



MILESTONE 2 BACKEND REPORT

Almost 5

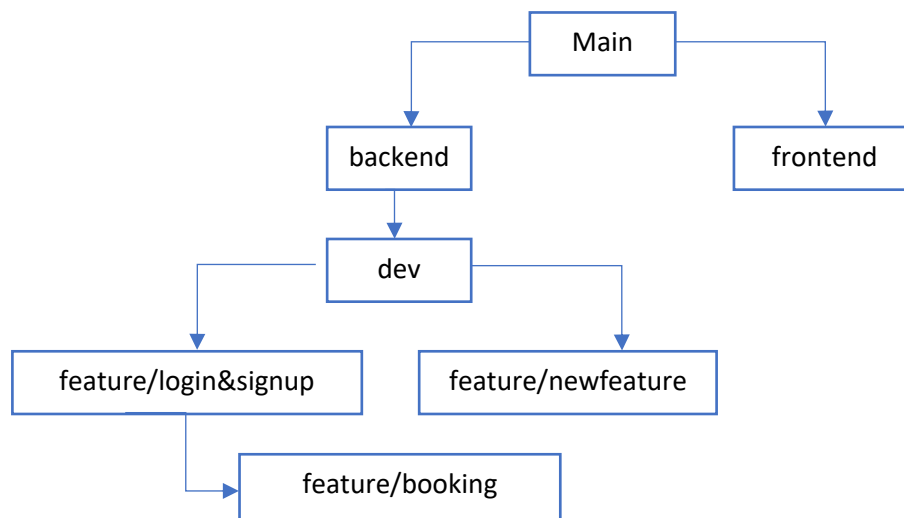
Backend Team

Contents

Branches Structure.....	2
Other Branches	2
File Structure.....	2
Database	3
Postman Testing.....	3
Signup Endpoints.....	3
Login Endpoints.....	5
Booking Endpoints	6
CI with Github	8
Next Steps	10

Branches Structure

For this project we have the structure of our git hub repository broken down in this format this is until sprint 2



Note -

- As the booking feature needs to be **inline with the users it was branches out from the Login and signup branch**
- Once we are done with the integration of the backend to frontend, we will merge the finished features into the dev branch and then onto the backend branch as we have not implemented integration, we are not doing this for this submission

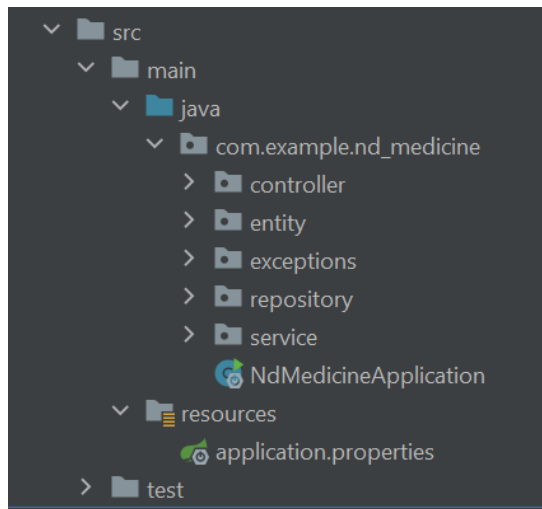
Other Branches

We discovered that we worked more efficiently when we were really implementing—that is, writing some code—rather than just reading about it and seeking up resources—so we started using private repositories during the development of the backend for each feature during sprint 1.

- ❖ To demonstrate that everyone was working, we built **trial/ branches** in the main repository SEPT-Project--Almost-5 for each feature
- ❖ This was to divide the work across the three backend team members.
- ❖ You can view the trial/ branches for each feature to see commits and who made which contributions.
- ❖ The final working feature is in the feature/ (name of feature) branch which includes the Junit testing.

File Structure

As instructed in class, each feature makes use of a standard file structure. The development branch adheres to this file structure, and as all features branch off the development branch, they all do as well



Packages are added as in needed by the feature configurations, but all follow this file structure.

