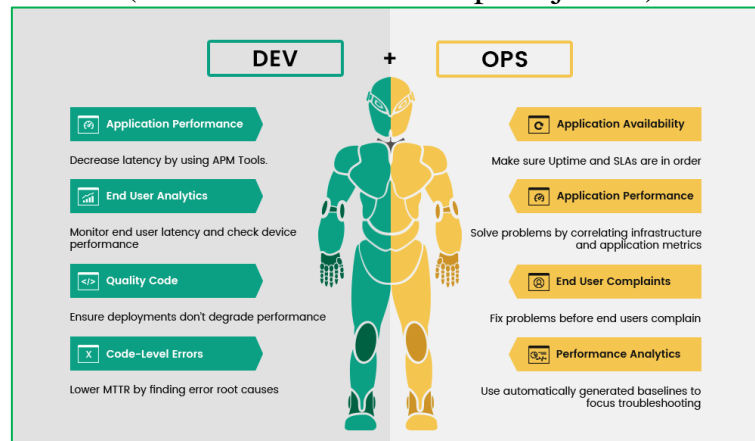


A Project Report On "DevBlog(Developer's Blogging Site) Website Using DevOps Technology"

(CE244-Software Group Project-II)



Prepared by
Dhyey Joshi(19DCS043)
Hirva Joshi(19DCS045)
Khushi Chauhan(19DCS017)

Under the Supervision of
Prof. Chandrashekhar Pawar

Submitted to
Charotar University of Science & Technology (CHARUSAT) for
the Partial Fulfillment of the Requirements for the Degree of
Bachelor of Technology (B.Tech.)
in Computer Science And Engineering (CSE)
for 4rd semester B.Tech

Submitted at



Devang Patel Institute of Advance Technology and Research (DEPSTAR)
Faculty of Technology & Engineering (FTE), CHARUSAT At:
Changa, Dist: Anand, Pin: 388421.

October, 2020



CERTIFICATE

This is to certify that the report entitled **“DevBlog(Developer’s Blogging Site) Website Using DevOps Technology”** is a bonafied work carried out by Ms. Hirva Joshi(19DCS045) under the guidance and supervision of **Assistant Prof. Chandrashekhar Pawar** for the subject CE244 (For CSE)- **Software Group Project-II** (CE/CSE/IT) of 4rd Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Chandrashekhar Pawar
Branch Name(CSE)
DEPSTAR, Changa, Gujarat.

Dr. Amit Ganatra
Principal, DEPSTAR
Dean, FTE
CHARUSAT, Changa, Gujarat.



CERTIFICATE

This is to certify that the report entitled “**DevBlog(Developer’s Blogging Site) Website Using DevOps Technology**” is a bonafied work carried out by Mr. Dhyey Joshi(19DCS043) under the guidance and supervision of **Assistant Prof. Chandrashekhar Pawar** for the subject CE244 (For CSE)- **Software Group Project-II** (CE/CSE/IT) of 4rd Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Chandrashekhar Pawar
Branch Name(CSE)
DEPSTAR, Changa, Gujarat.

Dr. Amit Ganatra
Principal, DEPSTAR
Dean, FTE
CHARUSAT, Changa, Gujarat.



CERTIFICATE

This is to certify that the report entitled “**DevBlog(Developer’s Blogging Site) Website Using DevOps Technology**” is a bonafied work carried out by Ms. Khushi Chauhan(19DCS017) under the guidance and supervision of **Assistant Prof. Chandrashekhar Pawar** for the subject CE244 (For CSE)- **Software Group Project-II** (CE/CSE/IT) of 4rd Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Chandrashekhar Pawar
Branch Name(CSE)
DEPSTAR, Changa, Gujarat.

Dr. Amit Ganatra
Principal, DEPSTAR
Dean, FTE
CHARUSAT, Changa, Gujarat.

ACKNOWLEDGEMENT

We, the developer of a “DevBlog(Developer’s Blogging Site) Website Using DevOps Technology”, with immense pleasure and commitment would like to present the project assignment. The development of this project has given us wide opportunity to think, implement and interact with various aspects of management skills as well as the new emerging technologies.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. I hereby avail this opportunity to express my gratitude to number of people who helped us directly or indirectly through their work in developing the project.

We express deep sense of gratitude towards our Head of the CSE Department, Prof. Parth Goel and project guides Prof. Priyal Vaghela and Prof. Chandrashekhar Pawar for the support during the whole session of study and development. It is because of them, that we was prompted to do hard work, adopting new technologies.

Thanks,
Dhyey Joshi
Hirva Joshi
Khushi Chauhan

ABSTRACT

This is a software-powered world where every programmer is a wizard and can do wonders with the magic wand of codes . It is no news that blogging is an authoritative source to share your ideas, connect, and inspire people.

What if programmers start blogging?

Because a programmer writes codes for the users just as writers write stories to inspire and engage their users. But think of the magic that will happen when a developer integrates his experience of coding and participates and benefits the users with his words?

Whoa!

It would be a jackpot for both readers and developers.

DEV is a community of software developers getting together to help one another out. The software industry relies on collaboration and networked learning. We provide a place for that to happen. Our application is open source, meaning you can inspect every little detail of the code, or chip in yourself! We are working to make our platform available for anyone to stand up similar communities in any niche or passion. We believe in transparency and adding value to the ecosystem. We hope you poke around and like what you see!

Keywords: Html , CSS , JavaScript , Docker , Kubernetes.

Contents

1	Project Definition	9
2	Project Description	13
3	Software and Hardware Requirements	19
3.1	Software Requirement.....	19
3.2	Hardware Requirement	19
4	Major Functionality	20
5	System Flow Chart	22
6	Screenshots of Project	23
7	Limitations of the Project	27
8	Project Outcomes	28
9	Conclusion and Future Scope	34
9.1	Conclusion34
9.2	Future Scope34
10	References	36

List of Figures

1.1 Virtualization	12
1.2 Containerization	12
2.1 Website Architecture.....	14
2.2 Docker Image – shanky25/node-web-app	14
2.3 Yaml Code for both .yaml files	15
2.4 Yaml Code for both .yaml files	16
2.5 Managing Load On Webserver Using Kubernetes.....	18
2.6 ngrok-create private tunnel to public world (Internet)	18
5.1 System Flow Chart	22
6.1 Index.html	23
6.2 About.html	23
6.3 Contact.html.....	24
6.4 Searchingblog.html.....	24
6.5 Server.js	25
6.6 Config.json.....	25
6.7 User.service.js.....	26
6.8 Db.js.....	26
7.1 DevOps Phases	27
8.1 Login Page	28
8.2 Registration Page	28
8.3 About Us Page	29
8.4 Contact Page	29
8.5 Authenticate - Username.Password.....	30
8.6 Authenticate - Username	30
8.7 Index Page	31
8.8 MySQL Database – Data Dictionary.....	31
8.9 Password Validation	32
8.10 Searching Blog Page.....	32
8.11 Click on any programming language displayed on screen to view related blogs.....	33

Chapter 1

Project Definition

A web project is the process of developing and creating a Web site, activities in a network which are aimed at a pre-defined goal. The network can be both accessible for everyone, as in the Internet, or only for certain people, as an intranet. The goal of Web projects is the transfer of static and dynamic content - both directly to end users, as well as indirectly through means of various kinds of interfaces. Web projects are based on TCP/IP (Transfer Control Protocol/Internet Protocol) technology and concern the transfer of static and dynamic content.

A Web project involves many aspects, including programming and the accompanying software development, Web business, Web server and network administration, hosting, graphics/design, the development and administration of databases, construction of interfaces, project management and quality assurance, search engine optimization, the maintenance of data in content management/editing systems and much more.

Programming for a web project may be accomplished using one or more markup languages (such as HTML or XML), scripting languages (JavaScript, Perl, PHP for example) or even more complex programming languages (such as C/C++/C# or Java).

When hosting a Web project, the primary objectives include the provision of the necessary hardware and software infrastructure, and an assurance that the highest possible levels of availability and reliability are offered. Graphic/Web design for Web projects must offer a high quality of use for persons interacting

with the website .Agile project management methods (e.g. Scrum) are used for the management of modern Web projects in order to respond to changes in customer requirements and constraints as the project progresses. The project manager is responsible for the efficient and result-oriented programming of the Web project.

In our project we have used HTML , CSS , JAVA-SCRIPT for frontend And for backend we have used NODEJS for building server with MySQL Database Connection. This was just first part of the project i.e hosting our website in local machine(Windows).

In our second part, we have make used Virtualization and Containerization technology for hosting & deploying purpose.

BASIC TERMINOLOGY USED IN DEVOP'S TECHNOLOGY

1. WHAT IS DEVOPS TECHNOLOGY ?

Ans :DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

2. WHY DEVOPS IS FAST GROWING TECHNOLOGY ?

Ans :It have shorter Development Cycles, Faster Innovation , Reduced Deployment Failures, Rollbacks, and Time to Recover , Improved Communication and Collaboration , Increased Efficiencies , Reduced Costs and IT Headcount.

3. WHAT IS VIRTUALIZATION TECHNOLOGY & VIRTUAL MACHINE ?

Ans :Virtualization is the process of creating a software-based, or virtual, representation of something, such as virtual applications, servers, storage and networks. It is the single most effective way to reduce IT expenses while

boosting efficiency and agility for all size businesses.

4. WHAT IS CONTAINERIZATION?

Ans : Application containerization is an OS-level virtualization method used to deploy and run distributed applications without launching an entire virtual machine (VM) for each app. ... Containers work on bare-metal systems, cloud instances and virtual machines, across Linux and select Windows and Mac OSes.

5. WHAT IS DOCKER CONTAINER?

Ans : Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers.

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

6. WHY DOCKER ?

Ans : Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.

7. WHAT IS KUBERNETES ?

Ans : Kubernetes is an open-source container-orchestration system for automating computer application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.

8. WHAT IS KUBECTL ?

Ans :The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters.

All detailed explanation is in next chapter...

