mobius
knowledge services

Redevelopment of webapp UI
And
Towards CI/CD method of software development

Niveydhithaa R
17PW25
28.05.2022

Agenda

- Introduction to organization

- Project Work

  - About the project

  - My contribution

- Work environment

- Tools and technologies used

- Timeline

- Challenges faced

Special Thanks to

- **Dr. R. Nadarajan**, *Professor & Head*,
  Department of Applied Mathematics and Computational Sciences, PSG College of Technology

- **Dr. R. M. Periakaruppan**, *Program Coordinator MSc Software Systems*,
  Department of Applied Mathematics and Computational Sciences, PSG College of Technology

- **Dr. Mohanraj. N**, *Associate Professor*,
  Department of Applied Mathematics and Computational Sciences, PSG College of Technology

- **Mr. Sankar Sundaram**, *CTO*,
  Mobius Knowledge Services

- **Mr. Prashanth Ravindran**, *Associate Director ML&AI Division*,
  Mobius Knowledge Services

Introduction to the organization

- Founded in 2002, the aim of Mobius is to offer platforms and products to companies to utilize the wealth of data, trapped in complex sources
- Employ big data technologies, robotic process automation, artificial intelligence, and self-learning bots in the solutions, currently offered to leading Fortune 2000 companies
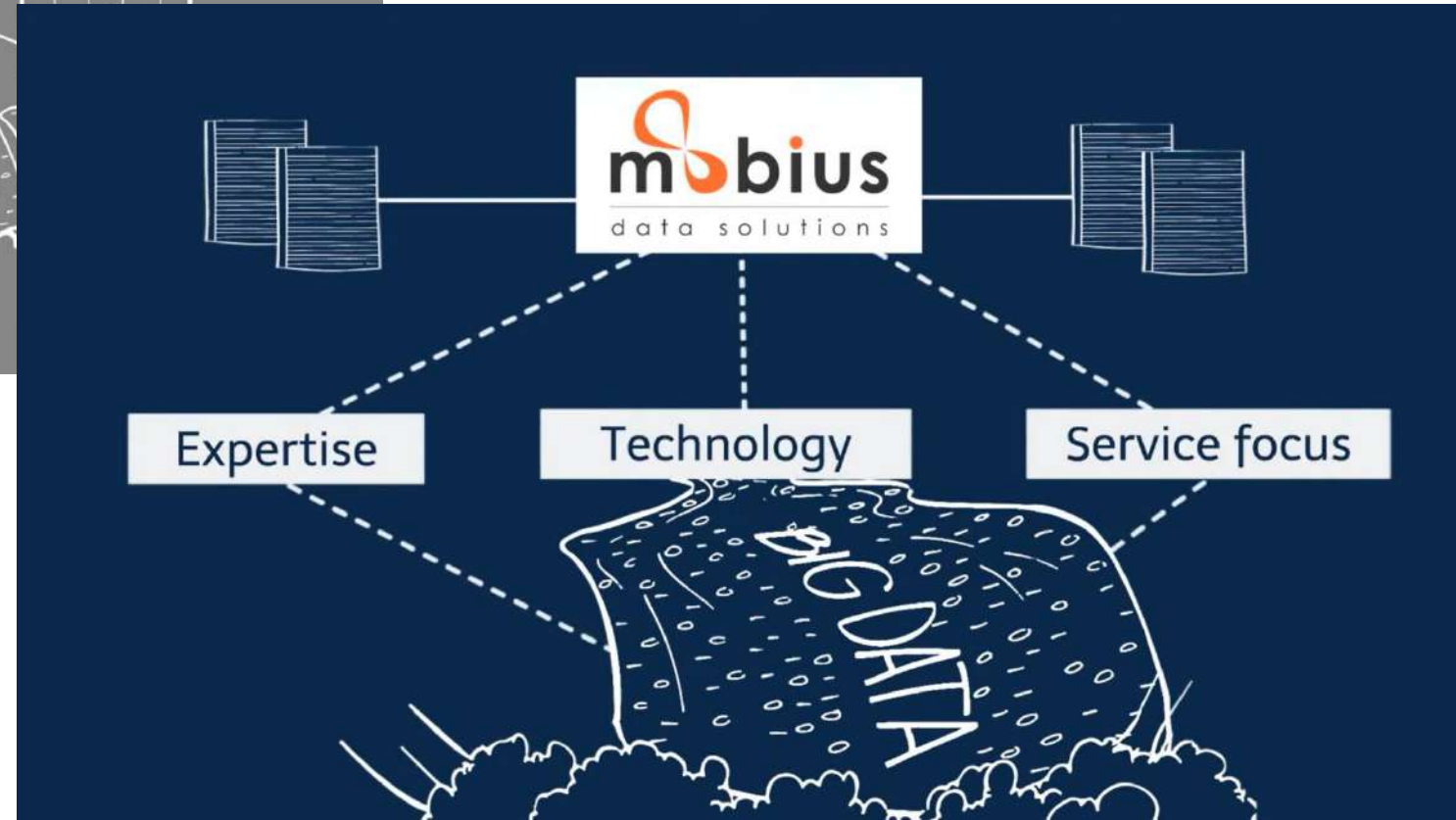- The Chief Executive Officer of the company is Mr. Karthik Karunakaran
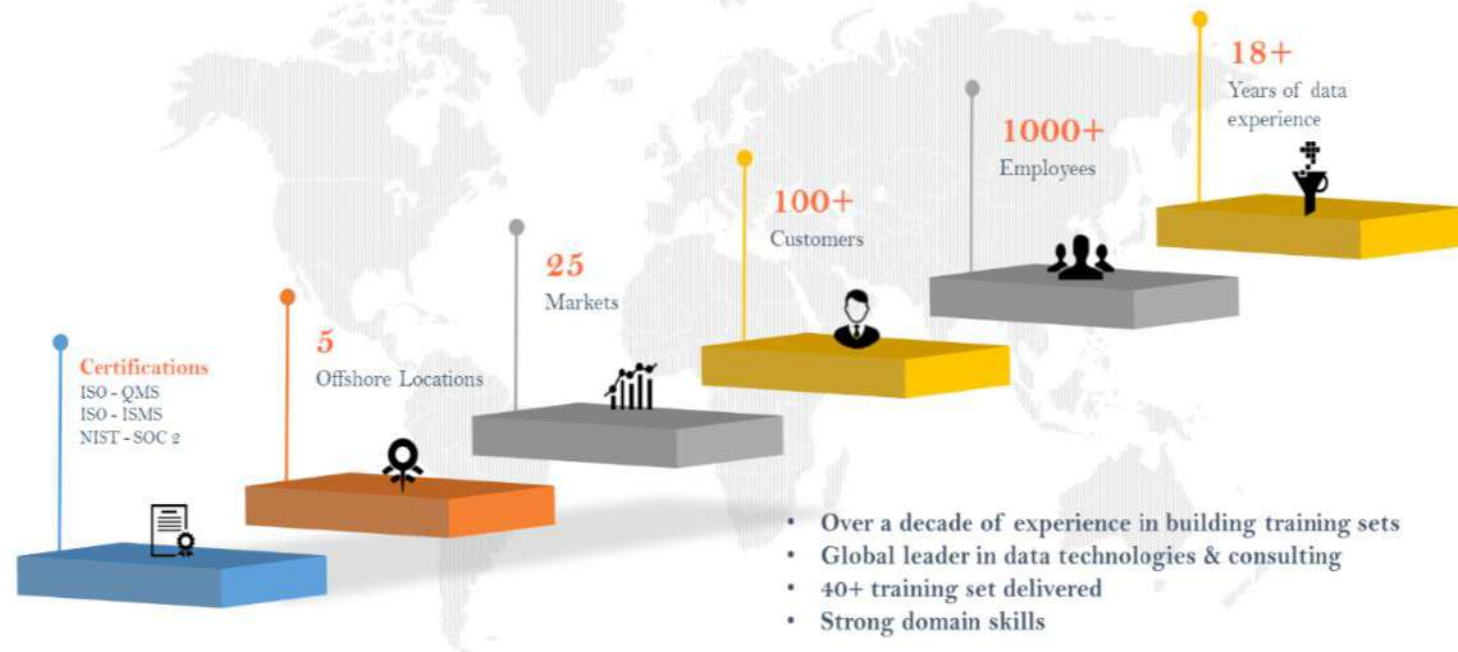
Introduction to the organization