Database

Using H2 database access on - <http://localhost:8080/h2-console>

The Database is not storing users that are being created at the moment as we have not created delete endpoint, so the database resets itself after each run.

Postman Testing

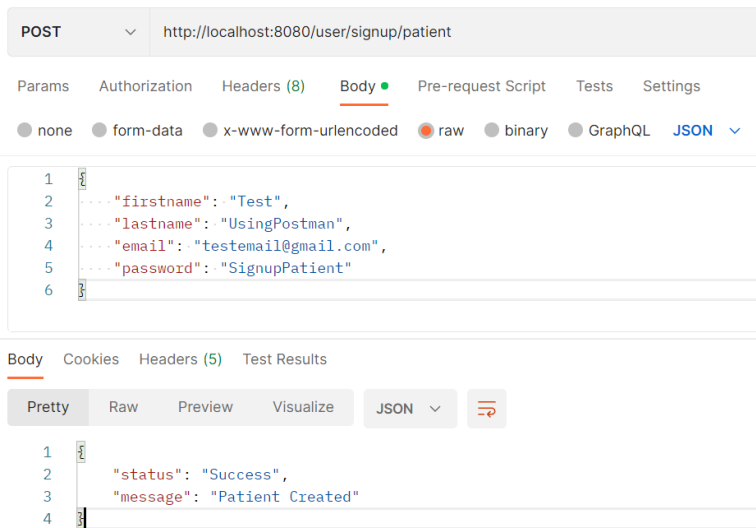
As previously mentioned, we did not integrate the front end and back end for this sprint. To test each feature's functionality is covered in the backend and it runs independently of frontend we have used Postman.

- Each feature has its own controller that has endpoints to connect and show the function of that feature.

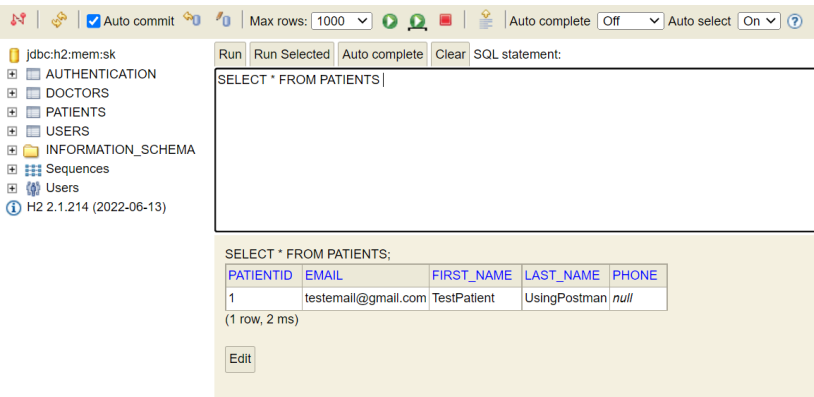
Signup Endpoints

1. Signup Patient

Method	POST
Endpoint	http://localhost:8080/user/signup/patient
Body	<pre>{ "firstname": "Test", "lastname": "UsingPostman", "email": "testemail@gmail.com", "password": "SignupPatient" }</pre>
Response	<pre>{ "status": "Success", "message": "Patient Created" }</pre>



