169     )
170     flash("Slot is Updated", "success")
171     return redirect('/bookings')
172
173
174
175 @app.route("/delete/<string:pid>", methods=['POST', 'GET'])
176 @login_required
177 def delete(pid):
178     db.engine.execute(f'DELETE FROM `patients` WHERE `patients`.`pid`={pid}')
179     flash("Slot Deleted Successful", "danger")
180     return redirect('/bookings')
181
182
183 @app.route('/signup', methods=['POST', 'GET'])
184 def signup():
185     if request.method == "POST":
186         username = request.form.get('username')
187         usertype = request.form.get('usertype')
188         email = request.form.get('email')
189         password = request.form.get('password')
190         user = User.query.filter_by(email=email).first()
191         if user:
192             flash("Email Already Exist", "warning")
193             return render_template('/signup.html')
194         encpassword = generate_password_hash(password)
195
196         new_user = db.engine.execute(
197             f'INSERT INTO `user` (`username`,`usertype`,`email`,`password`) VALUES ({username},{usertype},{email},{encpassword})')
198
199         # this is method 2 to save data in db
200         # newuser=User(username=username,email=email,password=encpassword)
201         # db.session.add(newuser)
202         # db.session.commit()
203         flash("Signup Succes Please Login", "success")
204         return render_template('login.html')
205
206     return render_template('signup.html')
207
208
209 @app.route('/login', methods=['POST', 'GET'])
210 def login():
211     if request.method == "POST":
212         email = request.form.get('email')
213         password = request.form.get('password')
214         user = User.query.filter_by(email=email).first()
215
216         if user and check_password_hash(user.password, password):
217             login_user(user)
218             flash("Login Success", "primary")
219             return redirect(url_for('index'))
220         else:
221             flash("invalid credentials", "danger")
222             return render_template('login.html')
223
224     return render_template('login.html')
