

13

Saturday

08.00

② Building patient medicine and appointment system with springboot, HTML, CSS, JS or thymeleaf.

09.00

STEP 1:- For Backend tools used is IntelliJ

11.00

Here the Swagger is Maven

12.00

I have used spring initialize to download few dependencies required for the project.

14.00

Step @:- <https://start.spring.io/>

15.00

this opens spring initialize under that project → Maven

14

Sunday

Language → Java

Spring boot → 3.4.2 (Version)

↓
project Metadata

↓
Group → com.pmaas

Artifact → Patmedandaptsys

Name → Patmedandaptsys

2

2024 January



PRIYA CEMENT®
Lives Forever

Week - 03

Day 015 - 351

15

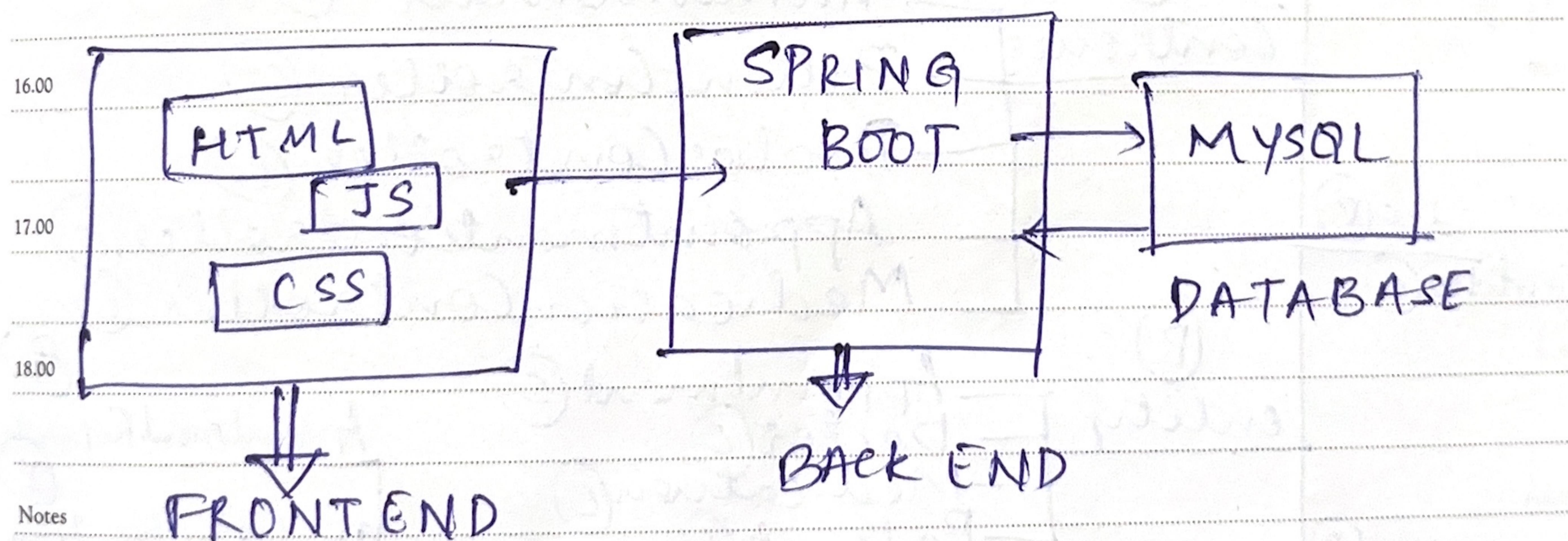
Description → Springboot application - pmaas
Monday
08.00 package name com.pmaas.Patmedandaptsys
packaging - Jar
09.00 Java → 23

10.00 Once the dependencies are added download
ed the zip file extracted opened intelliJ and
11.00 the folder is opened in intelliJ

12.00 Maven is downloaded using
springinitializr and is represented
13.00 by pom.xml file and checked pom.xml
file if everything is downloaded properly
14.00