Fig.1a Overwhelming data at the company side

Fig.1b How Mobius deals the data problem
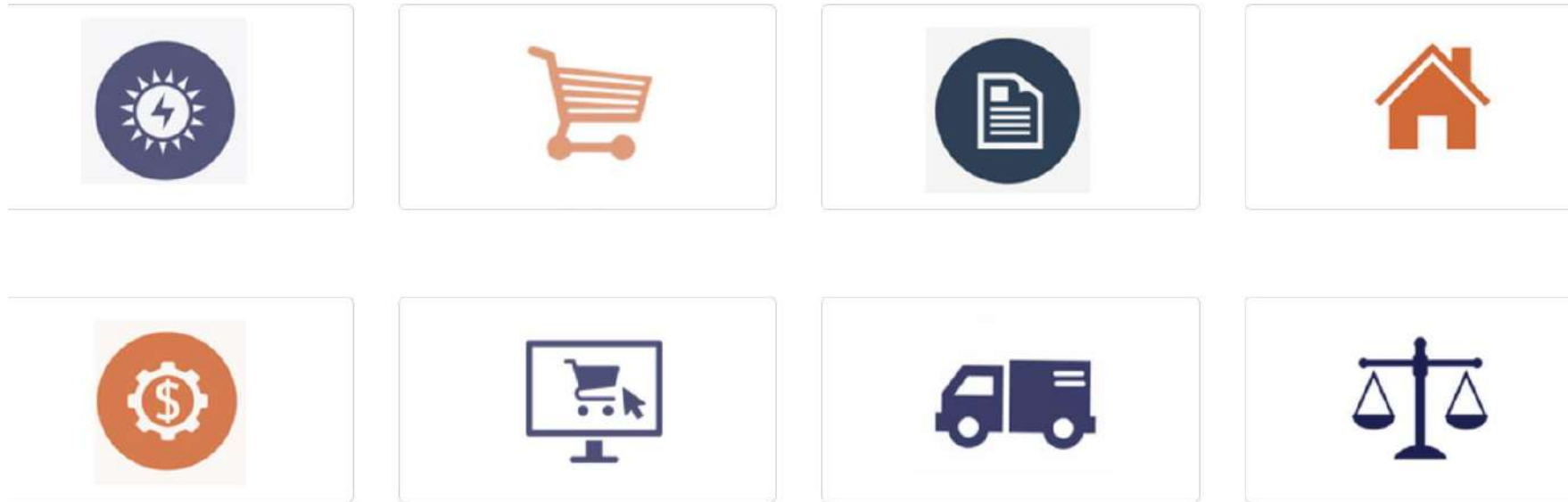
Introduction to the organization
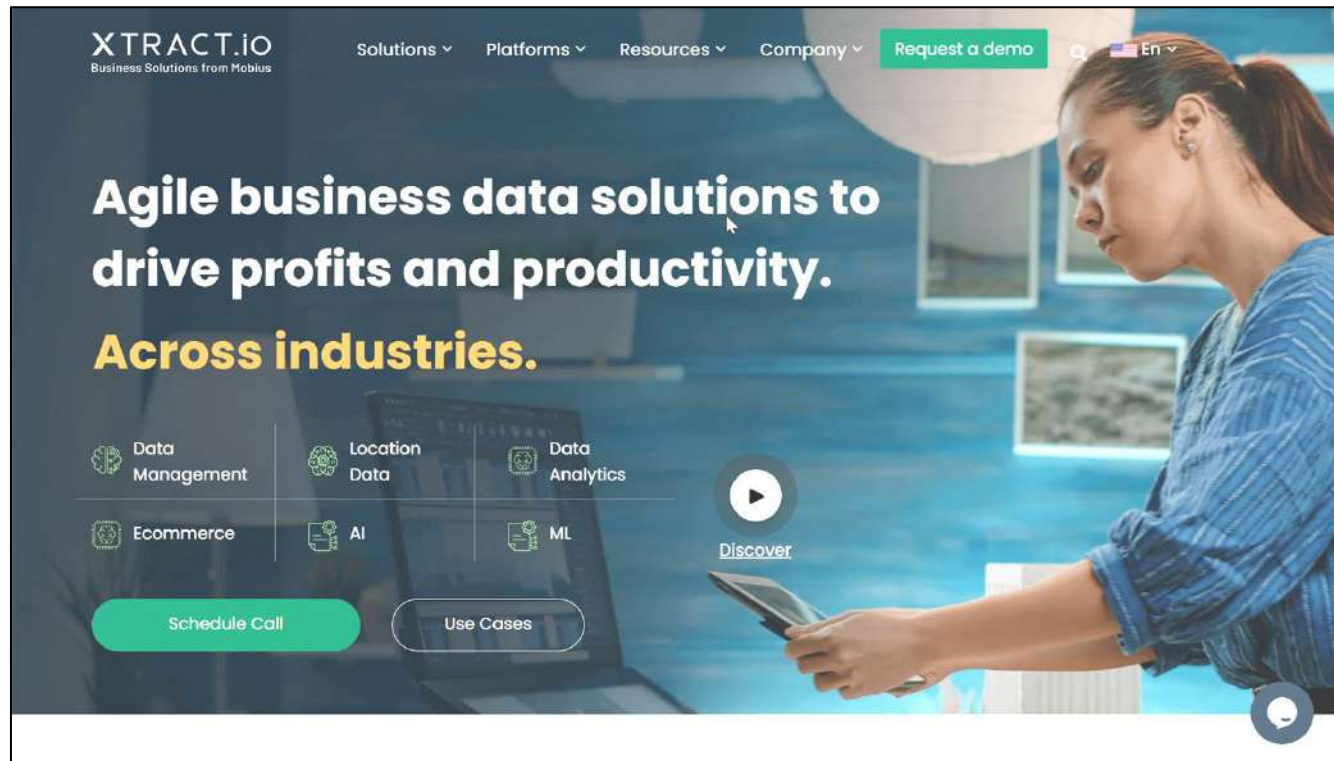
Fig.2 Various domains Mobius serves

PROJECT WORK

Two phases of internship work
The work is briefly explained in the upcoming slides

ABOUT **FINXTRACT**

FinXtract

- Xtract.io, catering to multiple industries across the world
- Helps businesses grow into new markets

Work done by me

- As part of the project, contributed to the team for redeveloping the existing FinXtract webapp.