225
226
227 @app.route('/logout')
228 @login_required
229 def logout():
230     logout_user()
231     flash("Logout Successful", "warning")
232     return redirect(url_for('login'))
233
234
235 @app.route('/test')
236 def test():
237     try:
238         Test.query.all()
239         return 'My database is Connected'
240     except:
241         return 'My db is not Connected'
242
243
244 @app.route('/details')
245 @login_required
246 def details():
247     # posts=Trigr.query.all()
248     posts = db.engine.execute("SELECT * FROM `trigr`")
249     return render_template('triggers.html', posts=posts)
250
251
252 @app.route('/search', methods=['POST', 'GET'])
253 @login_required
254 def search():
255     if request.method == "POST":
256         query = request.form.get('search')
257         dept = Doctors.query.filter_by(dept=query).first()
258         name = Doctors.query.filter_by(doctorname=query).first()
259         if name:
260             flash("Doctor is Available", "info")
261         else:
262             flash("Doctor is Not Available", "danger")
263
264     return render_template('index.html')
265
266
267
268 app.run(debug=True)

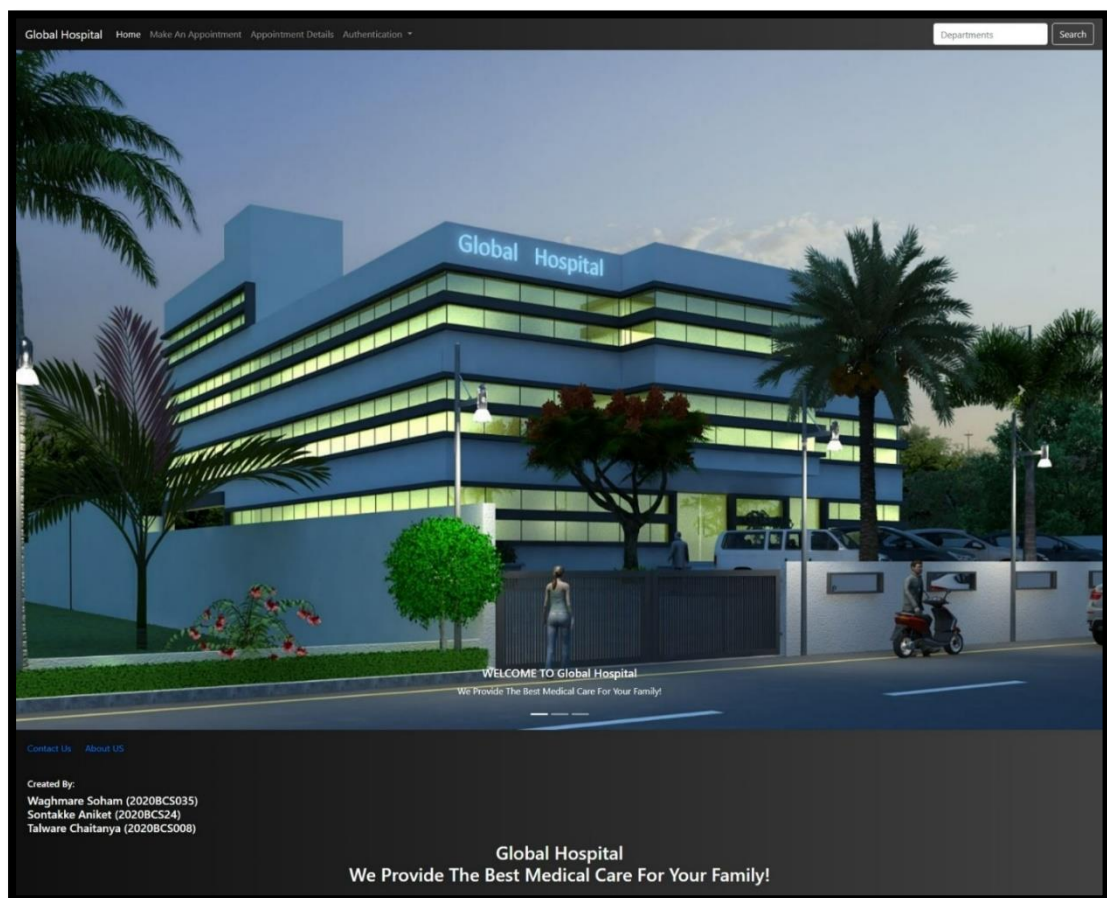
```

Chapter 4.

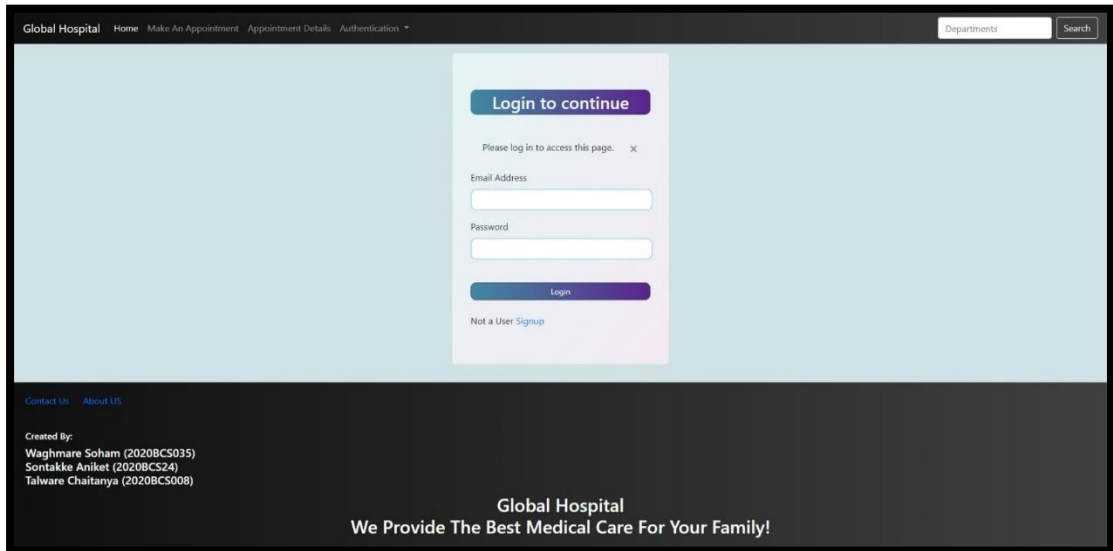
System Implementation

4.1 Screenshots

Home page



Login page



The screenshot shows the login page of the Global Hospital website. The header includes navigation links: Global Hospital, Home, Make An Appointment, Appointment Details, and Authentication. A search bar for departments is also present. The main content area features a central login form with a purple 'Login to continue' button. Below the button, a message states 'Please log in to access this page.' followed by a close icon. The form includes input fields for 'Email Address' and 'Password', and a 'Login' button. A link for 'Not a User? Signup' is located below the login button. The footer contains contact information, a list of creators (Waghmare Soham, Sontakke Aniket, Talware Chaitanya), and the hospital's slogan: 'Global Hospital We Provide The Best Medical Care For Your Family!'.

