

Programming Project Talk: **Hospital Management System** – An Old Concept

“There is no such thing as a new idea. We simply take a lot of old ideas and put them into a sort mental kaleidoscope.”

-Mark Twin

Programming has gone from a core aspect of Computer science to not just a basic but a high-tier skill for almost every technical profession. While being a programmer or a developer centers your profession around coding, many other fields also prefer people with programming skills over the rest.

In almost every job in the STEM (Science, Technology, Engineering, Math) field, being able to code gives you a distinct advantage over your peer group. Therefore, improving your skills in the digital realm has become a norm, especially for future engineers.

You can start with a simple programming language (I suggest Python or JS), and learn from books or online courses, and become proficient in that language by hard-work and self-study. However, like every other practical skill, practice plays the key role for the programmers. Programmers gain their skills and increase their mastery with every code written, and every challenge cleared, and are constantly in the learning process. So, you can say, “The best way to begin to **learn to code** is to simply start coding!” (So many ‘to’s 😊)

If you aspire to level up your skills, then you should take up projects, tackle more problems, and learn to utilize your knowledge in solving practical situations. You have to keep in mind that programming is no longer the one-dimensional idea of writing, running, and testing codes, although it is still a big section of it. You have to perform a lot of tasks in order to create and deploy your application. Again, there are many aspects to programming, like Data Science, Web Design, DevOps, and so on.

You must choose a project that not only fits your skill levels but is also sufficiently challenging for you to learn something from the end of it. While working on that project, you will be learning new methods, improving a specific skill, or finding a creative solution, challenging projects can help you grow as a programmer and will also look good on your portfolio.

Every developer will tell you coding can be very challenging, especially for beginners. In fact, a career in development can seem so limitless that you may be wondering where to begin. As you get started, other coders may stress the need for you to work on creative projects and build your portfolio - and they'd be right!

As a junior developer, your portfolio is your most valuable marketing piece. Potential employers will appreciate your proven track record of developing projects from end-to-end when considering you for a role or investing in you.

Improving your programming skills is a gradual process. Every project you complete will add something to your skillset and help you polish your skills. Platforms like GitHub allow you to contribute to open-source projects, build your portfolio, increase your knowledge

as a coder, and also learn from industry professionals. Such sites provide a favorable environment to foster your skillset and find existing issues that coders of any level (beginner, intermediate, self-taught expert, professional) can tackle.

Most of the time, when you will work on a project, you may find yourself abandoning it out of frustration or difficulty. The most important thing is to keep ongoing. When you gain experience developing challenging projects from inception to completion, you establish a strong work ethic that will benefit you throughout your career.

Here some tips to successfully complete a project:

- ✚ Learn a new technology.
- ✚ Link your newly gained knowledge with other technologies that you know.
- ✚ Research through different resources to find a project.
- ✚ Create the initial skeleton of that project.
- ✚ Add more methods and functionalities to your project.
- ✚ Learn throughout the project.
- ✚ Publish your project.
- ✚ Promote the project on multiple social network platforms.

Ok, now let's talk about a specific programming project. And we will discuss the Hospital Management System. Though it is an old topic, it is that kind of project in which you will find lots of opportunities to learn. So, we will be talking about what is HMS application, what is its functionalities, how to build an HMS application and I will also share my personal experience with you guys.

What is a Hospital Management System (HMS) application?

The hospital management system (HMS) is integrated software that handles different directions of clinic workflows. It manages the smooth healthcare performance along with administrative, medical, legal, and financial control. That is a cornerstone for the successful operation of the healthcare facility. In simple words, an HMS application can be a web-app or a desktop app which will provide enough functionalities to ensure a smooth workflow in the daily tasks of a medical institution.

Different functionalities that an HMS application should provide,

- ✚ A user-friendly frontend Interface
- ✚ A developer-friendly repository
- ✚ Administrative functionalities
- ✚ Doctor's tasks automation
- ✚ Patient tasks automation

- ✚ Employee tasks automation
- ✚ Store hospital records and provides when needed
- ✚ Flexible algorithm for financial control
- ✚ Settings panel for all users
- ✚ Communication between different categories of users
- ✚ Automate as many real-world tasks as you can
- ✚ Connect the tasks with each other to ensure smooth workflow

Building Guide and Personal Experience

So, in a Hospital Management System, we have to work with data mostly information of doctors, employees, and patients so that we can give them better services through this application. When working with data, we need to store data in someplace from where we can grab them easily. And Database gives us the exact functionality. I will be using PostgreSQL Relational Database Management System, but you can also go with other options as long you know how to work with the database.