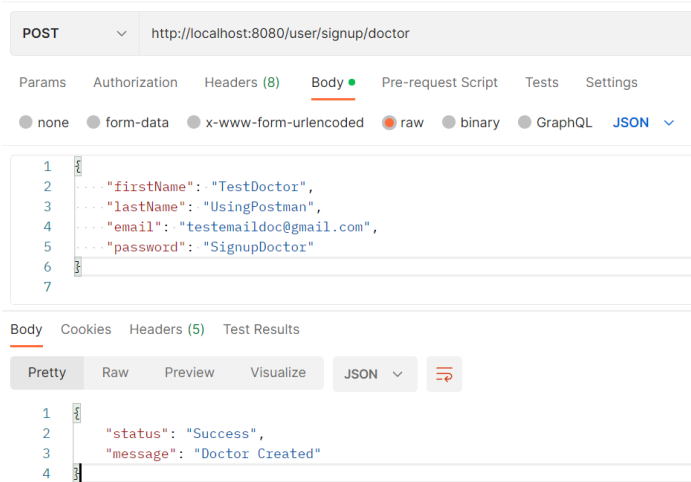
1. Signup Patient Endpoint from Postman



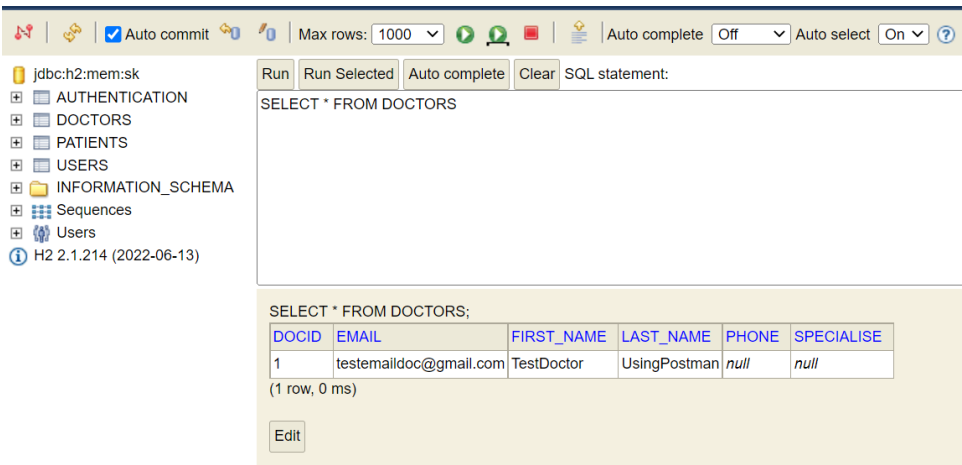
2. Database Patient Signup

2. Signup Doctor

Method	POST
Endpoint	http://localhost:8080/user/signup/doctor
Body	<pre>{ "firstName": "TestDoctor", "lastName": "UsingPostman", "email": "testemaildoc@gmail.com", "password": "SignupDoctor" }</pre>
Response	<pre>{ "status": "Success", "message": "Doctor Created" }</pre>



3. Signup Doctor Endpoint from Postman



4. Database Doctor Signup

Login Endpoints

Method	POST
Endpoint	http://localhost:8080/user/login
Body	<pre>{ "email": "testemail@gmail.com", "password": "SignupPatient" }</pre>
Response	<pre>{ "status": "Success", "token": "1188ed59-02ba-481c-a0f4-4d9c4aee9a0e" }</pre>

The first two screenshots show REST client requests to `http://localhost:8080/user/login`. The left request is for a doctor login, and the right is for a patient login. Both return a JSON response with `"status": "Success"` and a `"token"`.

The third screenshot shows a database query result for the `AUTHENTICATION` table. The query is `SELECT * FROM AUTHENTICATION;`. The result shows two rows of data:

ID	TOKEN	USER_ID
1	5e2741db-0caa-4b09-a351-8bee4ebd0f25	1
2	1188ed59-02ba-481c-a0f4-4d9c4aee9a8e	2

Booking Endpoints

Limitations

- We have hard coded the current doctor and current patient as we have no control over which doctor is currently using the app and which patient is logged in we will be able to do this after integration
- For now, hard coded doctor “Sai Kannan” adds his availability
- And “saiPatient” books availability for doctor “SaiKannan”

1. Adding Availability

Method	POST
Endpoint	http://localhost:8080/doctor/availability
Body	<pre>{ "startTime": "09:00", "endTime": "12:00" }</pre>
Response	<pre>{ "availId": 2, "doctor": { "id": 1, "firstname": "sai", "lastname": "kannan", "email": "sai@gmail", "phone": null, "specialise": null }, }</pre>