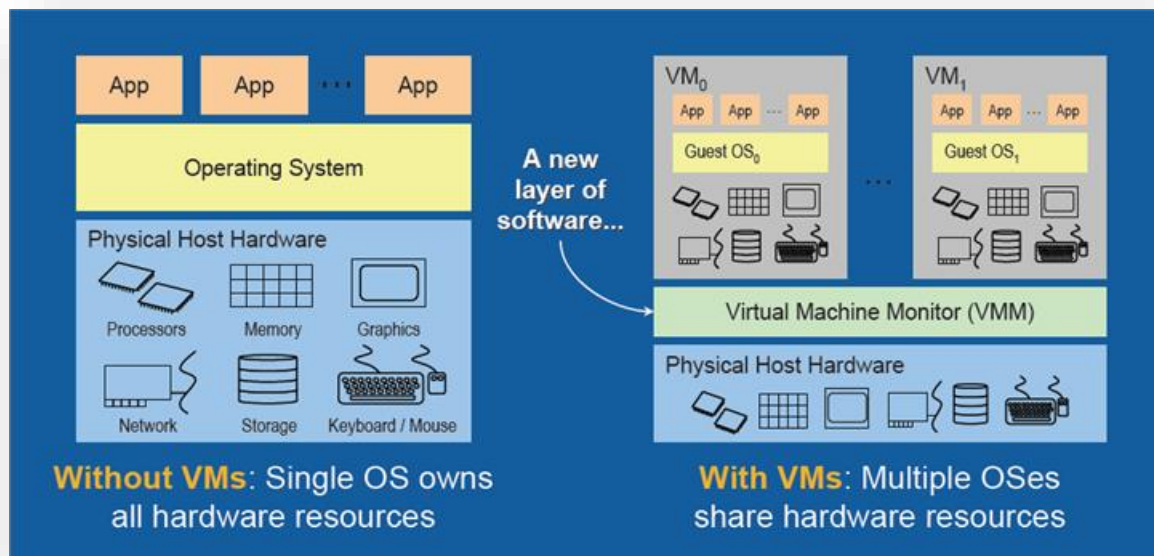


Figure (1.1) Virtualization

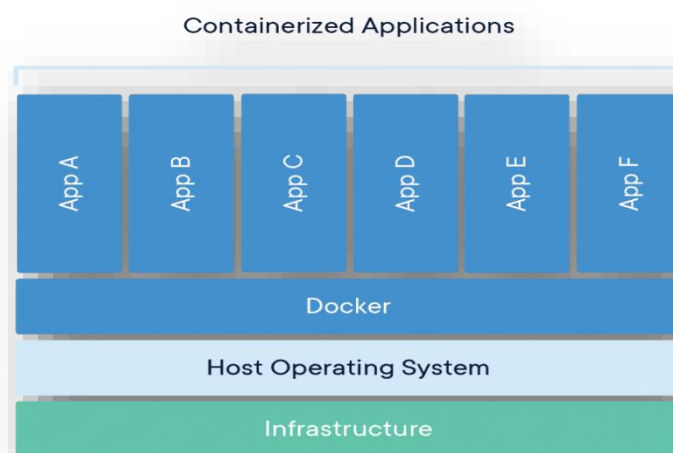


Figure (1.2) Containerization

Chapter 2

Project Description

For second part of project do following steps :

Step-01

- Let Windows be the Client and RedHat Linux be the Server.
- Windows with specifications:
 - PRIVATE IP - IPv4 : 192.168.29.164
 - HTTP : 8080
 - PORT : 80
 - TCP/IP PROTOCOL
- Virtual Machine Contain RedHat Linux installed so working on the top of virtual machine i.e called as Virtualization.
- Go Inside RedHat Linux – Install Docker which work on Containerization Engine.
- RedHat Linux with specifications:
 - PRIVATE IP - IPv4 : _____.____.____.____
 - LISTENING PORT : 8080
 - TCP/IP PROTOCOL
 - CONFIGURE : APACHE WEB-SERVER
 - PLACE YOUR WEBSITE : WWW.DEVBLOG.COM
- Create Image of your Docker Container.

- Push your image to Docker Hub – <https://hub.docker.com/>

`docker push shanky25/node-web-app`

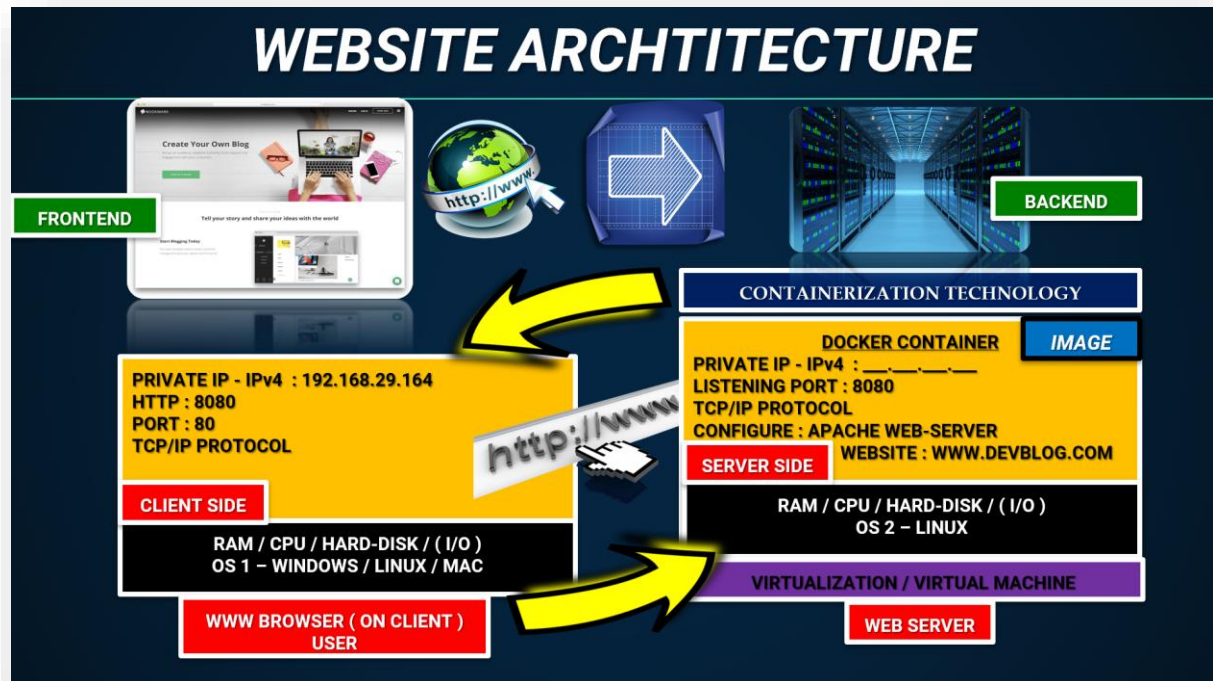


Figure (2.1) Website Architecture

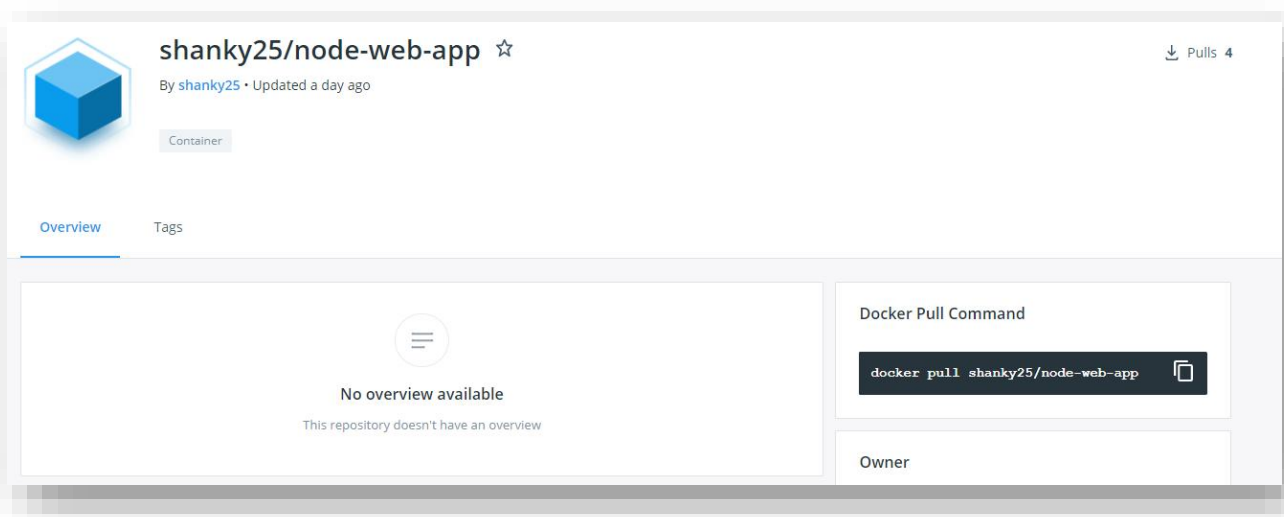


Figure (2.2) Docker Image – shanky25/node-web-app

Step-02

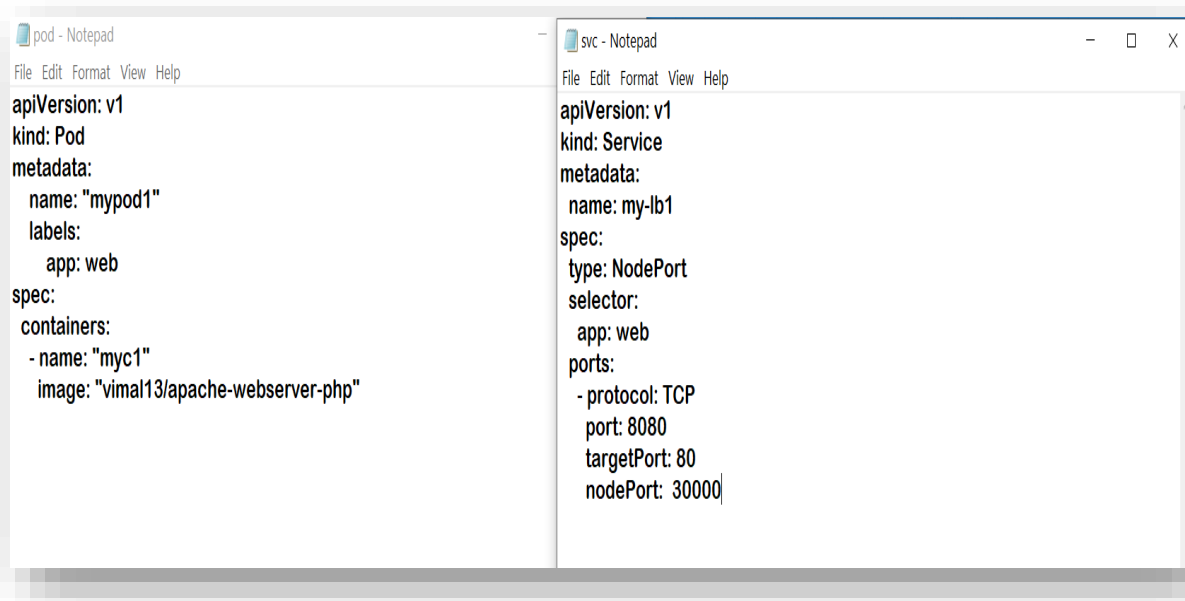
- So most of the part of project is done , now just use your image and deploy it on Kubernetes Server . To do so you have create Kubernetes Cluster lets begin..
- Do this steps in same order :
 - ❖ Open CMD
 - ❖ minikube status
 - ❖ minikube start (if not started)
 - ❖ kubectl get svc (TO check of any service is there or not , if not then lets create one service for load balancer)
 - ❖ Lets first create load balancer inside Kubernetes Cluster :

notepad svc.yml

notepad Healthlb.yml

(This will create one pod with apache image which will show your Ips of operating system running as a backend webserver)

Figure (2.3) Yaml Code for both .yaml files



Step-03

- Now, lets create MySQL in one pod:

notepad pod.yml

(Yaml code with all configuration for setting connection with my node application)

notepad s.yml

(It is known as secret file where all the critical information's are stored in encoded form . so even if someone get my files he will not able to see my password , username & database name.)