- Tasks were focused on
  - choosing the right framework to redevelop
  - creating an efficient web app
  - enhance the authentication and session storage
  - minimal client-side with login and register functions

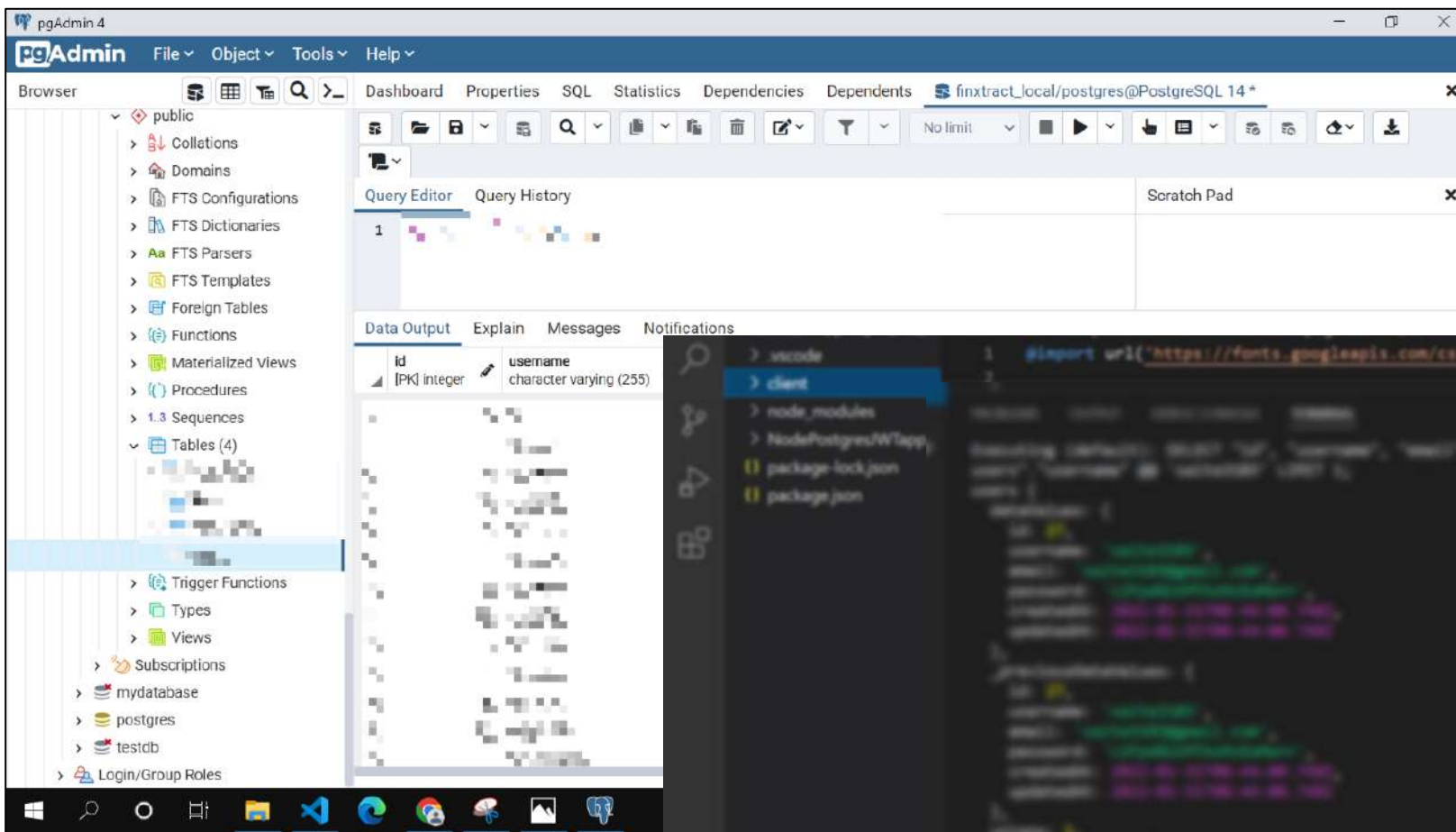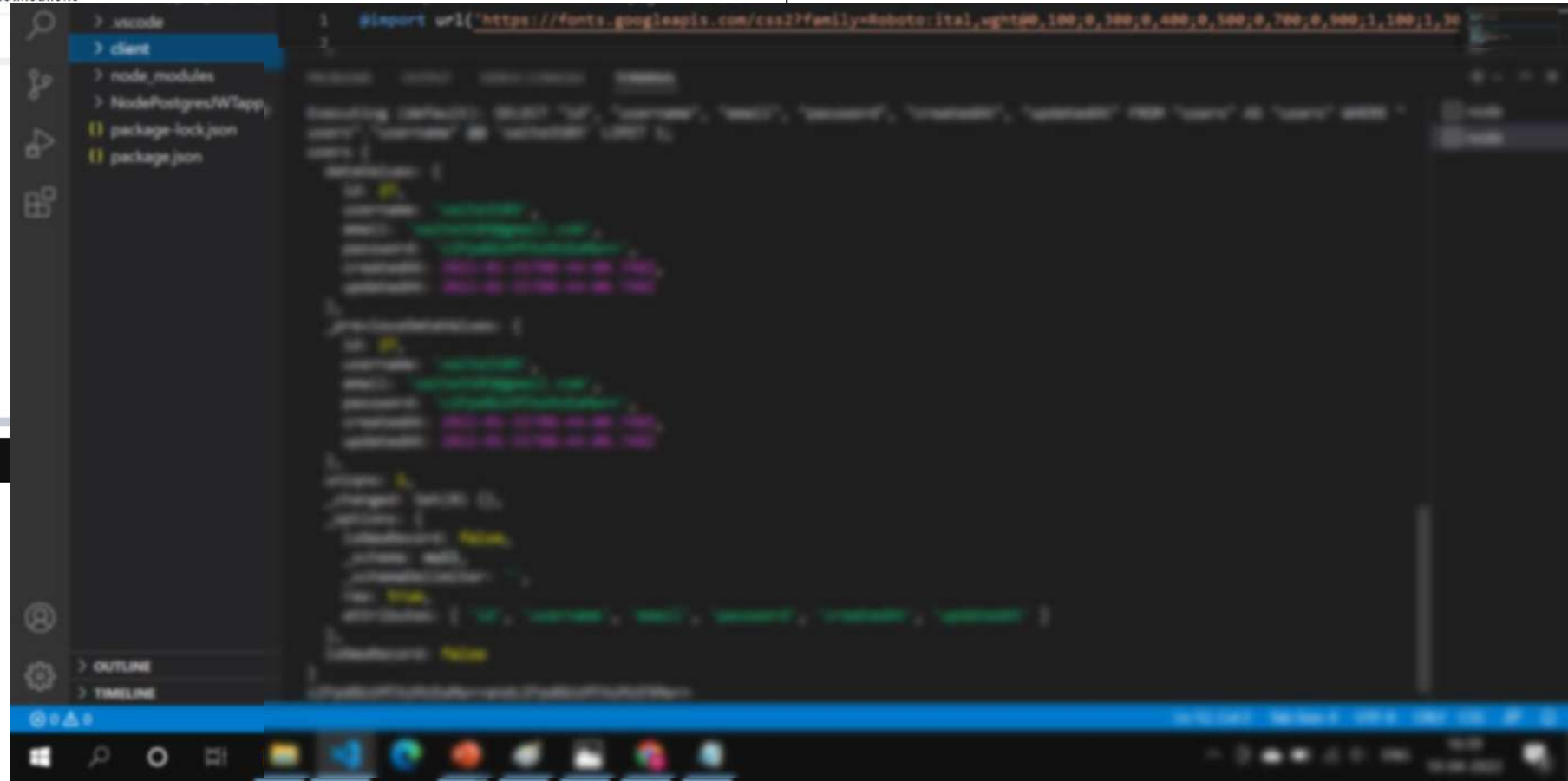|	Existing tech	Modified
Application	Python	React + Node (server)
Database	PostgreSQL	PostgreSQL