Step B:-

15.00 Project outline looks as below



Notes

2024	January	2024	February										
S	M	T	W	T	F	S	S	M	T	W	T	F	S
		1	2	3	4	5	6			1	2	3	
		7	8	9	10	11	12	13		4	5	6	7
		14	15	16	17	18	19	20		11	12	13	14
		21	22	23	24	25	26	27		18	19	20	21
		28	29	30	31					22	23	24	25
										26	27	28	29

January 2024

2024 January

Week - 03

Wednesday

16

Tuesday Backend project structure is as below:-

Below:-

```

graph TD
    A[IntelliJ Structure] --> B[Patmedandadapters]
    B --> C[main]
    C --> D[resources]
    D --> E[java]
    E --> F[com.pmaas.Patmedandadapters]
    F --> G[com.pmaas.PatmedandadaptersApplicationTests]
    G --> H[Step 3: MySQL and Postman]
    H --> I[Once all these are created in IntelliJ]
    I --> J[WebConfig C]
    J --> K[PatientController C]
    K --> L[DoctorController C]
    L --> M[AppointmentController C]
    M --> N[Doctor C]
    N --> O[Appointment C]
    O --> P[AppointmentRepository I]
    P --> Q[Medication C]
    Q --> R[User C]
    R --> S[Patient C]
    S --> T[MedicationRepository I]
    T --> U[Medication C]
    U --> V[User C]
    V --> W[Appointment C]
    W --> X[AppointmentRepository I]
    X --> Y[Medication C]
    Y --> Z[User C]
    Z --> AA[Patient C]
    AA --> BB[Medication C]
    BB --> CC[User C]
    CC --> DD[Patient C]
    DD --> EE[Medication C]
    EE --> FF[User C]
    FF --> GG[Patient C]
    GG --> HH[Medication C]
    HH --> II[User C]
    II --> JJ[Patient C]
    JJ --> KK[Medication C]
    KK --> LL[User C]
    LL --> MM[Patient C]
    MM --> NN[Medication C]
    NN --> OO[User C]
    OO --> PP[PatientMapper P]
    PP --> QQ[Patient C]
    QQ --> RR[PatientMapper P]
    RR --> SS[Patient C]
    SS --> TT[MedicationService C]
    TT --> TT[AppointmentService C]
    TT --> TT[DoctorService C]
    TT --> TT[PatientReportService C]
    TT --> TT[MedicationReportService C]
    TT --> TT[PatientService C]
    TT --> TT[UserDetailService C]
    
```

Be sure to save your words before you spit them out.

Used Detail Service C

Day 016 - 350
 Week - 03

January 2024

2024 January

Week - 03

Day 017 - 349
 Wednesday

Wednesday

Step 3:- MySQL and Postman

Once all these are created in IntelliJ

WebConfig C

PatientController C

DoctorController C

AppointmentController C

MedicationController C

entity

Appointment C

Doctor C

Patient C

Medication C

User C

PatientMapper P

Repository P

AppointmentService C

DoctorService C

PatientReportService C

MedicationReportService C

PatientService C

UserDetailService C

Table with rows and columns limiting to 1000

	S	M	T	W	T	F	S	S	M	T	W	T	F	S			
2024	January	2024	February					1	2	3							
				1	2	3	4	5	6								
				7	8	9	10	11	12	13	4	5	6	7	8	9	10
				14	15	16	17	18	19	20	11	12	13	14	15	16	17
				21	22	23	24	25	26	27	18	19	20	21	22	23	24
				28	29	30	31				25	26	27	28	29		

It's not hard to make decisions, when you know what your values are.

18

Thursday

Day 018 - 348



PRIYA CEMENT
Lives Forever

January 2024
Week - 03

2024 January
Week - 03

id	contact	medical-history	name
1	9876543210	Pre-diabetes, Hypertension	Anitha Amin
2	9131789745	Anxiety	Bharika B
3	745817165	Depression	Kumareenuma
5	8765432109	Knee Pain	Katte Gambika

Now we need to see how postman looks like to test if the configured is proper or not we are using postman.