For this kind of complex application, we can not only rely on the database, we need to have a powerful language that will give us the opportunity to not only working with databases but also implementing different algorithms in an object-oriented or functional way. I am using Python for this purpose. You can also use Java, JavaScript, C++, C#, and many more languages.

So, for working in both the PostgreSQL database and Python, we need to install the additional Psycopg2 library. I know the name is a little bit weird. Also, for creating the front-end part, if you want to create a web-app then you can go with Django/Flask, but for the desktop application, I prefer the Kivy framework of Python.

What we need: PostgreSQL, Python, Psycopg2 & Kivy

Now you know what you need to learn and use and how to implement, let me share my personal experience and project features that can help you to build your project at a faster velocity and also give ideas on what you can improve in this project and what difficulties you will face while implementing them.

So, 4 months earlier I was learning about how to work PostgreSQL RDBMS. You can read about my learning timeline and also additional resources and information about PostgreSQL by reading this blog named (#####) After learning the basics as well as the advanced topics of PostgreSQL, I felt the necessity of using my knowledge by creating a project. Otherwise, there is a huge probability that I will forget what I learned. So, I read a couple of blogs on database projects and found that there some projects which are very popular among developers like Hospital Management System, Library Management

System, Bank Management System, Academic Institution Management System, etc. I went for the HMS application. So, for the next 1 month, I created the entire backend part of that application.

After that, I felt exhausted working on the same project for months. As a result, I hadn't created the frontend part with Django at that time. Instead of working on the HMS project, I started to learn the Kivy framework of Python which is one of the best Graphical User Interface framework and provides so many pre-built classes and functions so that developers can create applications that are not only supported in pc but also all other platforms that you can imagine. So, after learning the basics of Kivy, I again felt the necessity of implementing my Kivy knowledge in programming projects. When I was searching for projects, I thought that it would be nice if I could create a user-friendly GUI for the HMS application using Kivy. So, I again started working on the previous project. After more than a month, I finally completed the first version of this Hospital Management System Application.

Here is the list of modules, classes, and functions that are used in this application. The names of the classes and functions describe what they do. So, I won't describe them further.

main.py

- ✚ Have zero function, will be used for glue code.

constant.py

- ✚ grab_constant
- ✚ set_constant

DB_config.py

- ✚ admin_config
- ✚ config

backup_restore.py

- ✚ backup_to_csv
- ✚ restore_from_csv

setup_engine

- ✚ start_program
- ✚ create_database
- ✚ delete_database

- ✚ create_admin_table
- ✚ delete_admin_table
- ✚ create_doctor_table
- ✚ delete_doctor_table
- ✚ create_patient_table
- ✚ delete_patient_table
- ✚ create_employee_table
- ✚ delete_employee_table
- ✚ add_unique_constraint
- ✚ add_check_constraint

login.py

- ✚ DB_pass
- ✚ login
- ✚ signup

admin.py

- ✚ see_info
- ✚ update_db
- ✚ recent_notifications
- ✚ add_notification
- ✚ total_earning
- ✚ add_employee
- ✚ see_my_employee
- ✚ remove_employee
- ✚ show_all_doctor
- ✚ show_all_patient
- ✚ show_all_employee
- ✚ all_doctor_username
- ✚ all_patient_username
- ✚ all_employee_username
- ✚ remove_doctor_permanently
- ✚ remove_patient_permanently
- ✚ remove_employee_permanently
- ✚ patient_joins_doctor

doctor.py

- ✚ recent_notifications
- ✚ notify_admin
- ✚ salary

- ✚ show_all_employee
- ✚ add_employee
- ✚ see_my_employee
- ✚ remove_employee
- ✚ see_all_requested_patient
- ✚ see_all_patients_of_my_speciality
- ✚ see_my_patient
- ✚ see_patients_report
- ✚ remove_patient

patient.py

- ✚ remaining_appointment_time
- ✚ recent_notifications
- ✚ notify_admin
- ✚ cost
- ✚ add_report
- ✚ see_all_doctors_for_my_problem
- ✚ request_doctor
- ✚ remove_request
- ✚ see_my_doctors_stat

employee.py

- ✚ recent_notifications
- ✚ notify_admin
- ✚ salary
- ✚ isreceptionist
- ✚ appoint_doctor
- ✚ see_my_doctors

frontend.py & frontend.kv

- ✚ used for the frontend part, contains multiple classes and functions and also glue code.

resources

- ✚ contains images and other resources

backup_&_restore_folder

- ✚ contains multiple csv files which are vital for the backup and restore process of the application.

For learning more about this application and seeing the code, visit the [GitHub repository](#). You can show some love by giving a star to the repository. If you want to contribute to the repository, you are invited to fork and pull request. I will merge your pull request if that provide constructive changes.