	<pre>"startTime": "09:00:00", "endTime": "12:00:00" }</pre>
--	---

2. Creating Appointments

Method	POST
Endpoint	http://localhost:8080/appointment/createAppointment
Body	<pre>{ "time": "09:00", "appointmentDate": "10-10-22" }</pre>
Response	<pre>{ "doctor": { "id": 2, "firstname": "saiDoc", "lastname": "kannan", "email": "sai@gmail", "phone": null, "specialise": null }, "patient": { "id": 3, "firstname": "saiPatient", "lastname": "kannan", "phone": null }, "time": "09:00:00", "appointmentDate": "10-10-0022", "id": 1 }</pre>

jdbch2:mem:sk
APPOINTMENTS
AUTHENTICATION
DOCTORAVAILABILITIES
DOCTORS
DOCID
EMAIL
FIRST_NAME
LAST_NAME
PHONE
SPECIALISE
Indexes
PATIENTS
SLOTS
USERS
INFORMATION_SCHEMA
Sequences
Users
H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:
SELECT * FROM SLOTS;

SELECT * FROM SLOTS;

SLOT_ID	BOOKED	SLOT_TIME	DOCID
1	FALSE	09:00:00	1
2	FALSE	09:30:00	1
3	FALSE	10:00:00	1
4	FALSE	10:30:00	1
5	FALSE	11:00:00	1
6	FALSE	11:30:00	1
7	FALSE	12:00:00	1

(7 rows, 1 ms)
Edit

5 30min slots are created for doctor from this start time to end time

CI with Github

We are using JAVA CI with maven on GitHub as our pipeline for CI.

The screenshot shows the GitHub Actions interface for the 'Java CI with Maven' workflow. The left sidebar lists 'All workflows' and 'Java CI with Maven'. The main panel shows '7 workflow run results' for the 'branch:feature/login&signup' search. The results are as follows:

Run Name	Status	Branch	Event	Time	Actor
File rename Tests	Success	feature/login&signup	Push	1 hour ago	ms-hacker404
All Tests Completed for Feature/Login&Signup	Success	feature/login&signup	Push	6 hours ago	DumpleDorf
4 Tests Completed, Incomplete Sign in Test (Need further...	Success	feature/login&signup	Push	17 hours ago	DumpleDorf
givenDuplicatePatient_whenSigningUp_thenAddNewPati...	Failure	feature/login&signup	Push	18 hours ago	DumpleDorf
First Working MVC Test	Success	feature/login&signup	Push	20 hours ago	DumpleDorf
Created MVC (Non-Functional)	Failure	feature/login&signup	Push	2 days ago	DumpleDorf
Create maven.yml	Success	feature/login&signup	Push	3 days ago	ms-hacker404

6. JAVA CI for feature/login&signup

The screenshot shows the GitHub Actions interface for the 'Java CI with Maven' workflow. The left sidebar lists 'All workflows' and 'Java CI with Maven'. The main panel shows '2 workflow run results' for the 'branch:feature/booking' search. The results are as follows:

Run Name	Status	Branch	Event	Time	Actor
booking done!!	Success	feature/booking	Push	8 hours ago	ms-hacker404
Finished Availability and Registration Controllers. Creat...	Success	feature/booking	Push	6 days ago	DumpleDorf

7. JAVA CI for feature/booking

Workflows

New workflow

Java CI with Maven

maven.yml

Filter workflow runs

42 workflow runs

Event

Status

Branch

Actor

File rename Tests

Java CI with Maven #43: Commit 9beac6a pushed by ms-hacker404

feature/login&signup

1 hour ago

24s

file Structure generic

Java CI with Maven #42: Commit 02c3840 pushed by ms-hacker404

dev

2 hours ago

26s

service updated

Java CI with Maven #41: Commit 2eaf97e pushed by ms-hacker404

trial/signup

3 hours ago

34s

All Tests Completed for Feature/Login&Signup

Java CI with Maven #40: Commit 1345f9f pushed by DimpleDorf

feature/login&signup

6 hours ago

24s

booking done!!

Java CI with Maven #39: Commit d59c991 pushed by ms-hacker404

feature/booking

8 hours ago

34s

booking done!!