Global Hospital Home Make An Appointment Appointment Details Authentication * Departments Search

Login to continue

Please log in to access this page. ✕

Email Address

Password

Login

Not a User? Signup

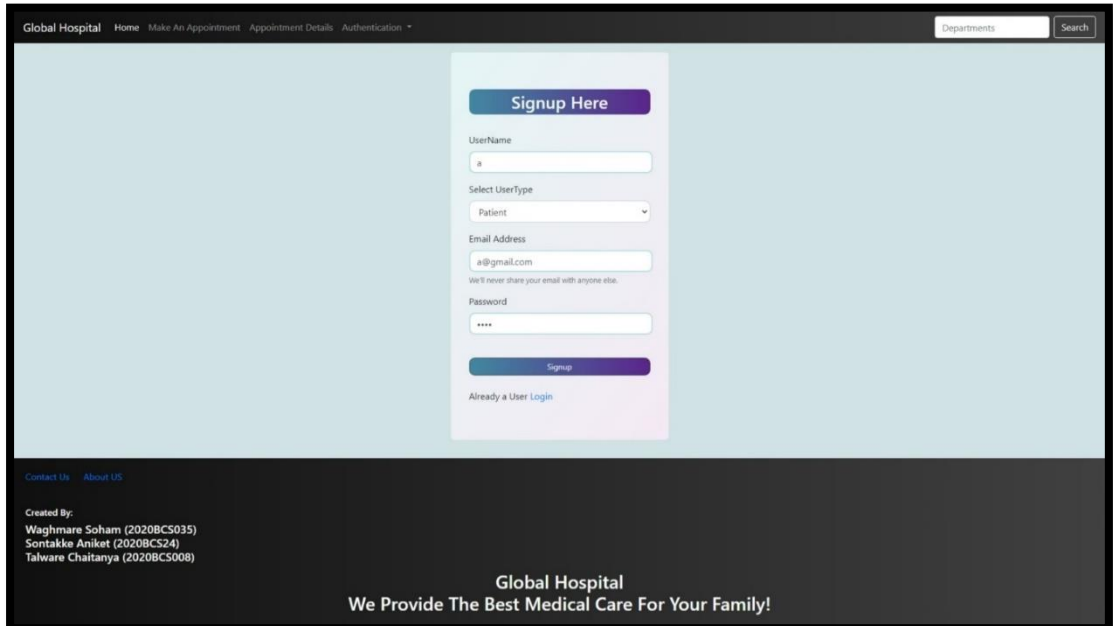
Contact Us About Us

Created By:
Waghmare Soham (2020BCS035)
Sontakke Aniket (2020BCS24)
Talware Chaitanya (2020BCS008)

Global Hospital
We Provide The Best Medical Care For Your Family!

Page for already signup users to login to the system.

Signup page



The screenshot shows the signup page of the Global Hospital website. The header is identical to the login page. The main content area features a central signup form with a purple 'Signup Here' button. The form includes input fields for 'UserName', 'Email Address', and 'Password', and a 'Select UserType' dropdown menu. A 'Signup' button is located below the form. A link for 'Already a User? Login' is located below the signup button. The footer is identical to the login page.

Global Hospital Home Make An Appointment Appointment Details Authentication * Departments Search

Signup Here

UserName

Select UserType

Email Address

Password

Signup

Already a User? Login

Contact Us About Us

Created By:
Waghmare Soham (2020BCS035)
Sontakke Aniket (2020BCS24)
Talware Chaitanya (2020BCS008)

Global Hospital
We Provide The Best Medical Care For Your Family!