Screenshots

Fig.3 PGAdmin to access Postgres DB

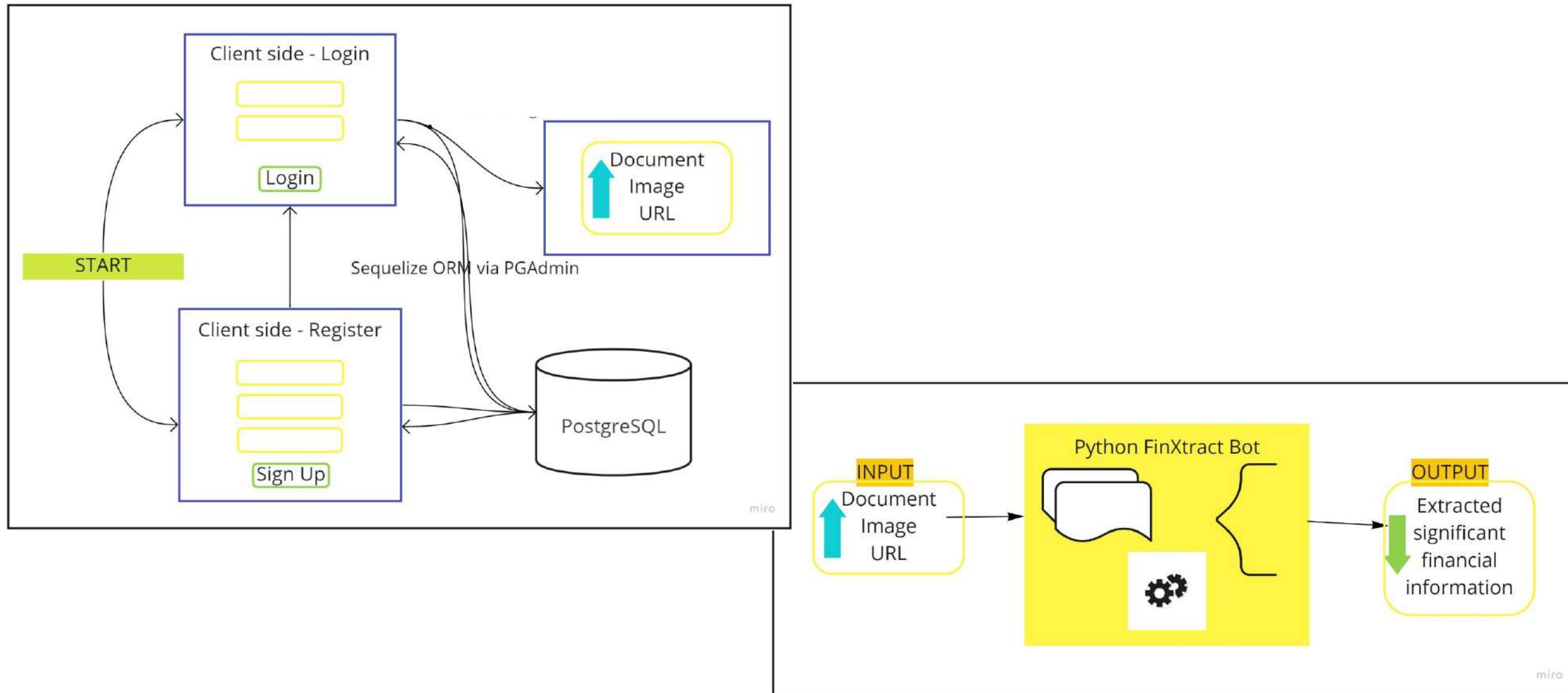Fig.4 VSCode snap from the webapp UI code

Workflow

Fig.5a, b: Finxtract Webapp & Bot workflow

ABOUT THE PROJECT - **NAIOMI**

NAiOmi Overview

- Cancer detection and diagnosis
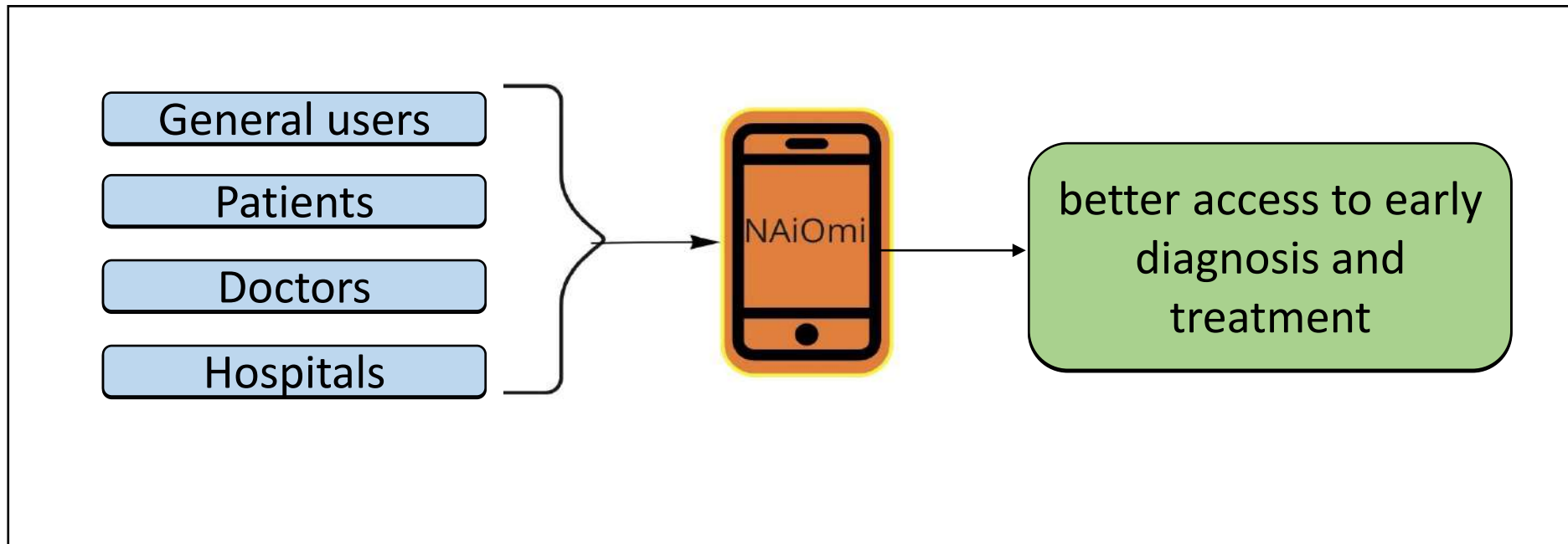 - *Every year, new cancer patients registered: Over 11,57,294 lakh. But do all of them get diagnosed and treated?*