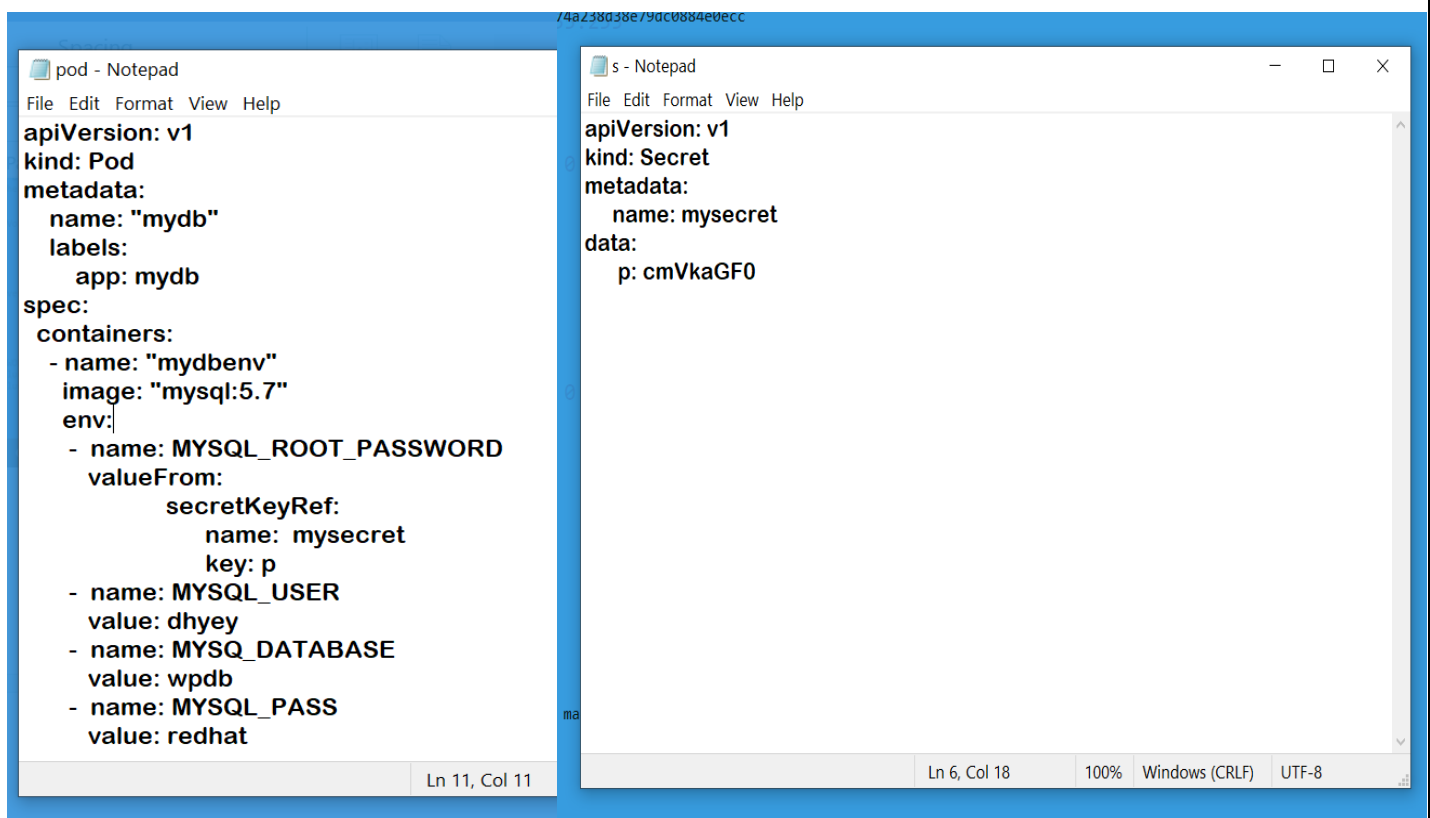


Figure (2.4) Yaml Code for both .yaml files

Step-04

- Now, lets host our website in one pod and then deploy it on fly :

```
Kubectrl run devblogweb -- image = shanky25/node-web-app --  
type=Nodeport
```

```
Kubectrl create deployment devblogweb -- image = shanky25/node-  
web-app
```

```
Kubectrl scale deployment devblogweb -- replicas = 20
```

(If load increases on our website then Kubernetes will automatically launch one more webserver with all the configuration and host it with the same Ip : Port. Here we have used **20 operating systems as a backend webserver** , Power of Kubernetes is like if incase due to some reason one of our pod goes down .. then it will automatically launch new webserver with same configuration on fly!!)

```
Kubectrl expose deployments devblogweb -- port = 80 -- type =  
Nodeport
```

(Using the concept of PAT – Port Address Translation)

```
minikube service devblogweb -- url
```

<http://192.168.99.100:30849/> - Our website common Public - URL

```
minikube service devblog -- url
```

<http://192.168.99.100:32344/> - This is to check load balancer

- If you want to expose your port to the internet then use ngrok for port forwarding :

```
ngrok http http://192.168.99.100:30849/
```

(minimum 40 connection per minute)