Step ⑥:-
In Postman,

Selected post and type

<http://localhost:8080/patients>

Select content-type as application/json under Headers.

Select raw and json under Body.

2

08.00

and add the entry as below.

09.00

```

f
  "name": "Gambika",
  "contact": "8765432109",
  "medicalHistory": "knee pain"
g
  
```

10.00

11.00

12.00

and then click on send and you can see the output under the response.

13.00

14.00

and GET is used to see if the stored value can be seen for that we need to select

15.00

GET <http://localhost:8080/patients/1>

16.00

and under response I can see the output.

17.00

18.00

UPDATE option is used to update the data and DELETE is used to delete the data or the entry once click on send it will ask if it can be deleted and then it says deleted successfully.

Notes

2

2024 January							2024 February						
S	M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6			1	2	3			
7	8	9	10	11	12	13	4	5	6	7	8	9	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24
28	29	30	31				25	26	27	28	29		

20

Saturday

Day 020 - 346



PRIYA CEMENT®
Lives Forever

Step 4:- For Frontend → tools used

Folder Name

→ PMAAS-MINI 2
↓
Vs code

Select React and file name → pmaas-frontend
Javascript and give file name

January 2024

Week - 03

2024 January

Week - 04

08.00

09.00

10.00

11.00

12.00

13.00

14.00

15.00

16.00

17.00

18.00

Notes

21

Sunday

The code structure for the Vscode can be written as-

2024 January

Week - 04



PRIYA CEMENT®
Lives Forever

Day 022 - 344

22

08.00

09.00

10.00

11.00

12.00

13.00

14.00

15.00

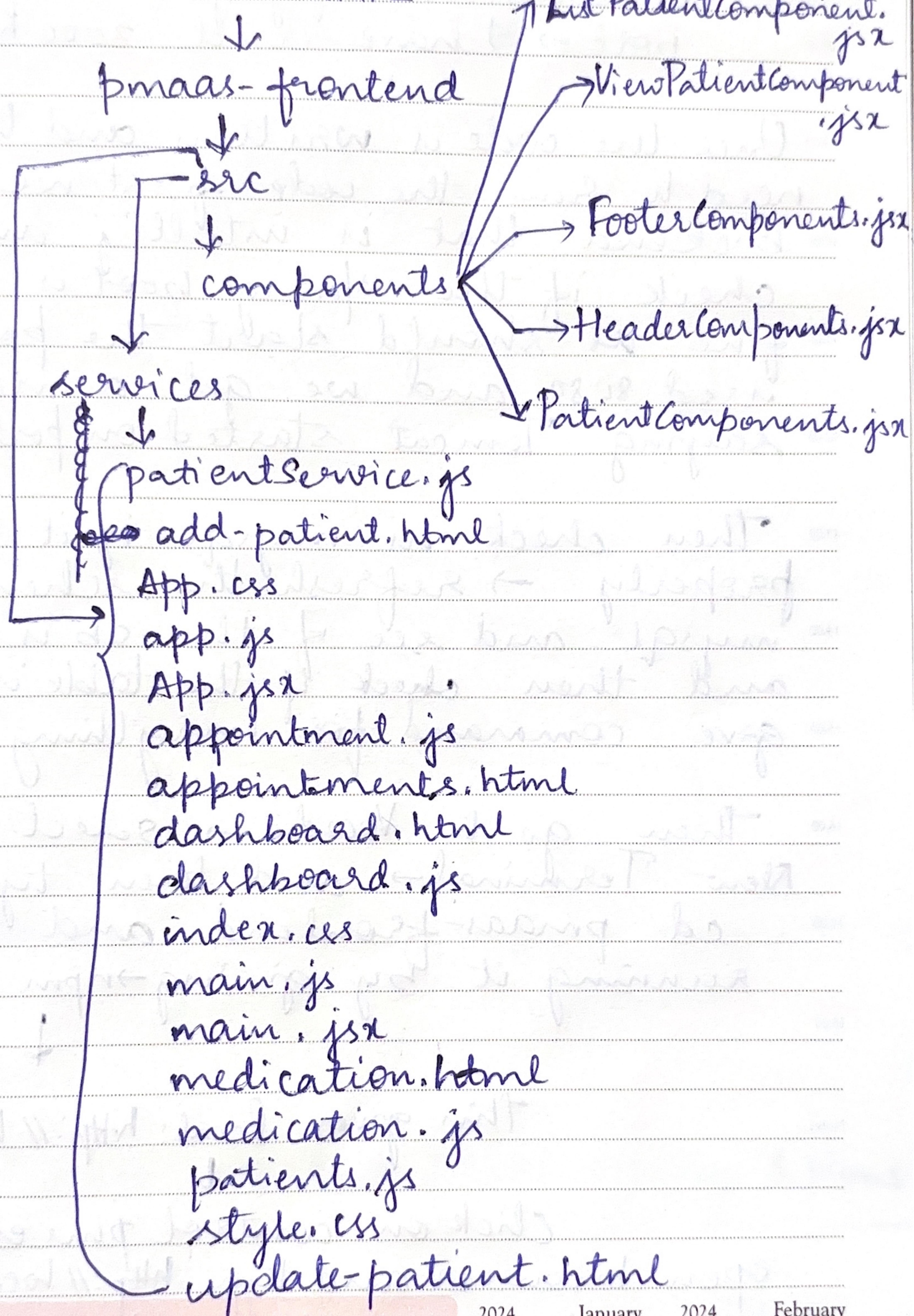
16.00

17.00

18.00

Notes

PMAAS - MINI 2



2024 January							2024 February					
S	M	T	W	T	F	S	S	M	T	W	F	S
1	2	3	4	5	6			1	2	3		
7	8	9	10	11	12	13	4	5	6	7	8	9
14	15	16	17	18	19	20	11	12	13	14	15	16
21	22	23	24	25	26	27	18	19	20	21	22	23