Fig.6 Need for NaiOmi

Towards CI/CD

- Continuously deliver code to production
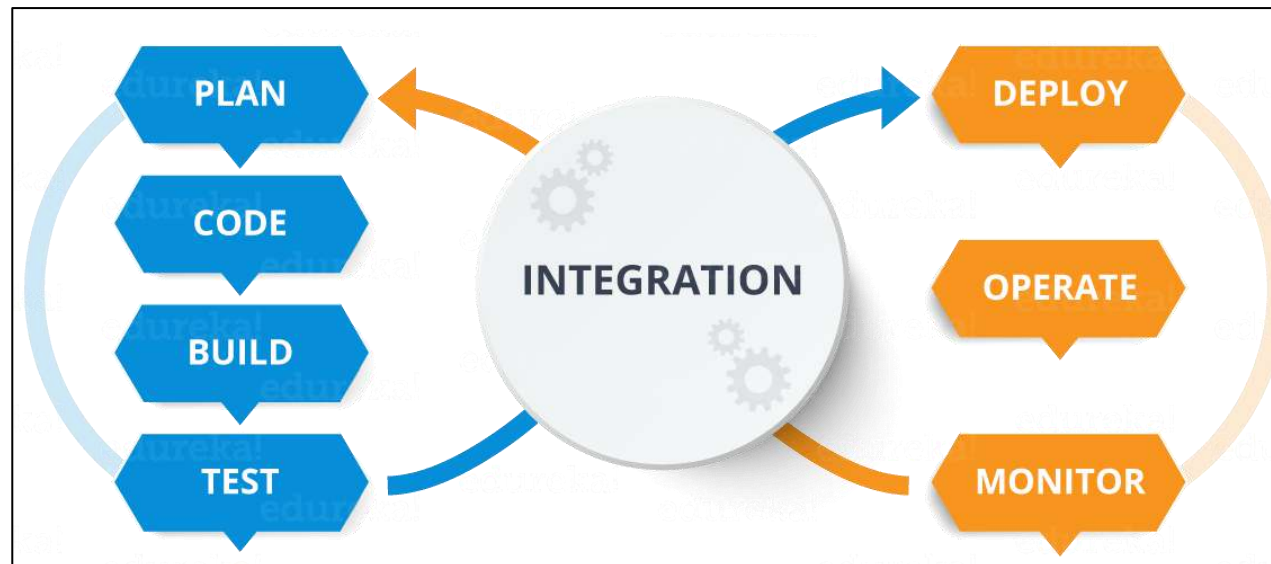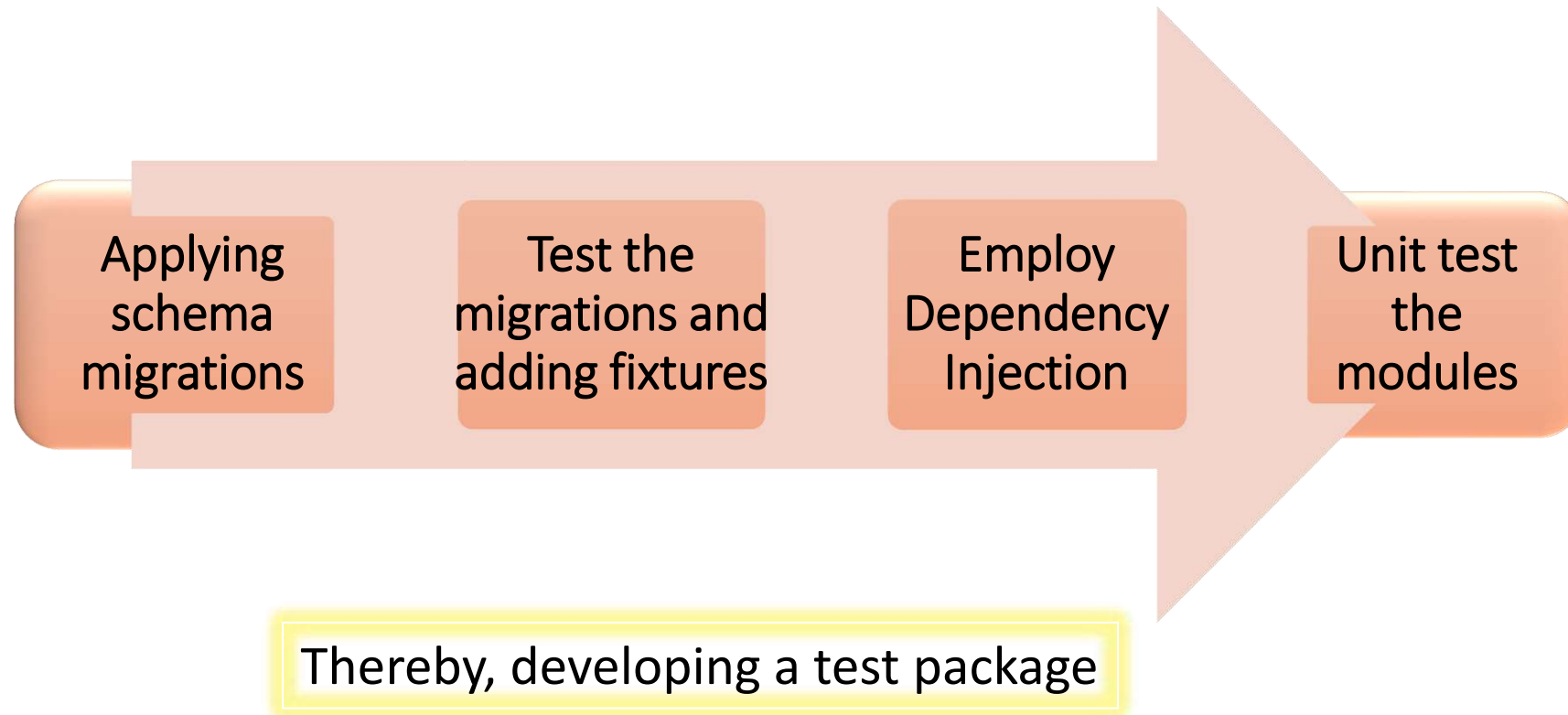- Ongoing flow of new features and bug fixes

Fig.7 CI/CD pipeline

Work done by me

Applying schema migrations

Test the migrations and adding fixtures

Employ Dependency Injection

Unit test the modules

Thereby, developing a test package

Migrations

- Changing requirements through the development process
- Migrations ensure backward compatibility
  - *Analogy, Github: Source code version control ; Migrations: DB changes version control*
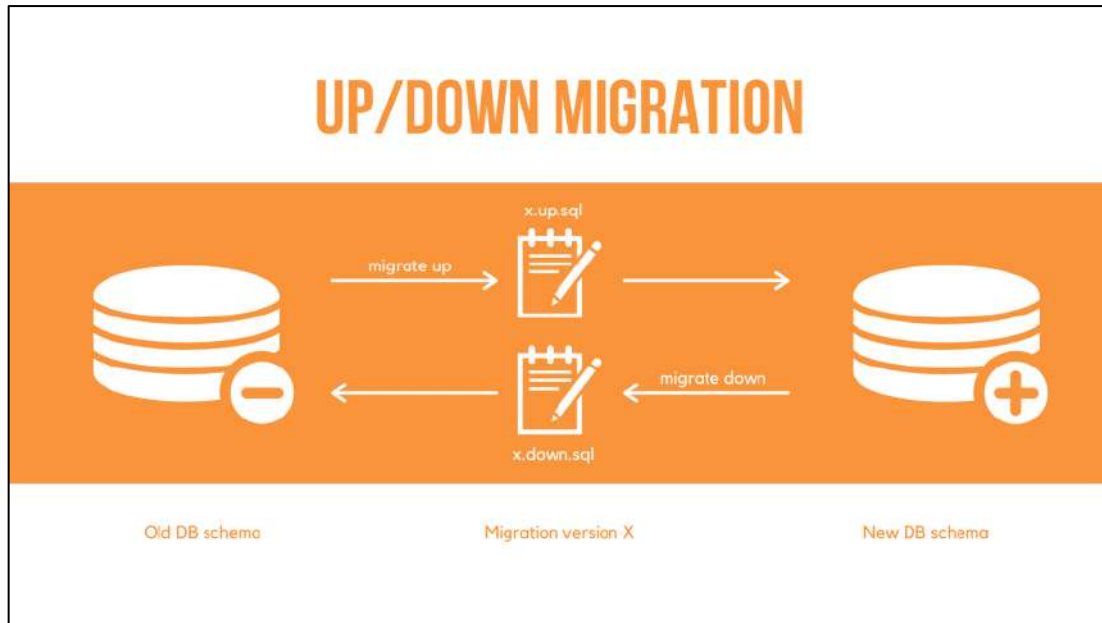- How can it be tested?
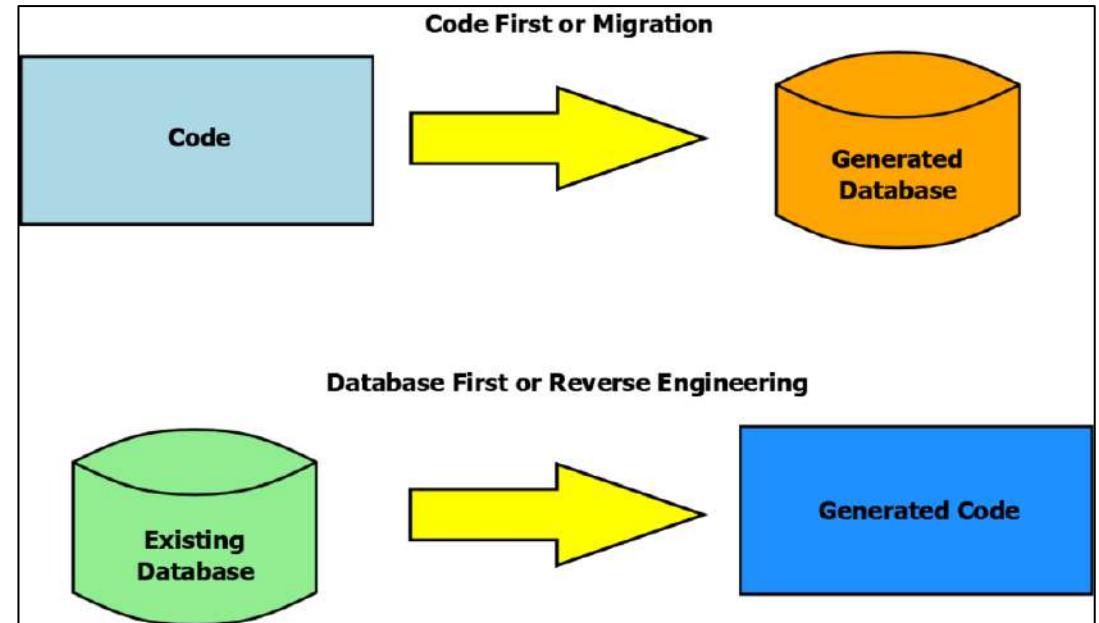