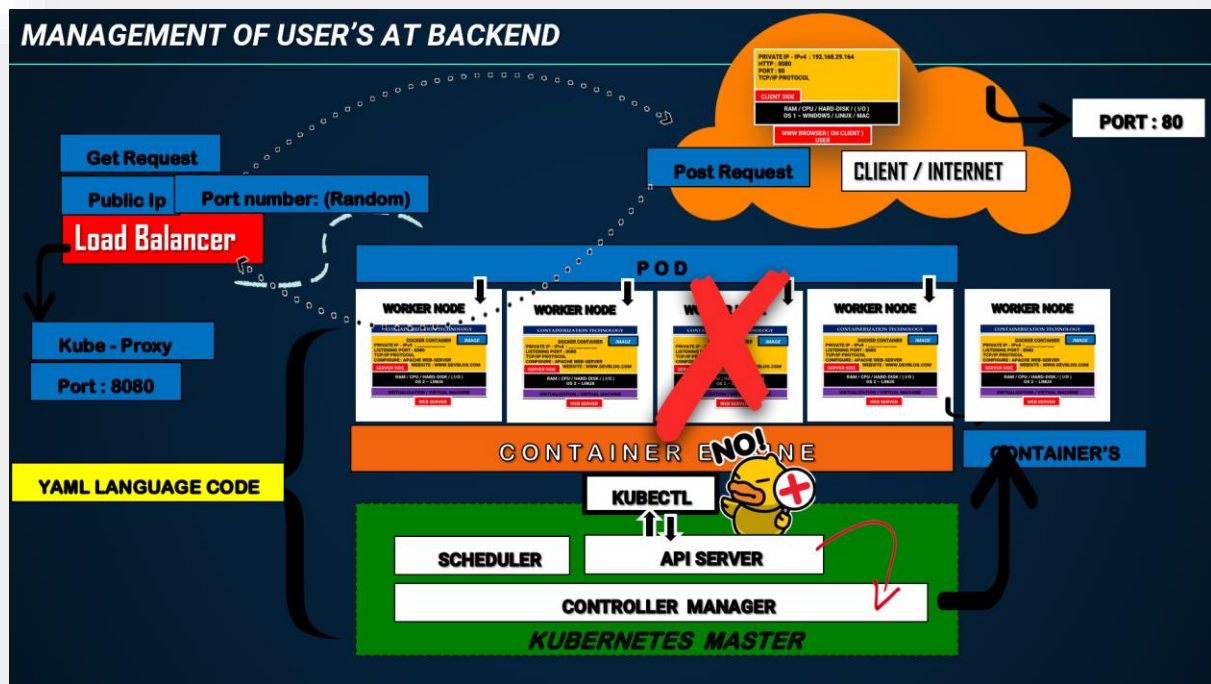


Figure (2.5) Managing Load On Webserver Using Kubernetes

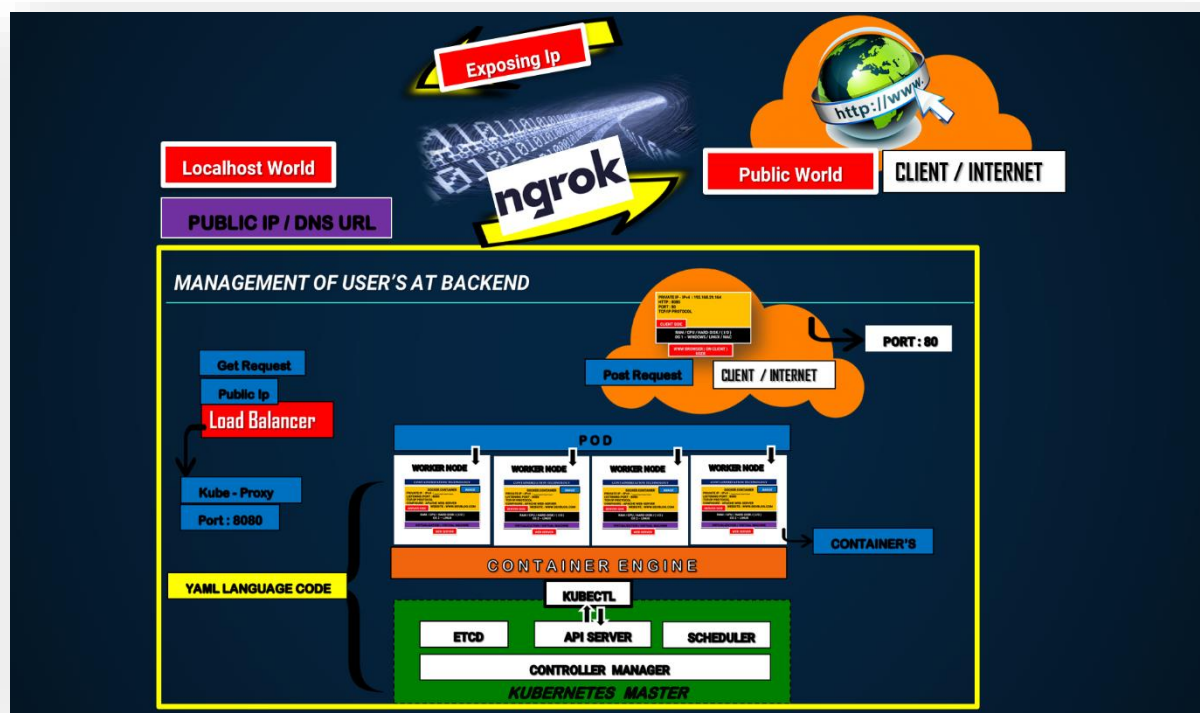


Figure (2.6) ngrok-create private tunnel to public world (Internet)

Chapter 3

Software and Hardware Requirement

3.1 Software Requirement

Following are the software requirement necessary of the project:

1. Html , JavaScript , CSS , Nodejs , SQL , YAML programming languages.
2. For Client Side:
 1. Operating System – Windows / Linux / Mac .
 2. Internet Connectivity
 3. If facing any connectivity issue – disable your system firewall .
 - 4.Min 2GB RAM / CPU / Hard – Disk.
3. Install Kubernetes Environment + Load Balancer + configuration
For Server Side:
 1. Virtual Machine : Operating System – Redhat (RHEL v8) Linux .
 2. No need of internet connectivity (For Security Purpose – Isolated System)
 - 3.”Disable” your firewall – Rule : only allow Tcp/Ip.
 - 4.As per Load on this server (RAM / CPU / Hard – Disk).
 - 5.Install httpd : Apache web server + All configuration .

3.2 Hardware Requirement

Following are the hardware requirement that is most important for the project:

1. Fluently working Laptops.
2. RAM minimum 16Gb.
3. Minimum Intel CORE i5 8th Gen Processor.

Chapter 4

Major Functionalities

As most modern software developers can attest, containers have provided us with dramatically more flexibility for running cloud-native applications on physical and virtual infrastructure. Containers package up the services comprising an application and make them portable across different compute environments, for both dev/test and production use. With containers, it's easy to quickly ramp application instances to match spikes in demand. And because containers draw on resources of the host OS, they are much lighter weight than virtual machines. This means containers make highly efficient use of the underlying server infrastructure.

So far so good. But though the container runtime APIs are well suited to managing individual containers, they're woefully inadequate when it comes to managing applications that might comprise hundreds of containers spread across multiple hosts. Containers need to be managed and connected to the outside world for tasks such as scheduling, load balancing, and distribution, and this is where a container orchestration tool like Kubernetes comes into its own. An open source system for deploying, scaling, and managing containerized applications, Kubernetes handles the work of scheduling containers onto a compute cluster and manages the workloads to ensure they run as the user intended. Instead of bolting on operations as an afterthought, Kubernetes brings software development and operations together by design. By using declarative, infrastructure-agnostic constructs to describe how applications are composed, how they interact, and how they are managed, Kubernetes enables an order-of-magnitude increase in operability of modern software systems.

Kubernetes was built by Google based on its own experience running containers in production, and it surely owes much of its success to Google's involvement.

Google has some of the most talented software developers on the planet, and it runs some of the largest software services by scale. This combination ensured that Kubernetes would become a rock-solid platform that can meet the scaling needs of virtually any organization.

Thus, using DevOps technology our website can handle any type of load on the web server and there is very little chance that our website goes down for a sec.

Chapter 5

System Flow Chart

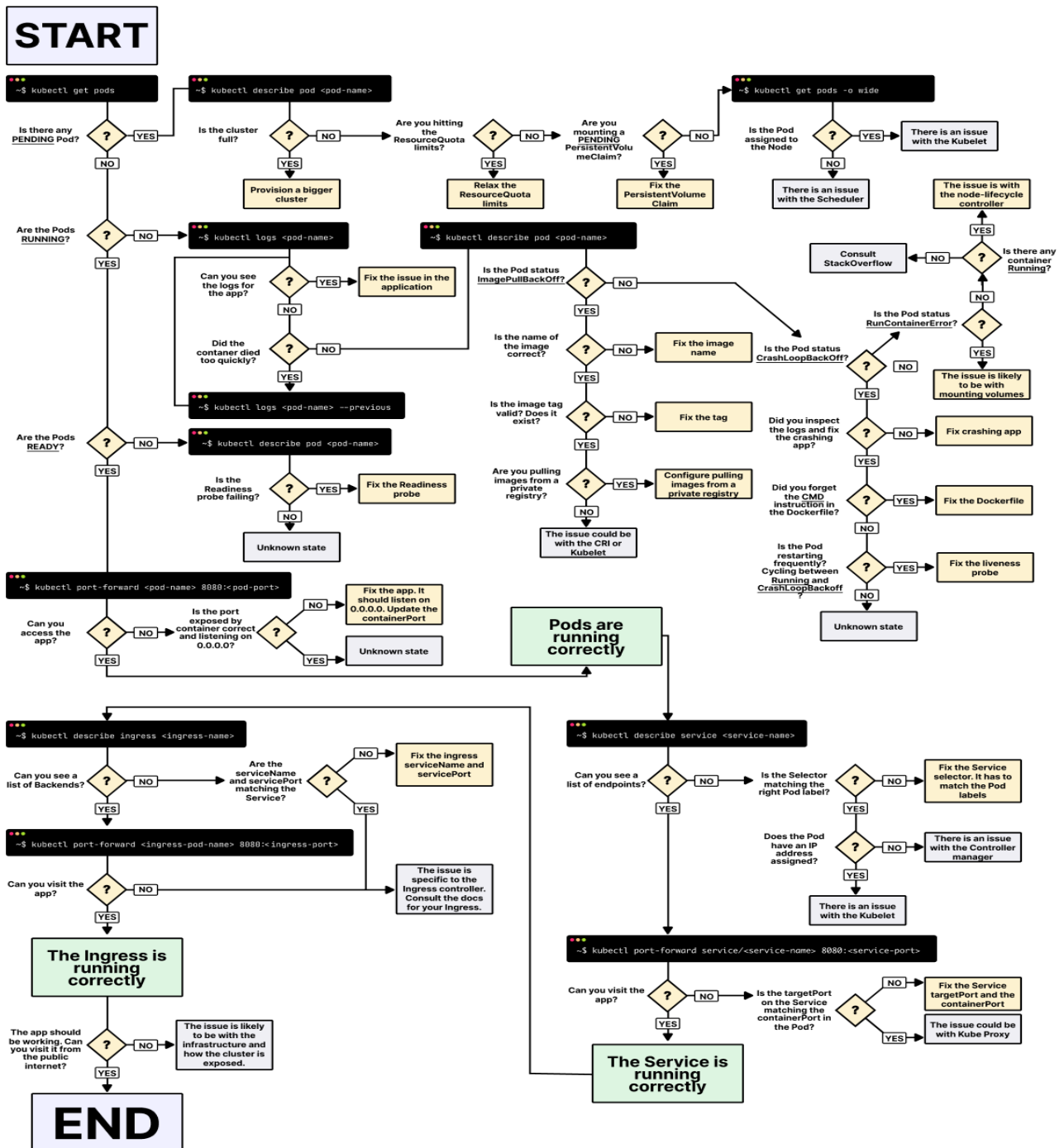


Figure (5.1) System Flow Chart

Chapter 6

Screenshots Of Project

Total, we have created 25 Html files in which 4 Html files are important so their code is pasted here.