Page for new users to signup. Both patients and Doctors can signup here.

Page for appointing a doctor

The screenshot shows the 'Appoint Doctor' page. At the top, there is a navigation bar with links: Global Hospital, Home, Doctors, Appointment Details, Patients Details, and Welcome a. A search bar for Departments is on the right. The main content area has a light blue background with a central white box titled 'Appoint Doctor'. Inside this box are three input fields: 'z@gmail.com', 'Dr A. Patel', and 'Covid19'. Below these is a black 'Appoint' button. At the bottom of the page, there is a dark grey footer with links 'Contact Us' and 'About US', a list of doctors (Waghmare Soham, Sontakke Aniket, Talware Chaitanya) with their IDs, and the text 'Global Hospital We Provide The Best Medical Care For Your Family!'.

More doctors can be appointed here. Can only be done by other doctors.

Page for booking an appointment

The screenshot shows the 'Appointment Booking' page. The navigation bar is similar to the previous page but includes 'Make An Appointment' and 'Appointment Details'. The main content area has a light blue background with a central dark grey box titled 'Appointment Booking'. On the left side of this box is a white panel with a caduceus icon, the text 'HOSPITAL DOCTORS', and a list of four doctors: Dr Aniket Patil, M.D, M.S.; Dr Amit Gupta, M.D.; Dr P. Dikshit, M.D.; and Dr Naresh Nandedkar, MBBS, MS, MCH. Below the list are links 'Contact Us' and 'About US'. On the right side of the dark grey box is a white form titled 'Appointment Booking'. The form contains input fields for 'b@gmail.com', 'Full name', 'Gender' (dropdown), 'Slot' (dropdown), a date field, a time field, 'Disease', 'Select Doctor Department' (dropdown), and 'Phone Number'. A purple 'Make an Appointment' button is at the bottom of the form.

Patients can book an appointment here.

Page showing appointments

Global Hospital Home Doctors Appointment Details Patients Details Welcome a *											
Departments										Search	
PID	Email	Name	Gender	Slot	Disease	Date	Time	D.Department	Contact	Edit	Delete
2	aniket@gmail.com	Aniket	Male	evening	cold	2020-02-02	21:20:00	ortho11predict	0602239256	Edit	Delete
5	patient@gmail.com	Soham	Male	morning	fevr	2020-11-18	18:06:00	Cardiologists	6434024388	Edit	Delete
7	patient@gmail.com	Aniket	Male	evening	cold	2020-11-05	22:18:00	Dermatologists	6762236845	Edit	Delete
8	patient@gmail.com	Chaitanya	Male	evening	cold	2020-11-05	22:18:00	Dermatologists	6935705380	Edit	Delete
9	soham@gmail.com	Soham	Male	morning	cold	2020-11-26	17:27:00	Anesthesiologists	5004762040	Edit	Delete
10	Chaitanya@gmail.com	Chaitanya	Male	evening	fever	2020-12-09	16:25:00	Cardiologists	9921119069	Edit	Delete
15	sontakke@gmail.com	Sontakke	Female	morning	corona	2021-01-23	20:42:00	Anesthesiologists	0839358903	Edit	Delete
16	Talware@gmail.com	Talware	Female	evening	fever	2021-01-31	15:46:00	Endocrinologists	9808112306	Edit	Delete
17	Waghmare@gmail.com	Waghmare	Female	evening	fever	2021-01-23	15:48:00	Endocrinologists	2936120099	Edit	Delete
Contact Us About Us											
Created By: Waghmare Soham (2020BCS035) Sontakke Aniket (2020BCS24) Talware Chaitanya (2020BCS008)											
Global Hospital We Provide The Best Medical Care For Your Family!											

Details of all the patient's appointments is showcased here.