28	29	30	31				25	26	27	28	29	

You can not give away Kindness; it will always come back to you.

23

Tuesday



Day 023 - 343

January 2024
Week - 04

Under vite.config.js → configure the port → I have used 3000 here.

Once the code is written and then we need to run the code first we will run backend that is intelliJ and then check if the spring boot is working fine at should start the port I have used 8080 and we get a message saying Tomcat started on port 8080

Then check in mysql if it is configured properly → refresh the schema in mysql and see if the db is created and then check if the table is created give command for everything.

then go to Vscode → select Terminal → New Terminal → and then type cd pmaas-frontend and then try running it by giving → npm run dev

Notes

This gives local: http://localhost:3000/

click on control plus enter or

open browser and type http://localhost:3000

and click on enter and check if your project is working fine.

2024 January

Week - 04

My project

08.00

09.00

10.00

Name

11.00

Contact

12.00

Medical +

13.00

ID

14.00

[Add Patient]

15.00

Patients

Notes

ID

No

2



2024 January



PRIYA CEMENT®
Lives Forever

Week - 04

Day 024 - 342

24

Wednesday

My project looks like below

08.00

09.00

10.00

11.00

12.00

13.00

14.00

15.00

16.00

17.00

18.00

Patient Management System

Name

Contact

Medical History

Add Patient

Patients

Notes

ID	Name	Contact	Medical History	Actions
----	------	---------	-----------------	---------

2

2024 January							2024 February						
S	M	T	W	T	F	S	S	M	T	W	T	F	S
			1	2	3	4	5	6			1	2	3
			7	8	9	10	11	12	13		4	5	6
			14	15	16	17	18	19	20		11	12	13
			21	22	23	24	25	26	27		14	15	16
			28	29	30	31					17	18	19
											20	21	22
											23	24	25
											26	27	28
											29		

Condemn the fault, and not the actor of it.