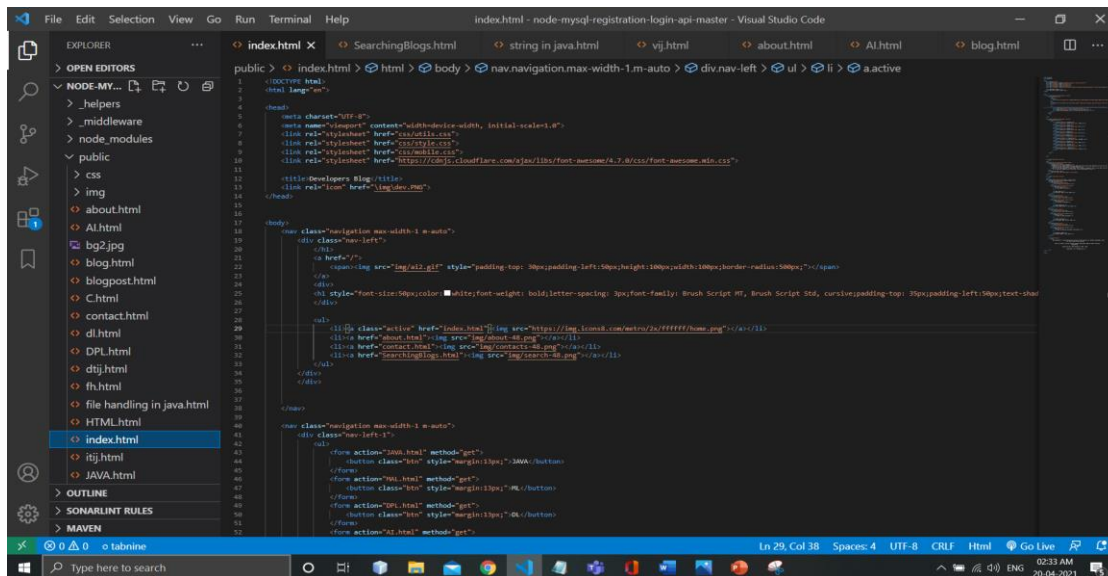


Figure (6.1) Index.html

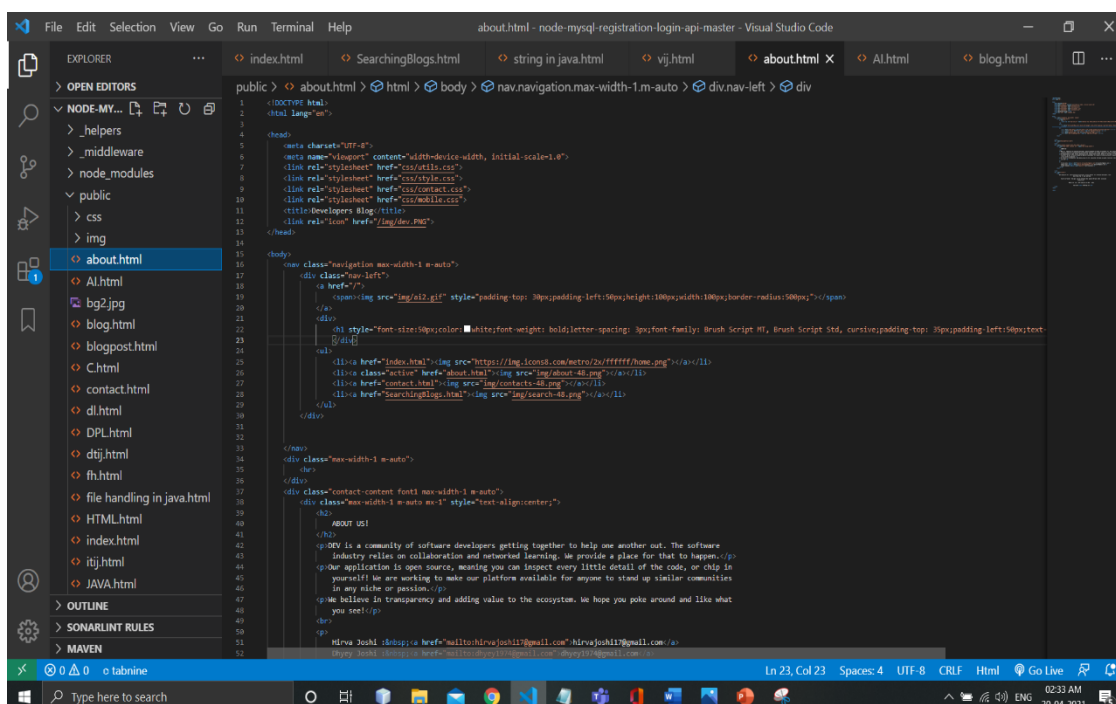


Figure (6.2) About.html

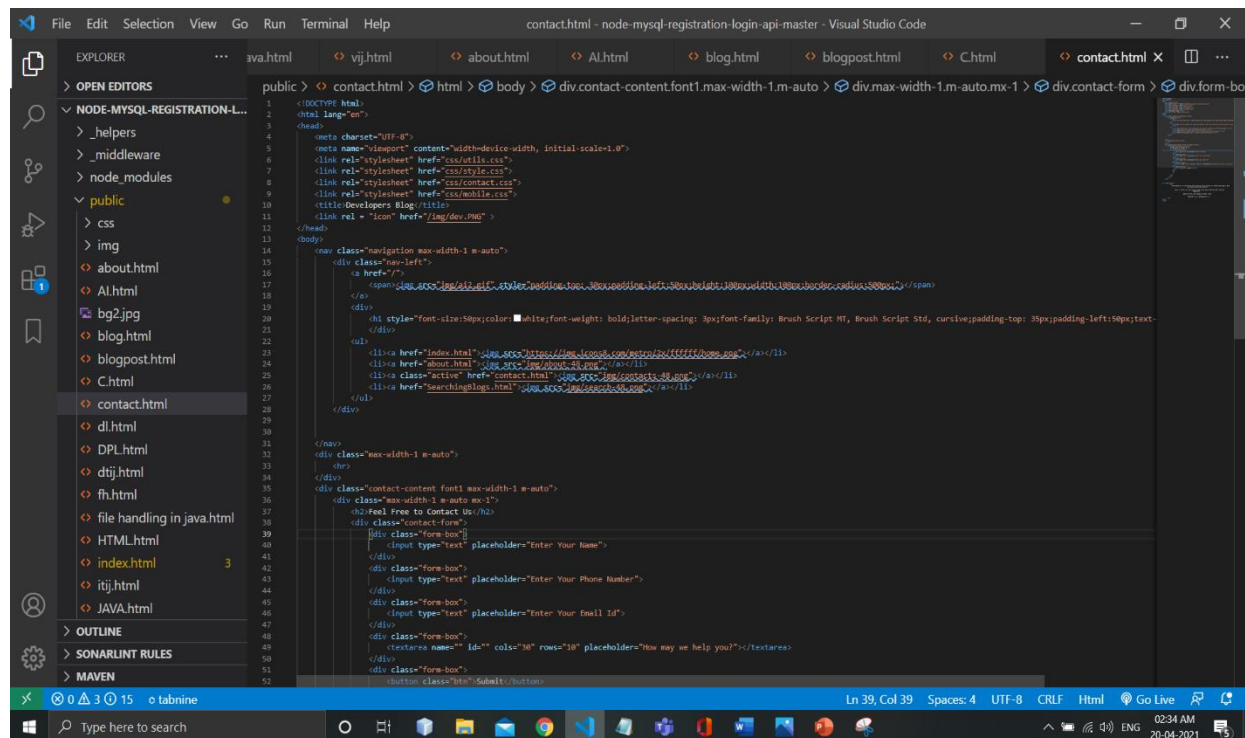


Figure (6.3) contact.html

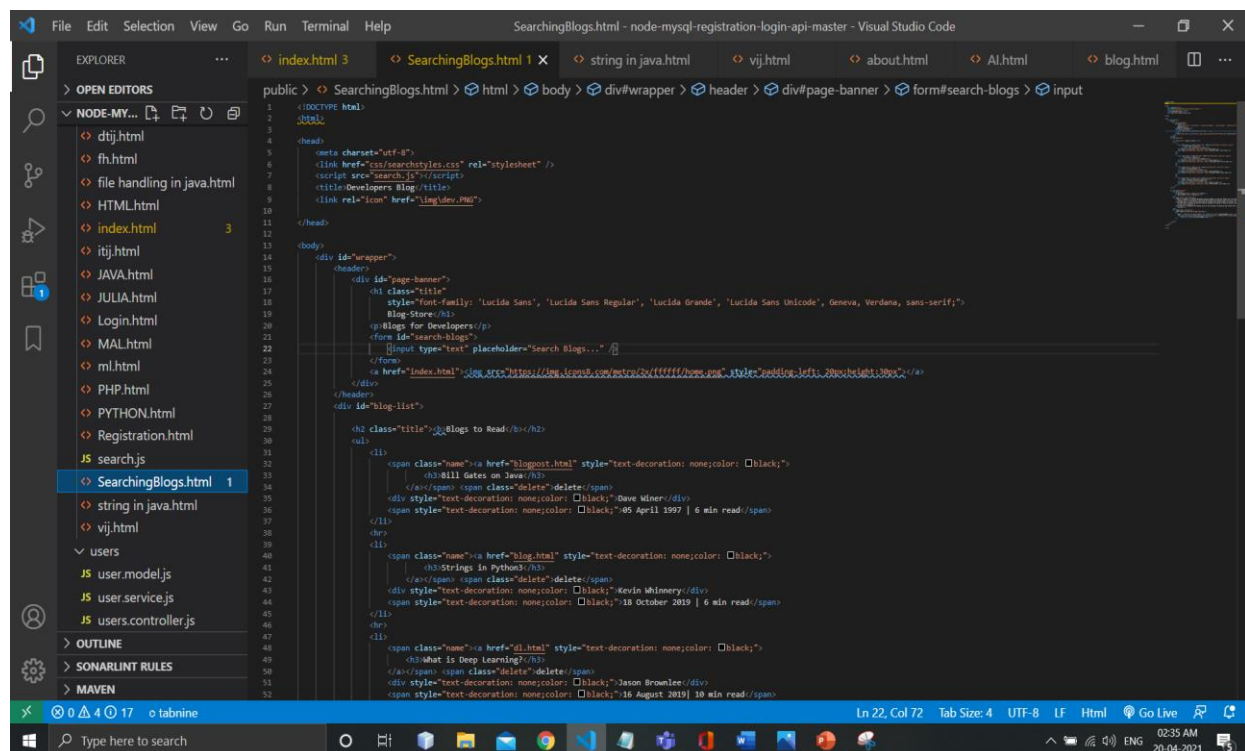


