

# WEEKLY INTERNSHIP REPORT

**FIRST WEEK REPORT**

*Submitted by*

**NAINEEL BHARATKUMAR SOYANTAR - ( 200280107003)**

***Student of***

**Bachelors of Engineering**

*in*

Computer Department

## Internal Guide

*Prof. Archana Gondaliya*

Internship (**3180701** )

**GUJARAT TECHNOLOGICAL UNIVERSITY**

**(Established under Gujarat Act No. 20 of 2007)**

**ગુજરાત ટેકનોલોજીકલ યુનનવનસિટી**

(ગુજરાત અિઅિનયમ ક્રમાાંકઃ ર૦/ર૦૦૭ દ્વારા સ્થાપના)

**Annexure 1 Enrollment no:**

**200280107003**



**NAME OF STUDENT: NAINEEL BHARATKUMAR SOYANTAR**

**DIARY OF THE WEEK: Dt: 13/02/2024 TO 19/02/2024 DEPARTMENT: COMPUTER ENGINEERING SEM: 8**

**NAME OF THE ORGANISATION: ZURU TECH INDIA PVT LTD. NAME OF THE PLANT/SECTION/DEPARTMENT: NAME OF OFFICER IN CHARGE OF THE PLANT/SECTION/DEPARTMENT:**

**DESCRIPTION OF THE WORK DONE IN BRIEF**

In this week, we started understanding the design patterns of production-grade applications. We understood the basics of such applications and the underlying folder structure also. Then we moved on to understanding database migrations and how to manage databases in such large-scale production environments. We learned about `dbmate` which is used to generate migration files and perform DDL operations on the databases with ROLLBACK facilities as well.

We also learned about SwaggerUI and it’s learned why documentation is highly required in developing a complete application as it is useful for other developers as well as the users.

Then to apply what we learned we documented the earlier developed Library management system.

Then we moved on to Ngrok to learn about forwarding a localhost port over the internet so that while developing any kind of testing required can be done. Also learned about Docker for containerization of the current application and PM2.

Dived deep into nginx to learn about its usage as reverse proxy, load balancer and rate limiter. Also, developed a full-fledged reverse proxy using nginx and pm2 which is a process manager.

**GUJARAT TECHNOLOGICAL UNIVERSITY**

**(Established under Gujarat Act No. 20 of 2007)**

**ગુજરાત ટેકનોલોજીકલ યુનનવનસિટી**

(ગુજરાત અિઅિનયમ ક્રમાાંકઃ ર૦/ર૦૦૭ દ્વારા સ્થાપના)



**TOTAL HOURS :**

**SIGNATURE OF STUDENT**

**The above entries are correct and the grading of work done by Trainee is EXCELLENT / VERY GOOD / GOOD / FAIR / BELOW AVERAGE / POOR**

**Signature of Faculty Mentor**

**Signature of officer-in-charge of Dept. / Section / Plant**

**Date:**

**Date:**

**Grading of Work, for trainee may be given depending upon your judgement about**

**his Punctuality, Regularity, Sincerity, Interest taken, Work done etc.**

**STUDENT’S DAILY DIARY/ LOG**

**Name of Student: NAINEEL SOYANTAR**

|  |  |  |  |
| --- | --- | --- | --- |
| **WEEK 5** | **13 FEBRUARY 2024** | **Page No** | **Sign of Industry Supervisor** |
|  | Design patterns of production-grade applications |  |  |
|  | Database migrations using dbMate |  |  |
|  | **14 FEBRUARY 2024** |  |  |
|  | SwaggerUI, SwaggerJSDoc, Definitions File |  |  |
|  | Generated Swagger UI for Library Management System |  |  |
|  | **15 FEBRUARY 2024** |  |  |
|  | Ngrok, Docker, PM2 |  |  |
|  | **16 FEBRUARY 2024** |  |  |
|  | Started with Nginx and its basics |  |  |
|  | **19 FEBRUARY 2024** |  |  |
|  | Built reverse proxy using nginx, docker and pm2 |  |  |

**WEEK 5: 13 FEBRUARY 2024**

* **Design pattern in production-grade applications:**

The design pattern in production-grade applications is built in such a way that the usability while developing the application is extremely high and overhead for managing the application is low.

The folder structure is –

A screenshot of a computer

Description automatically generated

Here, all the additional files which help in development are mentioned in the root folder. Like, configuration files, database and migration files, documentation files and log files.

The source code of the application which is stored inside the `src` folder. It contains error handling files, middleware files and module files.

Module files are the files that handle the logic of a particular module like Student module in School management system.

General structure of a particular module is –

* routes file – handling the routes of that module.
* Controller file – which handles the request for a particular route and diverts it to a particular service which is required for the business logic to happen.
* **A screenshot of a computer program

  Description automatically generated**Services file – The main business logic of a module’s all the routes is mentioned in the services file.
* DAL file – DAL stands for Data Access Layer and this file is where all the data is accessed. Services call the appropriate DAL function and DAL file fetches the data from the database.
* Schema file – Schema file is where all the expected types of the request and the responses are there. It is generally made using Joi.
* **Database migrations using dbMate:**

DBmate is a lightweight and easy-to-use tool for managing database migrations. It simplifies the process of applying incremental changes to your database schema and data.

**Example-**

dbmate new create\_user\_profiles\_table – this is used to create a new migration file.