Fig. 8 Migrations

Fig. 9 Migrations Vs. Reverse Engg

Unit testing

- Validate that each unit of the software code performs as expected.
- Unit testing is first level of testing done before integration testing.

Fixtures

- Providing initial data to models
- Useful during testing logic, forms the baseline for running tests

Fig. 10 Fixtures

Fixtures

Fig. 11 Test environment

Work done by me

Applying schema migrations
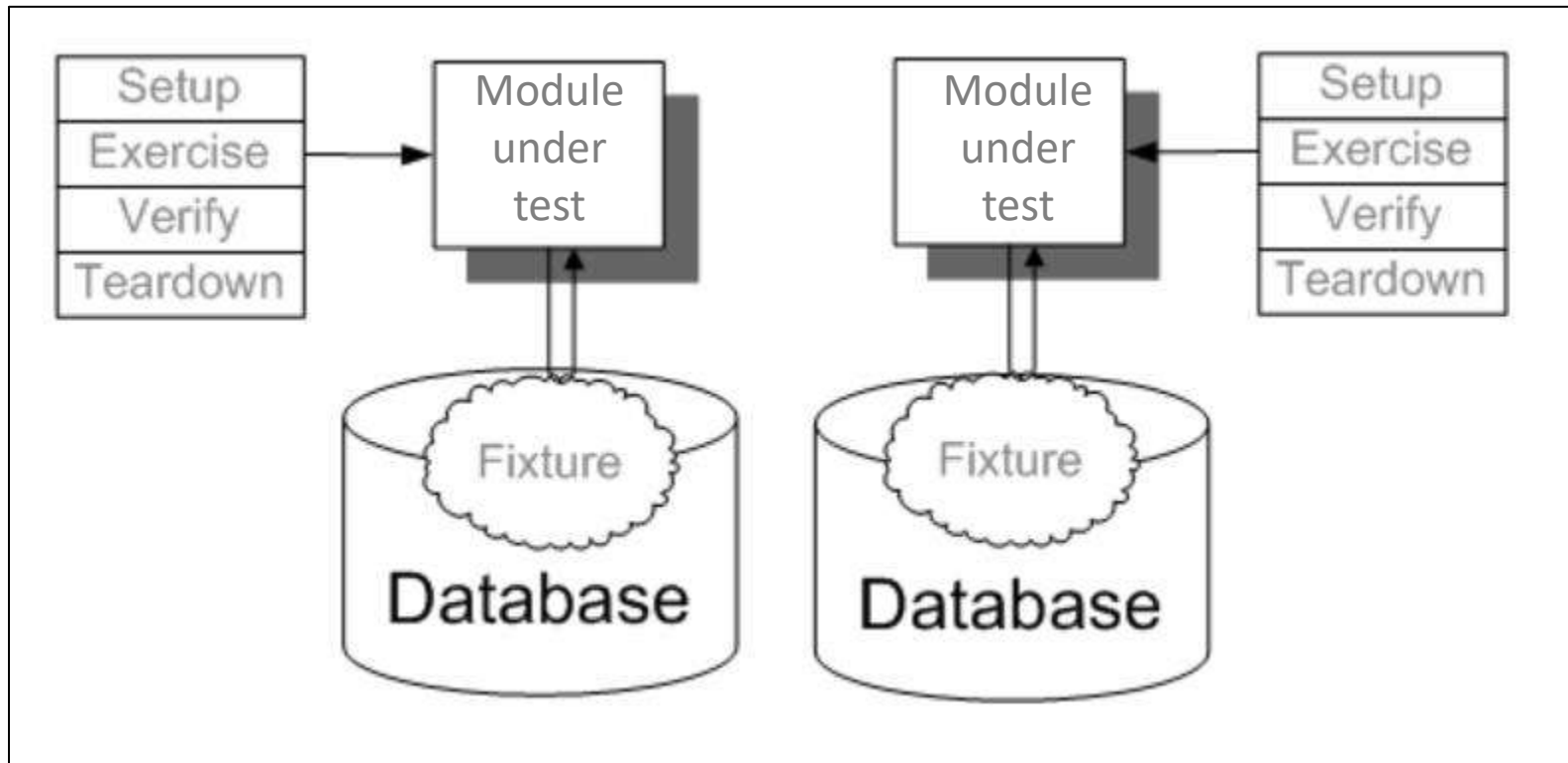
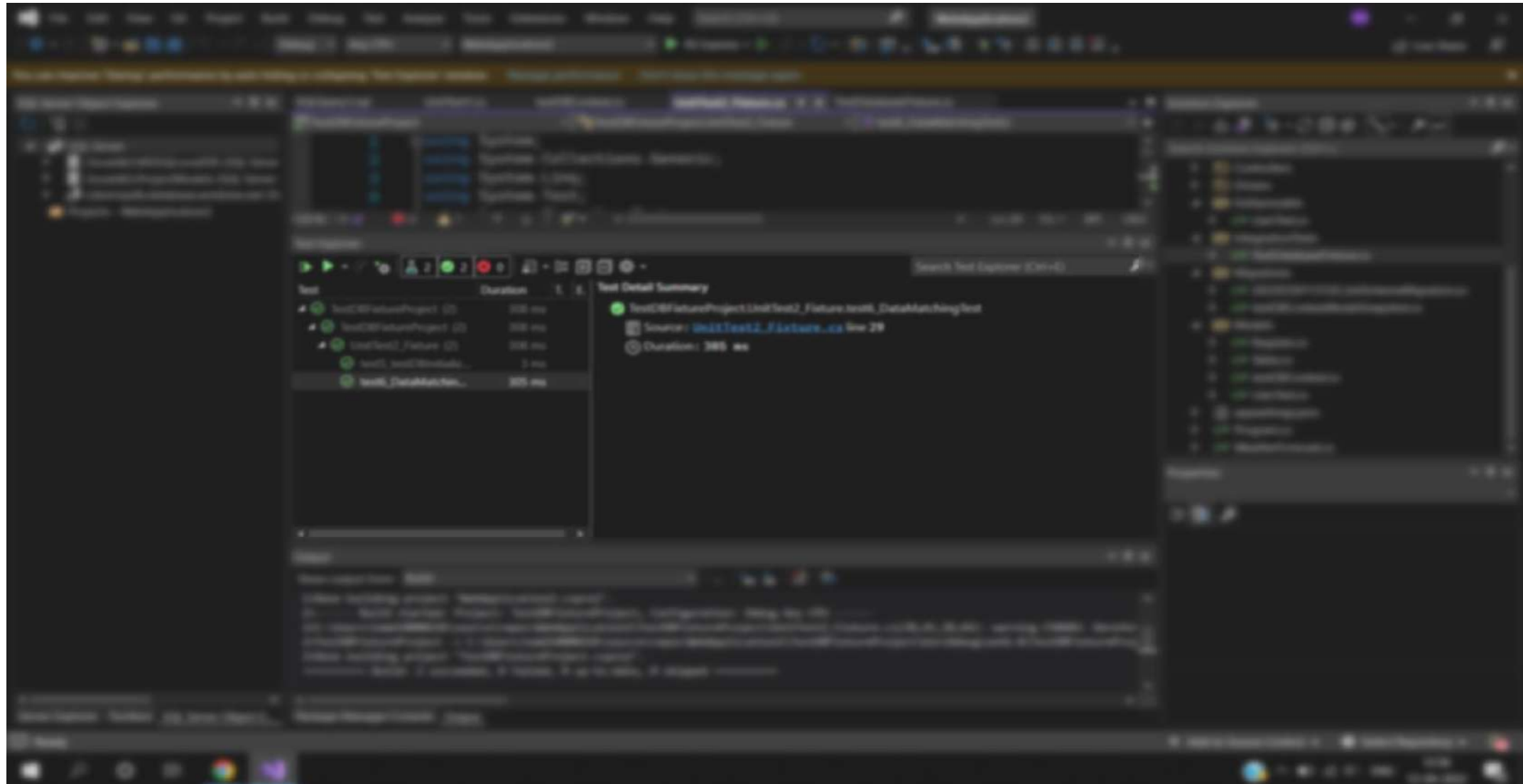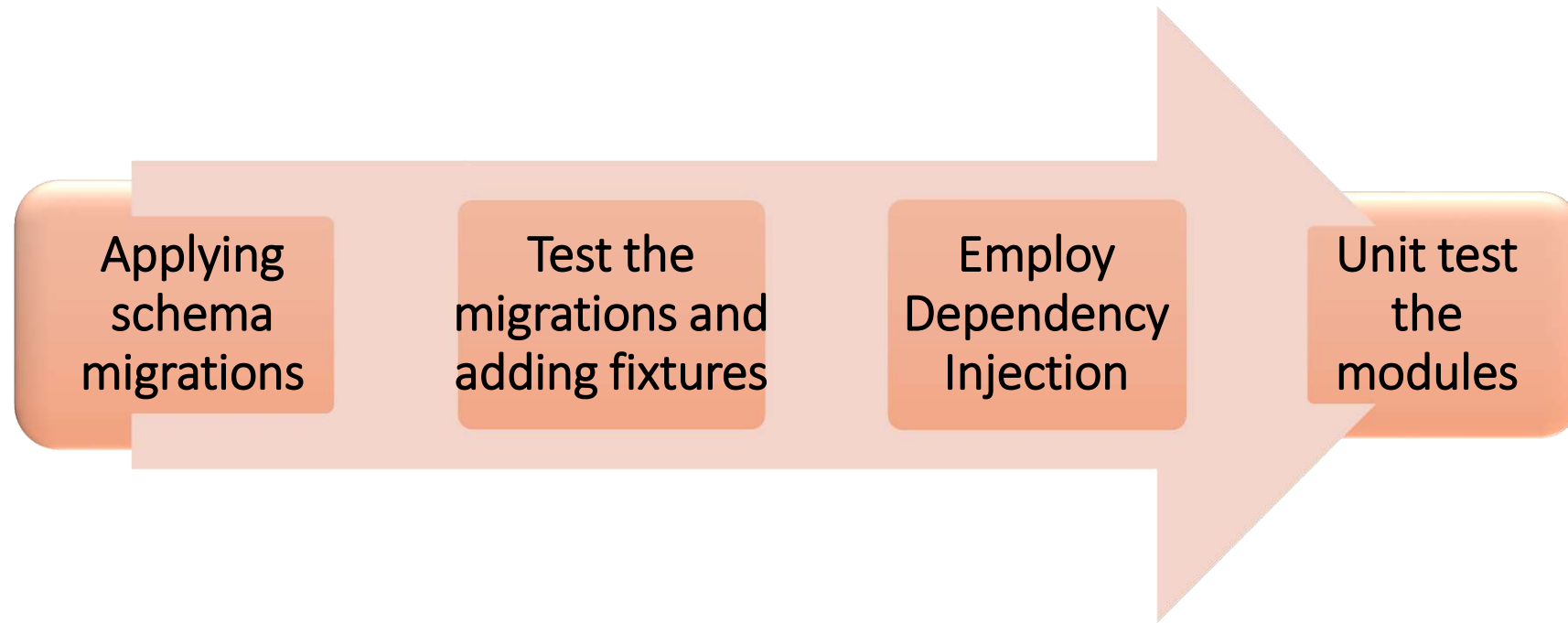Test the migrations and adding fixtures