Figure (6.4) SearchingBlog.html

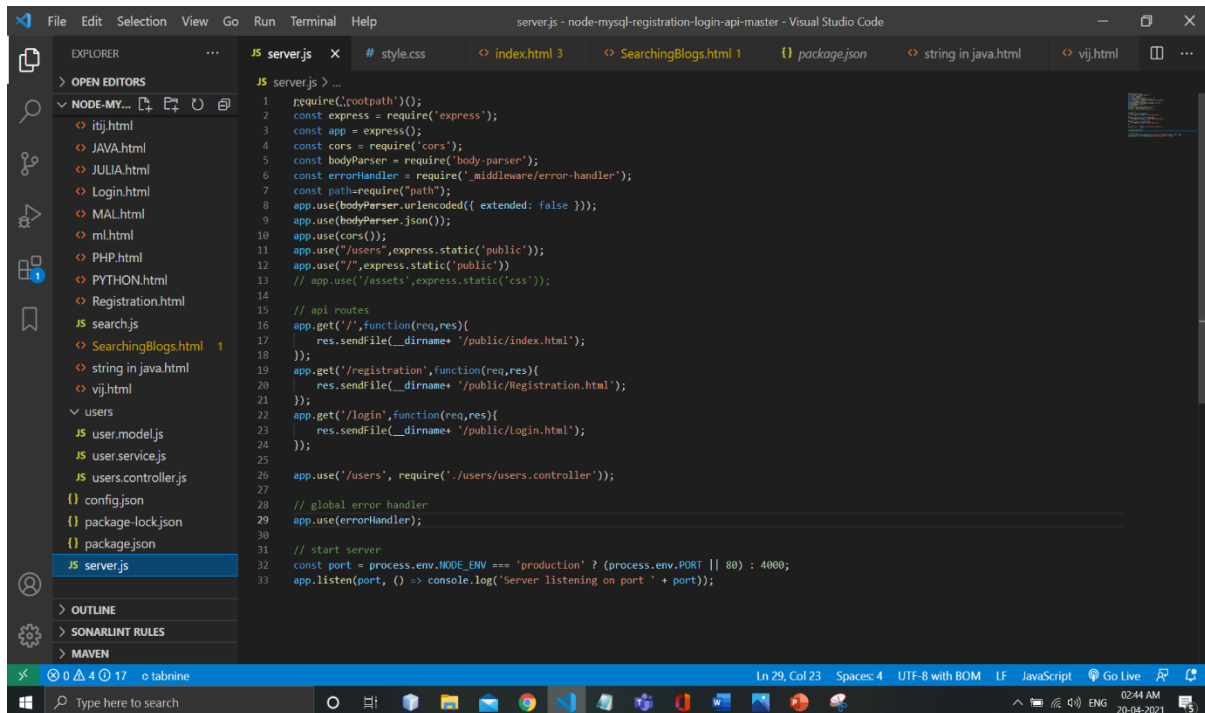


Figure (6.5) Server.js

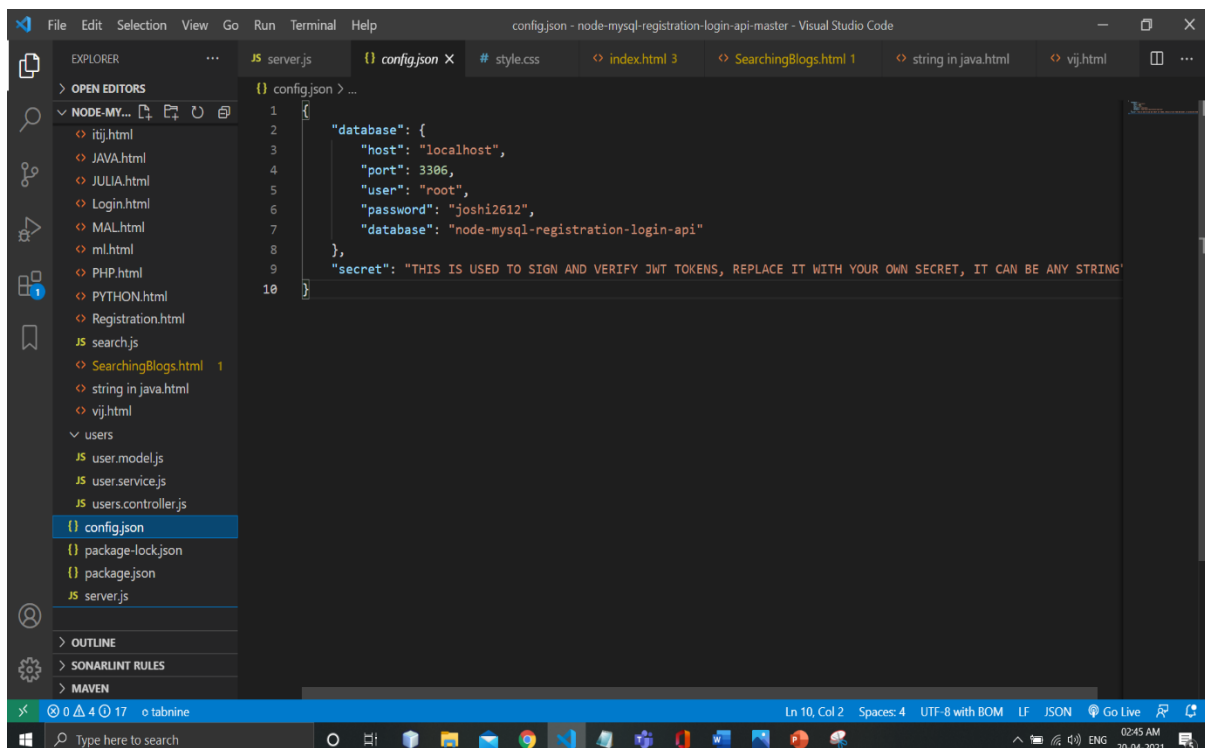
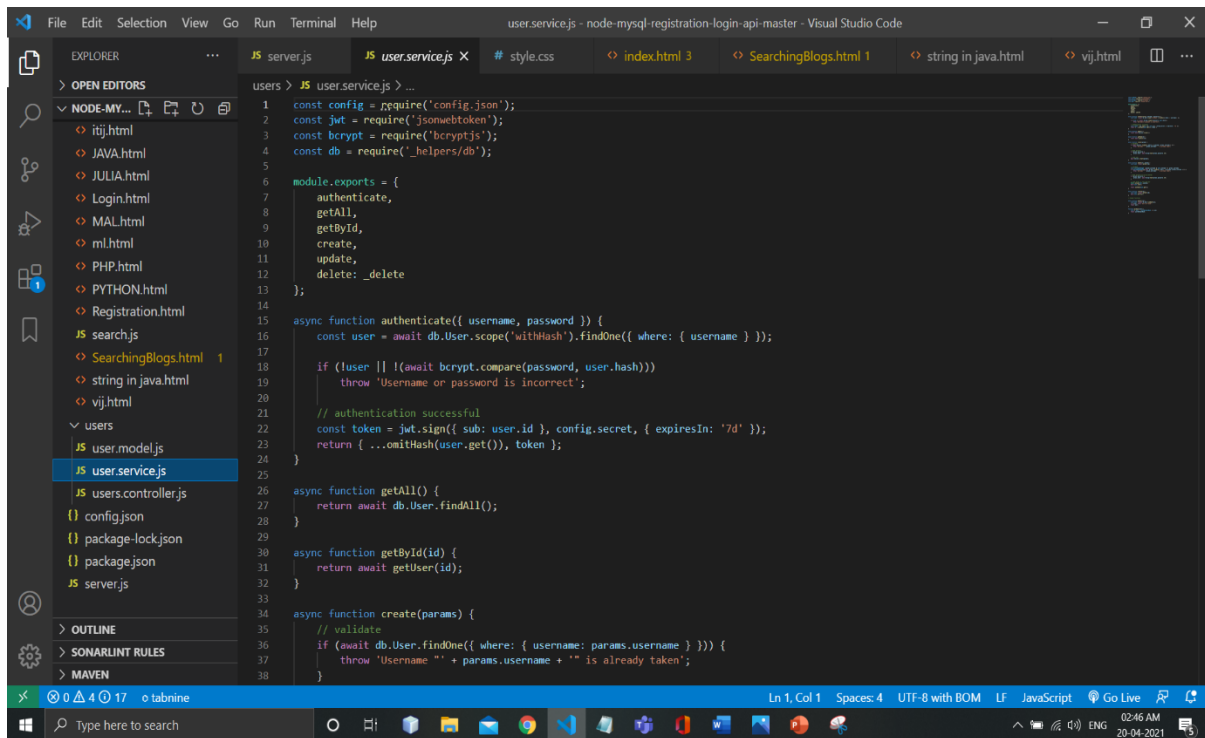
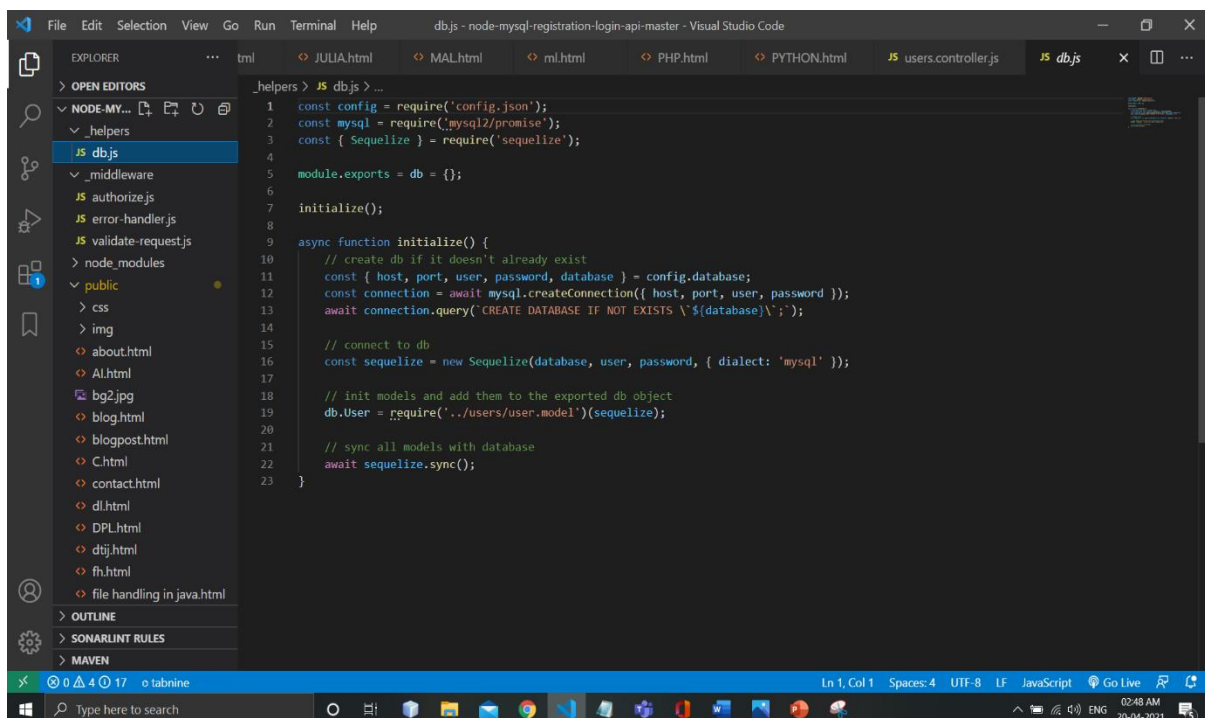