Then you can write your DDL statements in the migration file generated by dbmate. You can use the command `dbmate up` to run the DDL defined and `dbmate down` to rollback the defined DDL.

**WEEK 5: 14 FEBRUARY 2024**

* **SwaggerUI, SwaggerJSDoc:**

SwaggerUI is a user-friendly interface that allows developers to visualize and interact with API documentation generated by Swagger or OpenAPI Specification (OAS). It provides a web-based interface where users can explore the API endpoints, view request and response parameters, and even make test requests directly from the UI.

SwaggerUI can be easily integrated into web applications and serves as a helpful tool for both developers and API consumers to understand and test APIs.

SwaggerJSDoc is a tool used in Node.js applications to automatically generate Swagger or OpenAPI Specification (OAS) documentation based on JSDoc comments in the source code.By adding JSDoc comments to API endpoint functions and their parameters, SwaggerJSDoc parses these comments and generates a corresponding Swagger/OAS document in JSON format.

* **A screenshot of a computer

  Description automatically generatedGenerated SwaggerUI for Library Management Application:**

**WEEK 5: 15 FEBRUARY 2024**

* **Ngrok, Docker, PM2**

**Ngrok:**

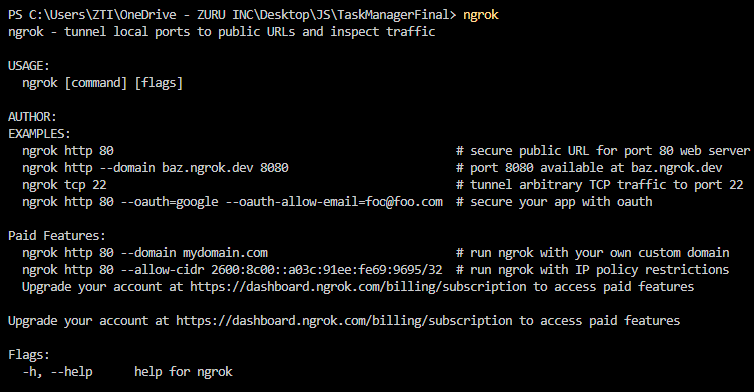
* Ngrok is a tool that exposes local servers behind NATs and firewalls to the public internet over secure tunnels.
* It allows developers to create secure, temporary URLs (with HTTPS support) that point to their local development environment, making it easy to share work-in-progress web applications with colleagues or clients for testing and feedback.
* Ngrok supports various protocols such as HTTP, HTTPS, TCP, and UDP, making it versatile for different types of applications.

**Docker**:

* Docker is a platform for developing, shipping, and running applications in containers.
* Containers are lightweight, portable, and self-sufficient environments that encapsulate an application and its dependencies, enabling consistent deployment across different environments (development, testing, production, etc.).
* Docker provides tools and APIs for building, managing, and orchestrating containers, making it easier to package applications and deploy them as isolated units that can run anywhere.

**PM2 (Process Manager 2):**

* PM2 is a production process manager for Node.js applications.
* It simplifies the management of Node.js applications in production environments by providing features such as automatic application restarts, log management, process monitoring, and clustering for load balancing.
* PM2 can be used to manage multiple Node.js applications on a single server, improving efficiency and reliability in production deployments.



A screenshot of a computer program

Description automatically generated

**Week 5: 16 FEBRUARY 2024**

* **Nginx and its basics**

Nginx is a high-performance web server and reverse proxy known for its efficiency and scalability. It serves web content, handles HTTP requests, and can act as a reverse proxy and load balancer. Nginx is lightweight, resource-efficient, and capable of handling thousands of concurrent connections. It supports features like reverse proxy caching, load balancing, SSL termination, and WebSockets, making it a popular choice for serving web applications and improving performance and reliability.

Example – All the basic commands of nginx used in nginx.conf file

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

# Other HTTP configurations...

}

server {

listen 80;

server\_name example.com;

location / {

root /usr/share/nginx/html;

index index.html index.htm;

}

# Other server configurations...

}

error\_log /var/log/nginx/error.log;

access\_log /var/log/nginx/access.log main;

**WEEK 5 : 19 FEBRUARY 2024**

* **Built Reverse Proxy using Nginx, PM2–**

# Context - Directives

events{

}

http{

include mime.types;

upstream node\_starter{

server 127.0.0.1:2000;

server 127.0.0.1:2001;

server 127.0.0.1:2002;

}

server{

listen 8080;

root "C:\\Users\\ZTI\\OneDrive - ZURU INC\\Desktop\\NGINX\\website";

# below location logs the express server running on port 2000, 2001, 2002

location / {

proxy\_pass http://node\_starter; #this will automatically load balance the requests between the node\_starter servers mentioned in the upstream block

}

# below locations logs the static files in the website folder

location /first {

# /first is implicitly added to the end of the root structure

root "C:\\Users\\ZTI\\OneDrive - ZURU INC\\Desktop\\NGINX\\website";

}

location /second {

# /alias is not implicitly added to the end of the root structure

alias "C:\\Users\\ZTI\\OneDrive - ZURU INC\\Desktop\\NGINX\\website\\first";

}

location /third {

# /third is implicitly added to the end of the root structure

root "C:\\Users\\ZTI\\OneDrive - ZURU INC\\Desktop\\NGINX\\website";

try\_files /third/index.html /third/index.htm /third/misc.html /third/third.htm =403;

}

}

}