

Hospital Management System

CSE-0318 Summer 2021

Rubyat Jesmin Shammi

Department of Computer Science and Engineering

State University of Bangladesh (SUB)

Dhaka, Bangladesh

rubyatshammiss@gmail.com

Abstract—project Hospital Management system includes registration of patients, storing their details into the system. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. 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 interface is very user-friendly.

Index Terms—

I. INTRODUCTION

Hospital management system is a computer system that helps manage the information related to health care and aids in the job completion of health care providers effectively. They manage the data related to all departments of healthcare.

II. LITERATURE REVIEW

Ahmed, E and El-Alem, M (2005). “On multi objective optimization in Hospital management system” and computation167.1, In: Knowledge-Based Systems, p. 105944. Alexander, Gordon J, Bruce G (1985). “Using linear and goal programming to immunize bond”. In: Journal of Banking and Finance9.1, pp. 35–54. Akbay, Mehmet Anil, Kalayci, Can B, and Polat, Olcay (2020). “A parallel variable neighborhood search

III. OUTLINE

Introduction
Requirement
Features
Advantages
Disadvantages
Conclusion

IV. REQUIREMENT

Language Used : Php
Database : My Sql
User Interface Design : Html , Css, Bootstrap Javascript
Web Browser : Google Chrome
Software : XAMPP Server

V. FEATURES

Login Systems (Patients,Admin,Doctor)
Book Appointment
Doctor Specialization

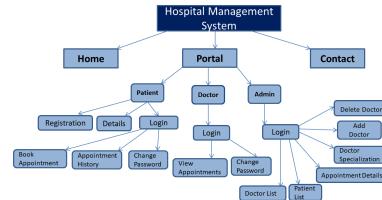


Fig. 1.

VI. CONCLUSION AND FUTURE WORK

Manage Patients, Doctors

When the hospital considers applying this approach, it is necessary to learn how to develop a hospital management system. Planning and setting priorities is one of the important stages. It outlines the functions and benefits that the medical institution expects to get at the final point.

ACKNOWLEDGMENT

I would like to thank my honourable **Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

REFERENCES

- [1] Pittet V, Burnand B, Yersin B , et al . Trends of pre-hospital emergency medical services activity over 10 years: A population-based registry analysis. *BMC Health Serv Res* 2014; 14: 380.
- [2] Erhardt L, Herlitz J, Bossaert L , et al . Task force on the management of chest pain. *Eur Heart J* 2002; 23: 1153–1176.
- [3] Nolan J P, Hazinski M F, Billi J E , et al . Part 1: Executive summary: 2010 International Consensus on Cardiopulmonary Resuscitation and Emergency Cardiovascular Care Science with Treatment Recommendations. *Resuscitation* 2010; 81(Suppl. 1): e1–e25.
- [4] Tubaro M, Vranckx P, Price S , et al . (eds). *The ESC textbook of intensive and acute cardiovascular care*. 2nd ed. The European Society of Cardiology, Sophia Antipolis, France, 2015.

VII. ER DIAGRAM

VIII. CODE

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 2.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 9.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 3.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 10.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 4.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 11.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 5.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 12.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 6.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 13.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 7.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 14.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 8.

```
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~#
```

Fig. 15.

```
git:(master) > curl -X POST -H "Content-Type: application/json" -d '{"id": "1", "name": "John Doe", "age": 30, "city": "New York"}' http://127.0.0.1:8000/api/users/
```

Fig. 16.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/1
```

Fig. 23.

```
git:(master) > curl -X PUT -H "Content-Type: application/json" -d '{"id": "1", "name": "Jane Doe", "age": 28, "city": "Los Angeles"}' http://127.0.0.1:8000/api/users/1
```

Fig. 17.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/1
```

Fig. 24.

```
git:(master) > curl -X DELETE http://127.0.0.1:8000/api/users/1
```

Fig. 18.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/1
```

Fig. 25.

```
git:(master) > curl -X POST -H "Content-Type: application/json" -d '{"id": "2", "name": "Jane Doe", "age": 28, "city": "Los Angeles"}' http://127.0.0.1:8000/api/users/
```

Fig. 19.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/2
```

Fig. 26.

```
git:(master) > curl -X PUT -H "Content-Type: application/json" -d '{"id": "2", "name": "John Doe", "age": 30, "city": "New York"}' http://127.0.0.1:8000/api/users/2
```

Fig. 20.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/2
```

Fig. 27.

```
git:(master) > curl -X DELETE http://127.0.0.1:8000/api/users/2
```

Fig. 21.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/2
```

Fig. 28.

```
git:(master) > curl -X POST -H "Content-Type: application/json" -d '{"id": "3", "name": "John Doe", "age": 30, "city": "New York"}' http://127.0.0.1:8000/api/users/
```

Fig. 22.

```
git:(master) > curl -X GET http://127.0.0.1:8000/api/users/3
```

Fig. 29.

```
git:(master) > ./script.sh
```

Fig. 30.

```
git:(master) > ./script.sh
```

Fig. 37.

```
git:(master) > ./script.sh
```

Fig. 31.

```
git:(master) > ./script.sh
```

Fig. 38.

```
git:(master) > ./script.sh
```

Fig. 32.

```
git:(master) > ./script.sh
```

Fig. 39.

```
git:(master) > ./script.sh
```

Fig. 33.

```
git:(master) > ./script.sh
```

Fig. 40.

```
git:(master) > ./script.sh
```

Fig. 34.

```
git:(master) > ./script.sh
```

Fig. 41.

```
git:(master) > ./script.sh
```

Fig. 35.

```
git:(master) > ./script.sh
```

Fig. 42.

```
git:(master) > ./script.sh
```

Fig. 36.

```
git:(master) > ./script.sh
```

Fig. 43.