Figure (6.6) config.json (Database Connection)



```
1 const config = require('config.json');
2 const jwt = require('jsonwebtoken');
3 const bcrypt = require('bcryptjs');
4 const db = require('_helpers/db');
5
6 module.exports = {
7   authenticate,
8   getAll,
9   getById,
10  create,
11  update,
12  delete: _delete
13 };
14
15 async function authenticate({ username, password }) {
16   const user = await db.User.scope('withHash').findOne({ where: { username } });
17
18   if (!user || !(await bcrypt.compare(password, user.hash)))
19     throw 'Username or password is incorrect';
20
21   // authentication successful
22   const token = jwt.sign({ sub: user.id }, config.secret, { expiresIn: '7d' });
23   return { ...omitHash(user.get()), token };
24 }
25
26 async function getAll() {
27   return await db.User.findAll();
28 }
29
30 async function getById(id) {
31   return await getUser(id);
32 }
33
34 async function create(params) {
35   // validate
36   if (await db.User.findOne({ where: { username: params.username } })) {
37     throw 'Username ' + params.username + ' is already taken';
38   }
39 }
```

Figure (6.7) user.service.js (User authentication)



```
1 const config = require('config.json');
2 const mysql = require('mysql2/promise');
3 const { Sequelize } = require('sequelize');
4
5 module.exports = db = {};
6
7 initialize();
8
9 async function initialize() {
10   // create db if it doesn't already exist
11   const { host, port, user, password, database } = config.database;
12   const connection = await mysql.createConnection({ host, port, user, password });
13   await connection.query('CREATE DATABASE IF NOT EXISTS `$(database)`');
14
15   // connect to db
16   const sequelize = new Sequelize(database, user, password, { dialect: 'mysql' });
17
18   // init models and add them to the exported db object
19   db.User = require('../users/user.model')(sequelize);
20
21   // sync all models with database
22   await sequelize.sync();
23 }
```

Figure (6.8) db.js (SQL Query)

Chapter 7

Limitations of the project

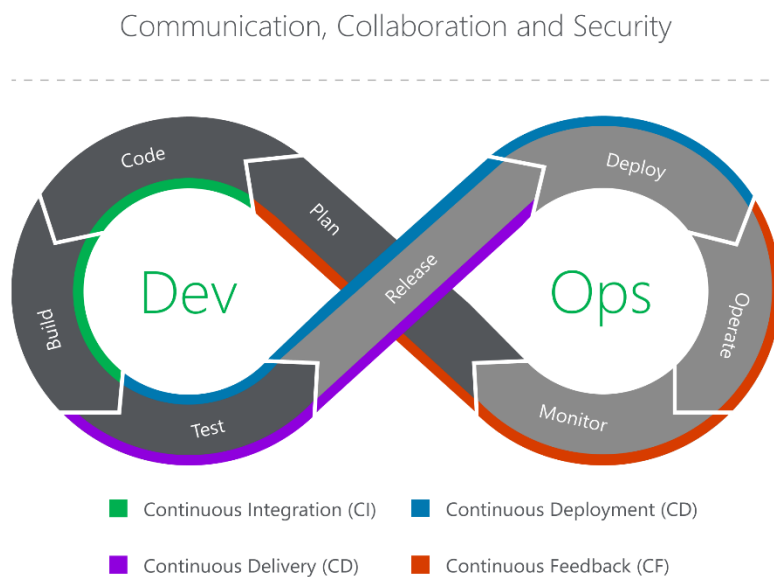
When talking about DevOps, it's useful to divide the process into phases which come together to make a DevOps pipeline. This way, we can break down the problem of describing the tools and processes used throughout the various phases.

Different people will apply their scalpel to cut the pipeline in different places and come up with a different list of phases, but the result usually describes the same process. The important thing is to be consistent with terminology within your organization so everybody is on the same page.

It's worth noting that, while it's useful to break the DevOps pipeline into phases to make it easier to discuss, in practice it is a continuous workflow followed by the same team or blend of teams, depending on the organizational structure.

There are no hard barriers between each of the phases of the pipeline.

We have just used one of the tool of DevOps i.e. Continuous Deployment –
So this is our limitation that we are not ready to deploy our application live on fly.