Page showing all records of patient's activities

Global Hospital

Home

Doctors

Appointment Details

Patients Details

Welcome a *

Departments

Search

TID	PID	Email	Name	Action	Time
1	12	aniket@gmail.com	Aniket	PATIENT INSERTED	2020-12-02 16:35:10
2	11	aniket@gmail.com	Aniket	PATIENT INSERTED	2020-12-02 16:37:34
3	10	aniket@gmail.com	aniket	PATIENT UPDATED	2020-12-02 16:38:27
5	12	soham@gmail.com	Soham	Patient Deleted	2020-12-02 16:40:40
6	11	anees@gmail.com	anees	PATIENT DELETED	2020-12-02 16:41:10
7	13	testing@gmail.com	testing	PATIENT INSERTED	2020-12-02 16:50:21
8	13	testing@gmail.com	testing	PATIENT UPDATED	2020-12-02 16:50:32
9	13	testing@gmail.com	testing	PATIENT DELETED	2020-12-02 16:50:57
12	14	rahul@gmail.com	Rahul	PATIENT DELETED	2021-01-22 15:41:48
13	15	patil@gmail.com	Patil	PATIENT INSERTED	2021-01-22 15:43:02
14	15	test@gmail.com	test001	PATIENT UPDATED	2021-01-22 15:43:11
15	16	ramu@gmail.com	ramu	PATIENT INSERTED	2021-01-22 15:43:37
16	16	trump@gmail.com	trump	PATIENT UPDATED	2021-01-22 15:43:49
17	17	soham@gmail.com	Soham	PATIENT INSERTED	2021-01-22 15:44:41
18	17	talware@gmail.com	Talware	PATIENT UPDATED	2021-01-22 15:44:52
19	17	chaitanya@gmail.com	Chaitanya	PATIENT UPDATED	2021-01-22 15:44:59
20	18	1234@gmail.com	1234	PATIENT INSERTED	2022-11-10 12:26:26

Contact Us

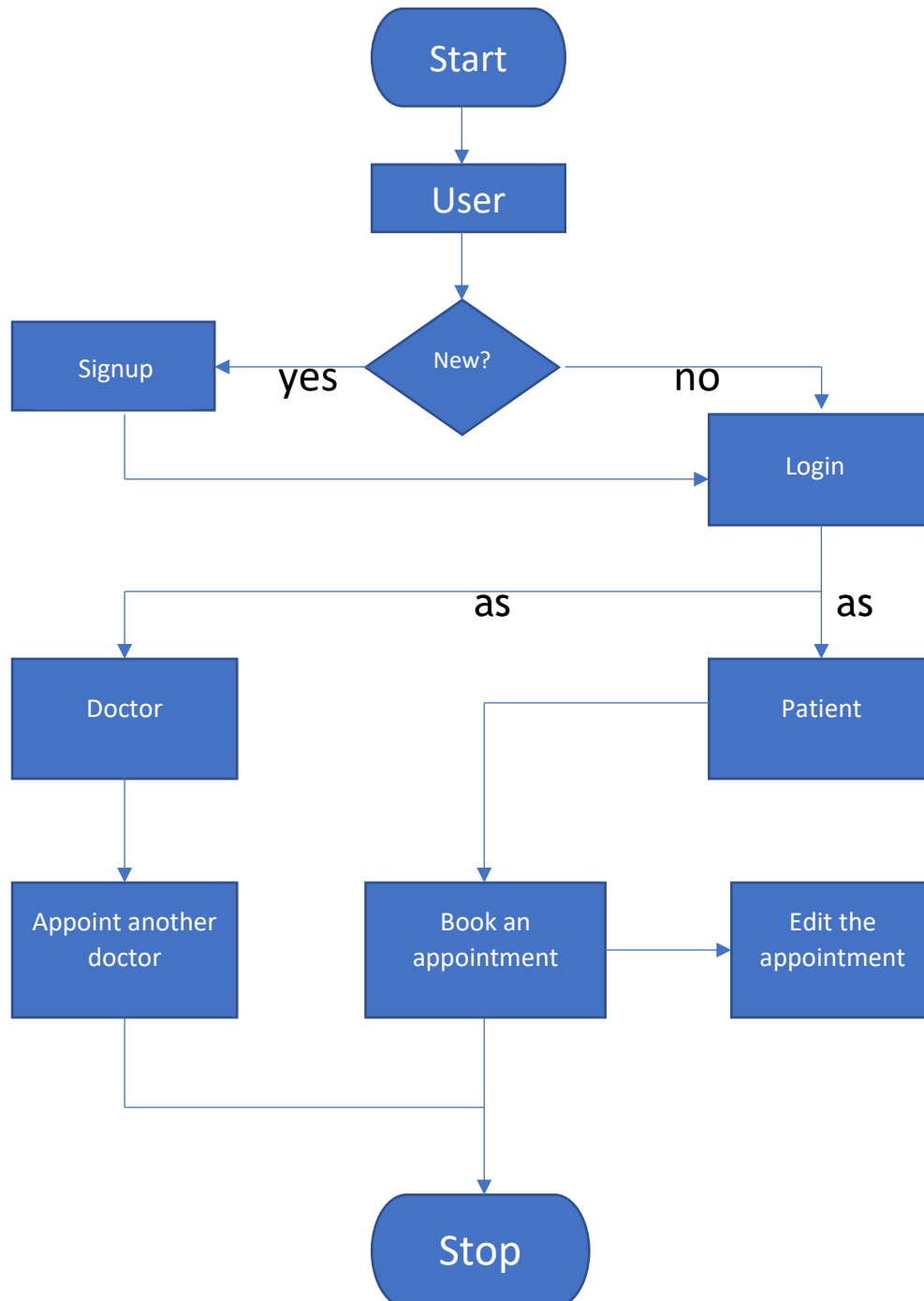
About Us

Created By:
Waghmare Soham (2020BCS035)
Sontakke Aniket (2020BCS24)
Talware Chaitanya (2020BCS008)

Global Hospital

We Provide The Best Medical Care For Your Family!

4.2 Flowchart:



Chapter 5.

Conclusion

Working on the project was an excellent experience. It helped us to understand the importance of planning, designing and implementation so far, we have learnt in our theory books. It helped us unleashing our creativity while working in a team. It also realized the importance of team working, communication as a part of this project.

The project was successfully completed after a lot of efforts and work hours. This project underwent number of compiling, debugging, removing errors, making it bug free, adding more facilities in Hospital Management System and interactivity making it more reliable and useful.

This project focused that scheduling a project and adhering to that schedule creates a hard sense of time- management. It has also let us know that co-operative teamwork always produces effective results.