Employ Dependency Injection

Unit test the modules

Dependency Injection

- Organizing the various modules working together
- Maintainability & Testability is enhanced

Fig. 12 Dependency Injection

Conclusion

Performing the initial operations (change schemas, adding migrations)

↓

Creating the test DB

↓

Applying the Migrations

↓

Adding test fixtures

↓

Disposing the DB after test
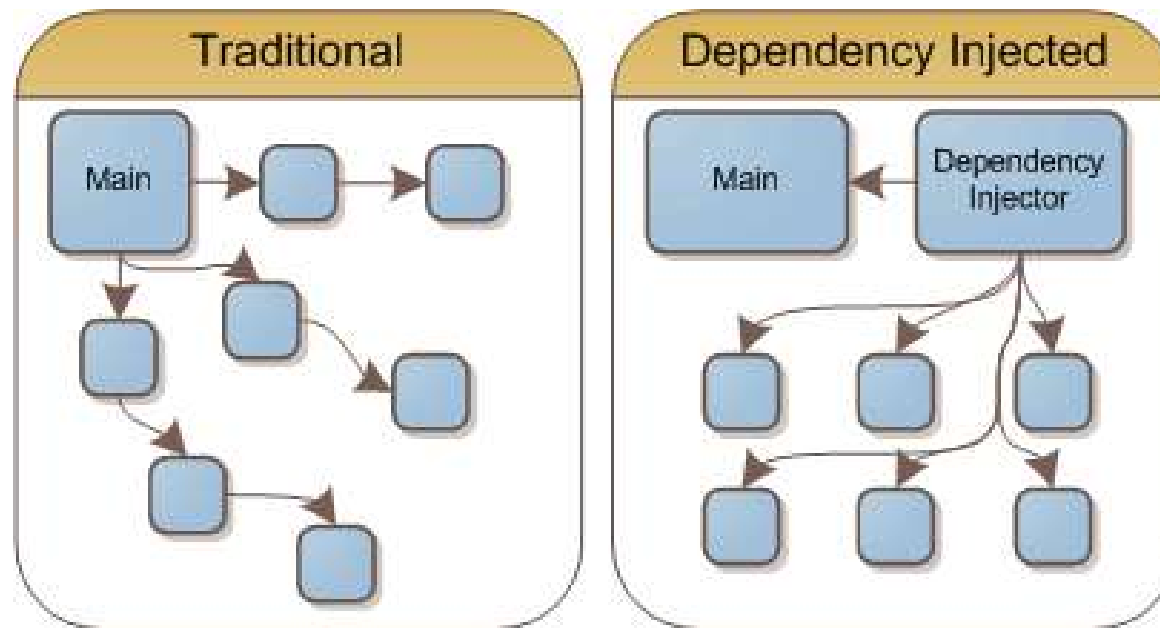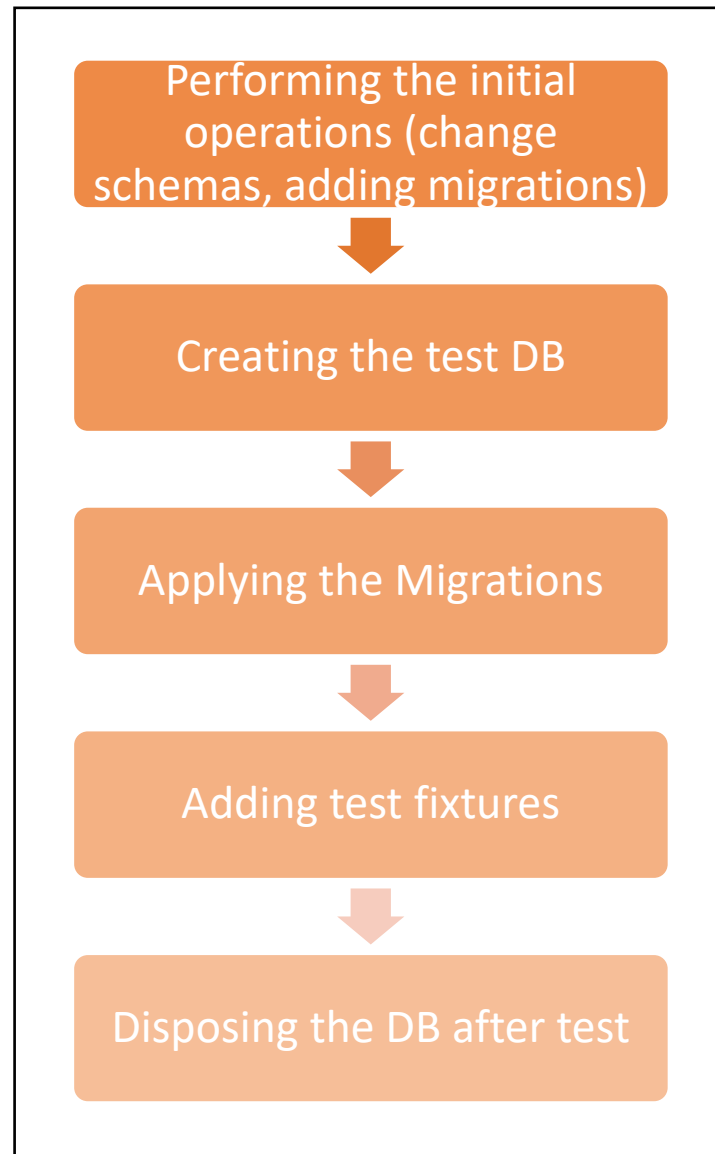
Thereby, this results in continuously integrate any database changes, as well as test it from unit tests of various modules.

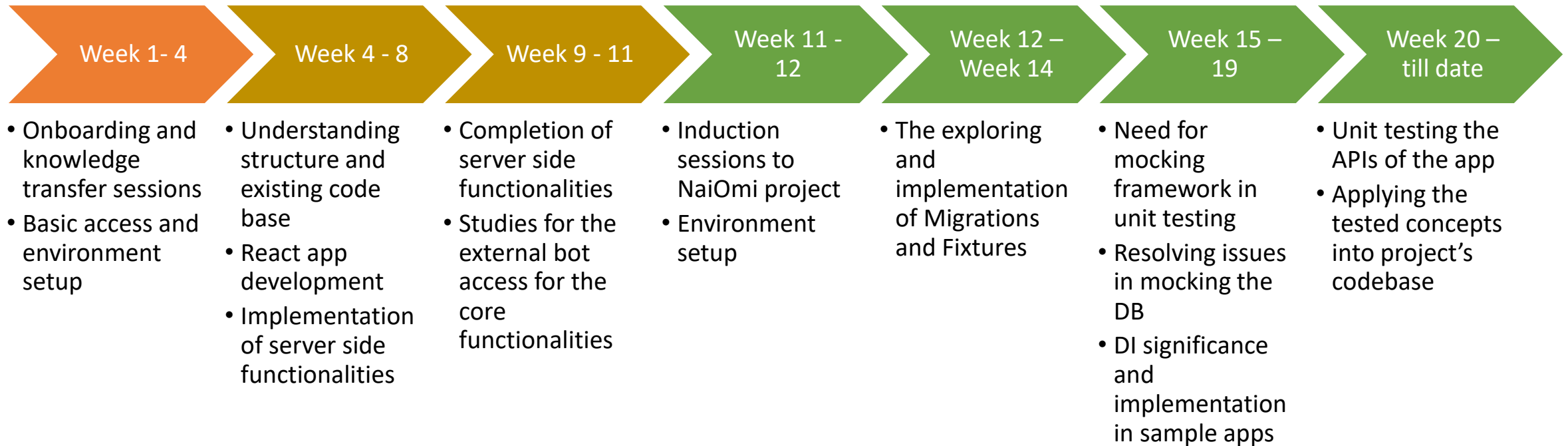This paves the way for CI/CD, to deliver changes more frequently and reliably.

Workflow

Fig. 13 Azure repos

Timeline

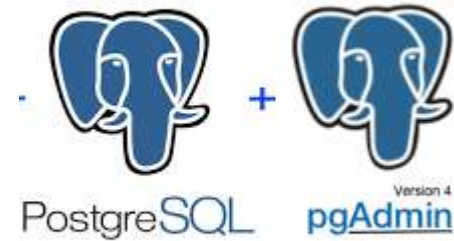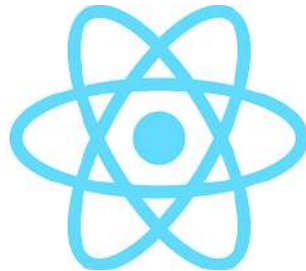| Week 1- 4	Week 4 - 8	Week 9 - 11	Week 11 - 12	Week 12 – Week 14	Week 15 – 19	Week 20 – till date
• Onboarding and knowledge transfer sessions • Basic access and environment setup	• Understanding structure and existing code base • React app development • Implementation of server side functionalities	• Completion of server side functionalities • Studies for the external bot access for the core functionalities	• Induction sessions to NaiOmi project • Environment setup	• The exploring and implementation of Migrations and Fixtures	• Need for mocking framework in unit testing • Resolving issues in mocking the DB • DI significance and implementation in sample apps	• Unit testing the APIs of the app • Applying the tested concepts into project's codebase

Work Environment

- <u>Hardware Specifications</u>
  - Processor : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
  - Installed Memory : 8.00 GB
  - System Type : 64-bit Operating System, x64-based processor
- <u>Software Specifications</u> : Windows 10 Pro

Tools and technologies used

Challenges faced

- Environment setup
- Understanding the components from existing code base

References

- https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli
- https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests?view=aspnetcore-6.0
- https://sweetcode.io/schema-changes-devops-including-database-development-model/
- https://www.youtube.com/watch?v=tNcWX9qPcCM&t=353s
- https://www.section.io/engineering-education/how-to-build-authentication-api-with-jwt-token-in-nodejs/

Thank You

Niveydhithaa R
17PW25

https://www.mobiusservices.com/
https://www.xtract.io/