Java CI with Maven #38: Commit d59c991 pushed by SaicharanKannan

feature/login,signup,appoint...

13 hours ago

22s

4 Tests Completed, Incomplete Sign in Test (Need furthe...

Java CI with Maven #37: Commit 64c7ec7 pushed by DimpleDorf

feature/login&signup

17 hours ago

28s

givenDuplicatePatient_whenSigningUp_thenAddNewPati...

Java CI with Maven #36: Commit baed116 pushed by DimpleDorf

feature/login&signup

18 hours ago

32s

First Working MVC Test

Java CI with Maven #35: Commit e9e23d9 pushed by DimpleDorf

feature/login&signup

20 hours ago

35s

chat controller config files

Java CI with Maven #34: Commit 0934c1b pushed by ms-hacker404

feature/chat

yesterday

21s

chat controller config files

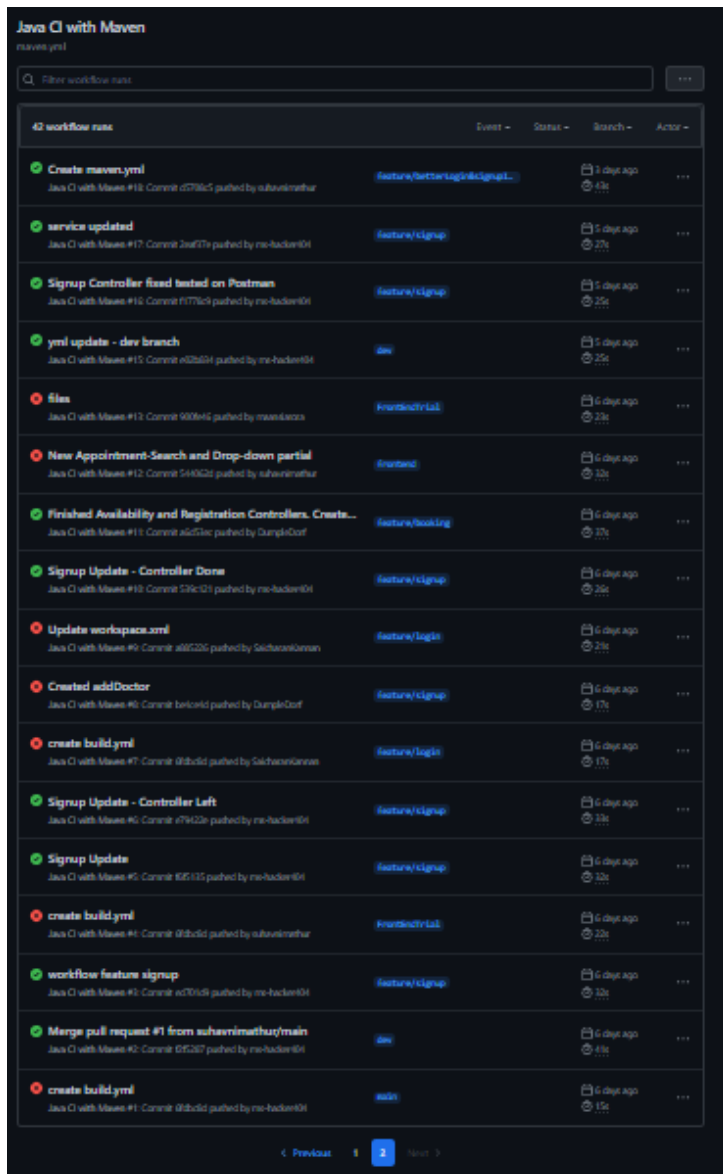
Java CI with Maven #33: Commit 0934c1b pushed by ms-hacker404