The entire project has been developed and deployed as per the requirements stated by the user. It is found to be bug free as per the testing standards that are implemented.

Chapter 6.

Future Scope

Hospitals and healthcare centres have undergone a change for its betterment. The administrations of healthcare sector are opting IT solutions for the better management and patient care in their hospital campus. Have a look at some salient features of hospital management software.

Daily functions like patient registration, monitoring blood bank, managing admission and overall management of various departments can be easily performed with higher accuracy after the installation of hospital software.

The modules of hospital management software are user-friendly and easy to access. It has a common user-friendly interface having several modules. The officials can utilize these modules in their processes without any hassle and make the best possible use of hospital management system.

Since, every hospital has some or the other points of worth those vary in comparison with to its competitors. Hence, most of the IT companies give on-demand solutions or feature of customization. It further implicates that hospital information management software can be customized by specifying personal requirements of the campus.

The automated functions of online hospital software make productivity effective. This web-based IT solution has automated operations and permit officials to continue with their work in a swift manner. It further implicates that complete automation of the hospital software makes productivity easily obtainable. All in all, this enhances the infrastructure of hospital administration.

This tool is a comprehensive solution that integrates all the departments by creating a common platform. In brief, hospital management system has all the modules that serve purpose of all the departments of healthcare institute. In fact, these modules have been competitively designed to make all the operations simplified.

Chapter 7.

References

<https://www.karexpert.com/blogs/what-is-hospital-management-system/>

<https://mocdoc.in/blog/a-detailed-view-of-hospital-management-system-hms>

<https://1000projects.org/hospital-database-management-system-project-using-php-mysql.html>

GitHub link to the project:

<https://github.com/2020BCS008/Hospital-Management-System>