feature_chat

yesterday

23s

Chat entities and updated pom.xml	Java CI with Maven #32: Commit 977e5c5 pushed by ms-hacker404	feature/chat	yesterday	54s	
Chat entities and updated pom.xml	Java CI with Maven #31: Commit 977e5c5 pushed by ms-hacker404	feature_chat	yesterday	50s	
Created MVC (Non-Functional)	Java CI with Maven #30: Commit 87fe88d pushed by DimpleDorf	feature/login&signup	2 days ago	17s	
Chat Files setup	Java CI with Maven #29: Commit bd9f584 pushed by ms-hacker404	feature/chat	2 days ago	38s	
Chat Files setup	Java CI with Maven #28: Commit bd9f584 pushed by ms-hacker404	feature_chat	2 days ago	31s	
Created First MVC Test (Issue with JSON Path) (Pushed fo...	Java CI with Maven #27: Commit 7e6d7b1 pushed by DimpleDorf	trail/appointment	3 days ago	37s	
Created First MVC Test (Issue with JSON Path) (Pushed fo...	Java CI with Maven #26: Commit db1f794 pushed by DimpleDorf	trail/appointment	3 days ago	29s	
Created First MVC Test (Issue with JSON Path) (Pushed fo...	Java CI with Maven #25: Commit 8be9290 pushed by DimpleDorf	trail/appointment	3 days ago	35s	
Fixed Build Error, Finished 1st Test	Java CI with Maven #24: Commit bfacab1 pushed by DimpleDorf	trail/appointment	3 days ago	37s	
Merge remote-tracking branch 'origin/feature/login,sign...	Java CI with Maven #23: Commit 8cf32f1 pushed by SaicharanKannan	feature/login,signup,appoint...	3 days ago	17s	
Create booking_build.yml	Java CI with Maven #22: Commit d8212e0 pushed by suhavnimathur	feature/login,signup,appoint...	3 days ago	39s	
Create maven.yml	Java CI with Maven #21: Commit d5708c5 pushed by ms-hacker404	feature/login&signup	3 days ago	33s	
service updated	Java CI with Maven #20: Commit 2eaf97e pushed by ms-hacker404	trial/individual_signup	3 days ago	41s	
Create maven.yml	Java CI with Maven #19: Commit 6405f5d pushed by suhavnimathur	trail/appointment	3 days ago	23s	



The screenshot displays the 'Java CI with Maven' workflow runs page on GitHub. It features a search bar at the top and a table of workflow runs. The table columns are 'workflow name', 'Event', 'Status', 'Branch', and 'Actor'. The runs are listed in descending order of time, with the most recent at the top. Each row includes a green checkmark for successful runs or a red circle with an exclamation mark for failed runs. The 'workflow name' column contains the name of the workflow, the 'Event' column shows the trigger (e.g., 'push', 'pull_request'), the 'Status' column shows the result (e.g., 'success', 'failure'), the 'Branch' column shows the branch name, and the 'Actor' column shows the user who triggered the run. The table is currently showing 15 runs, with a pagination bar at the bottom indicating '1' of 2 pages.

workflow name	Event	Status	Branch	Actor
Create maven.yml	push	success	feature/setting-up-the-backend	subhramathur
service updated	push	success	feature/signup	me-hacker001
Signup Controller fixed tested on Postman	push	success	feature/signup	me-hacker001
yml update - dev branch	push	success	dev	me-hacker001
files	push	failure	feature/try-1-1	me-hacker001
New Appointment-Search and Drop-down partial	push	failure	feature/	subhramathur
Finished Availability and Registration Controllers. Create...	push	success	feature/booking	DimpleDor
Signup Update - Controller Done	push	success	feature/signup	me-hacker001
Update workspace.xml	push	failure	feature/login	SakshamGoswami
Created addDoctor	push	failure	feature/signup	DimpleDor
create build.yml	push	failure	feature/login	SakshamGoswami
Signup Update - Controller Left	push	success	feature/signup	me-hacker001
Signup Update	push	success	feature/signup	me-hacker001
create build.yml	push	failure	feature/try-1-1	subhramathur
workflow feature signup	push	success	feature/signup	me-hacker001
Merge pull request #1 from subhramathur/main	pull_request	success	dev	me-hacker001
create build.yml	push	failure	dev	me-hacker001

Next Steps

We will be integrating login signup and booking to the frontend and working on chat and store the users on the database once we will have integrated frontend