Chapter 8

Project Outcomes

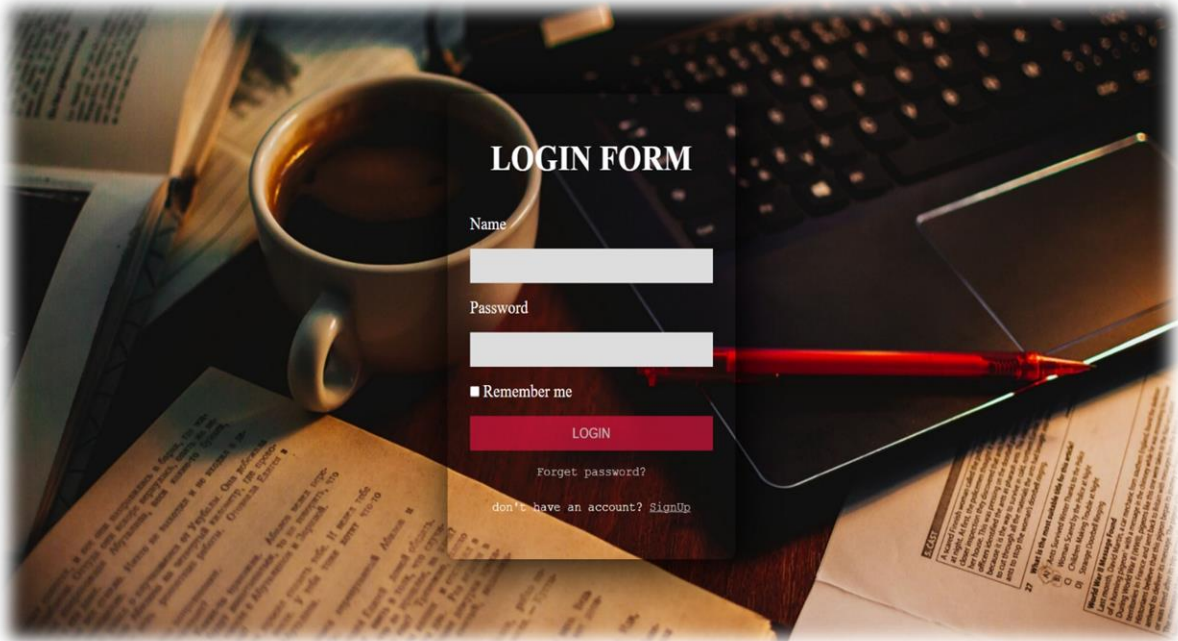


Figure (8.1) Login Page

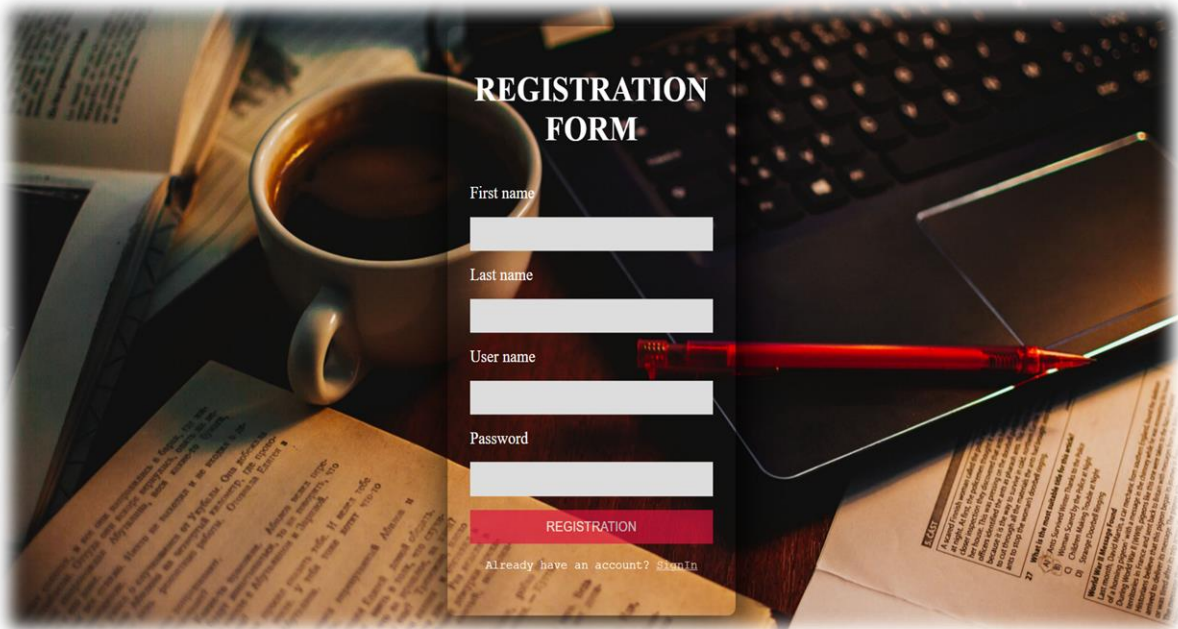


Figure (8.2) Registration Page

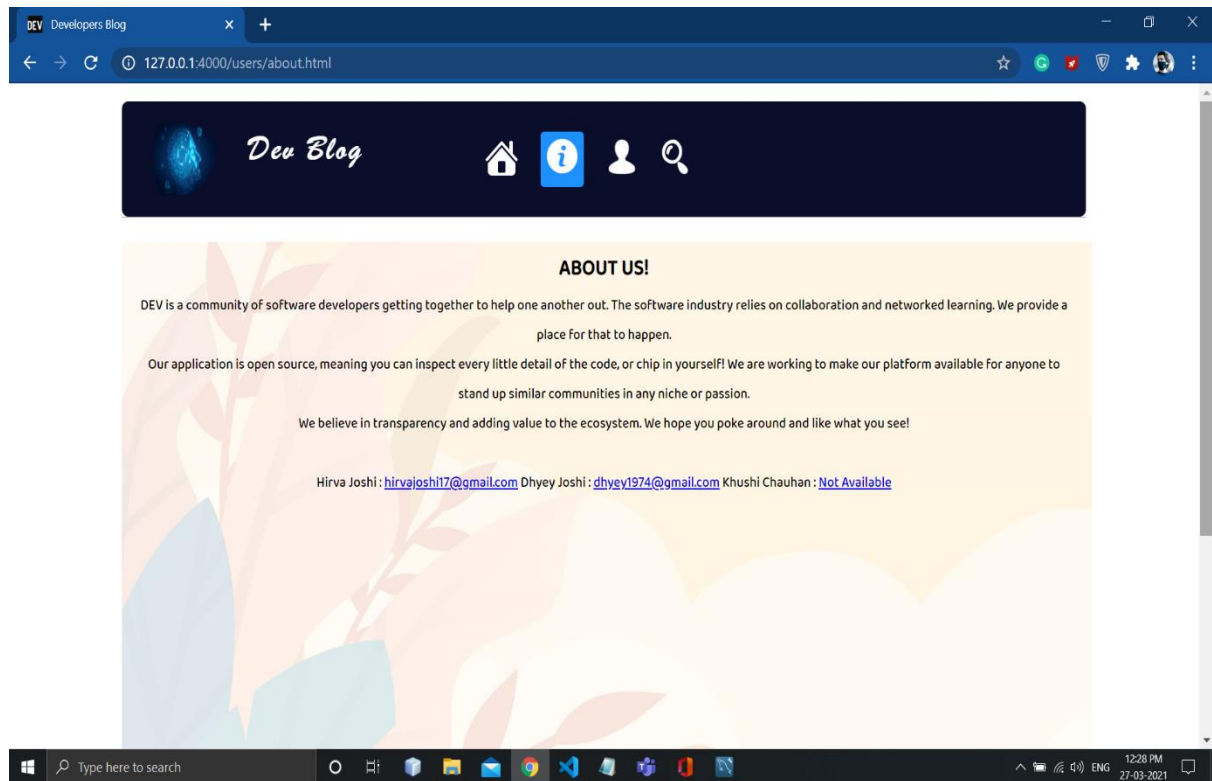


Figure (8.3) About Us Page

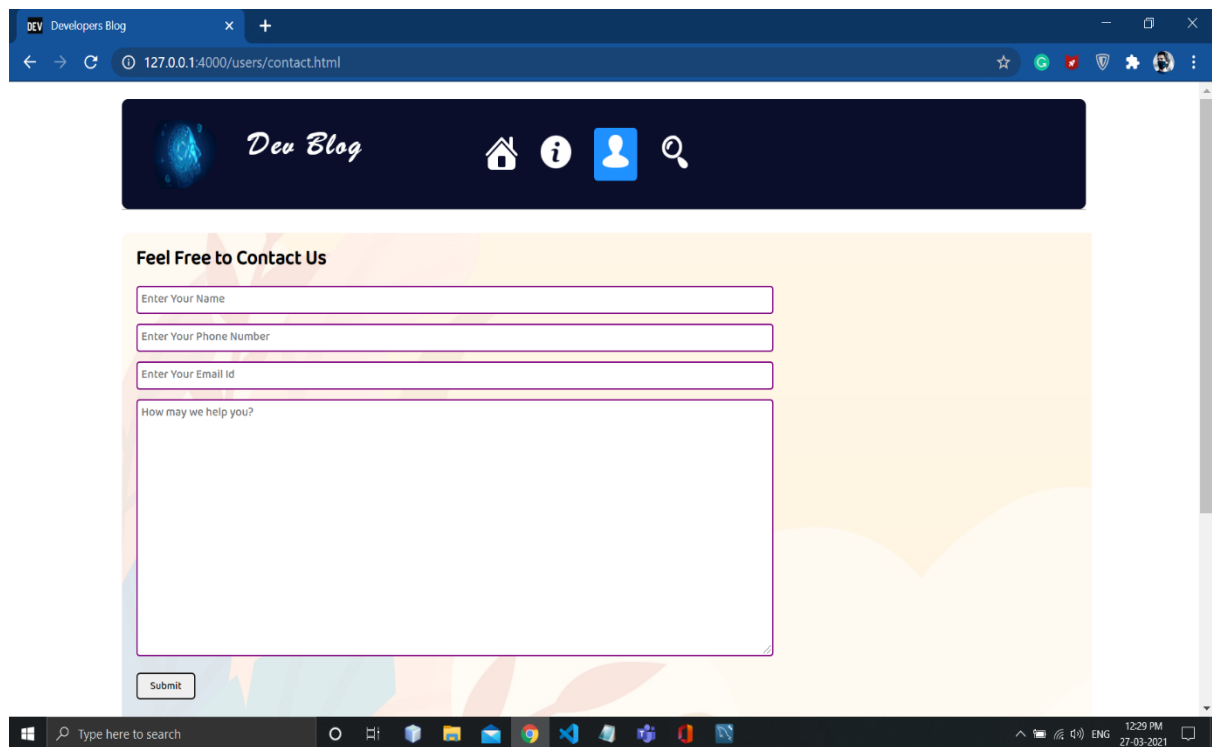


Figure (8.4) Contact Page

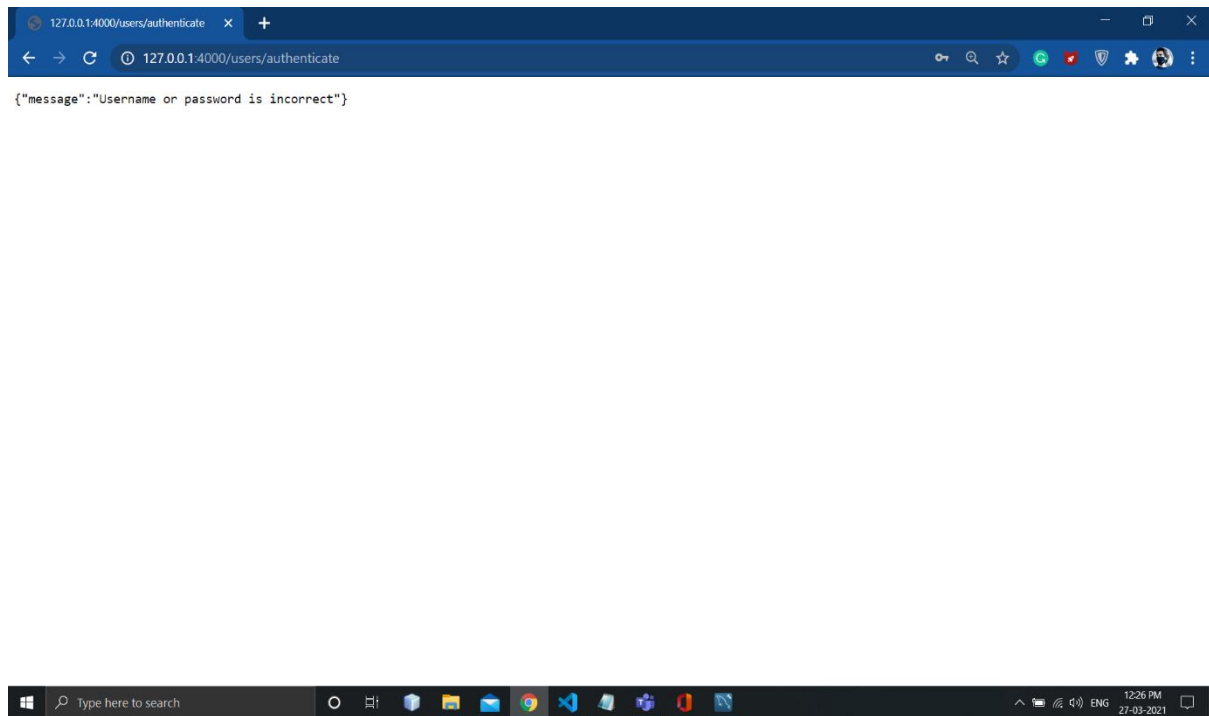


Figure (8.5) Authenticate - Username.Password

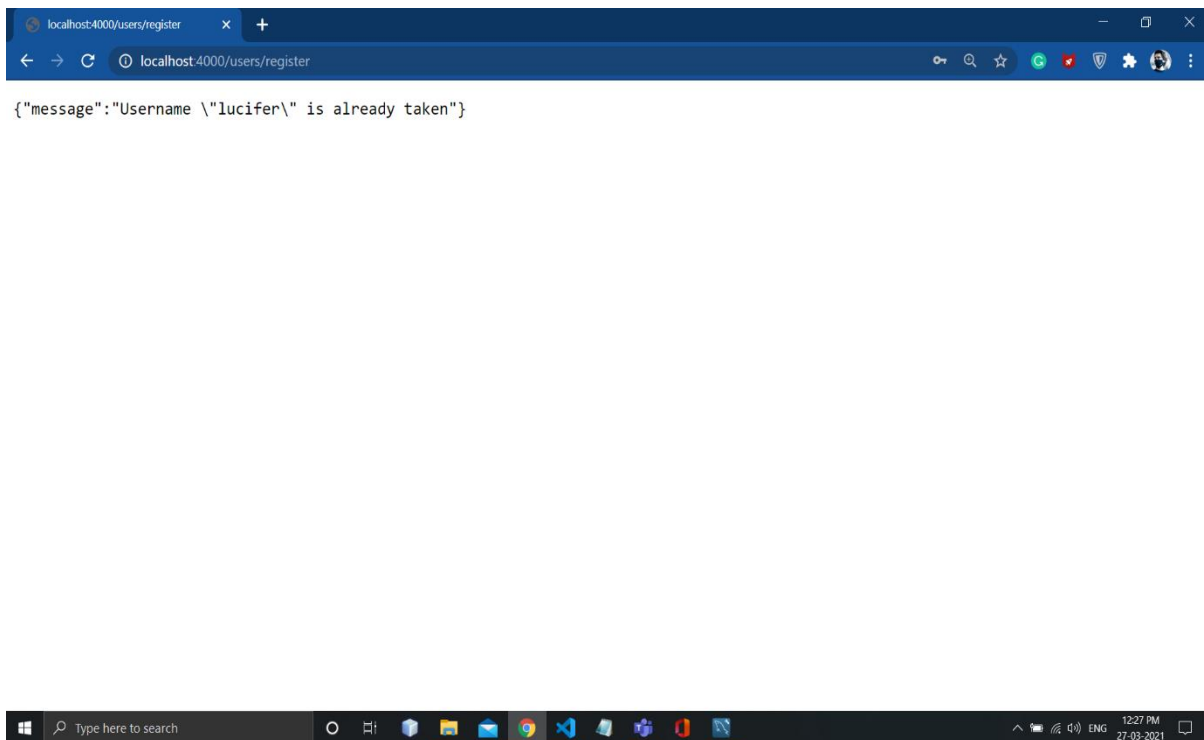


Figure (8.6) Authenticate - Username

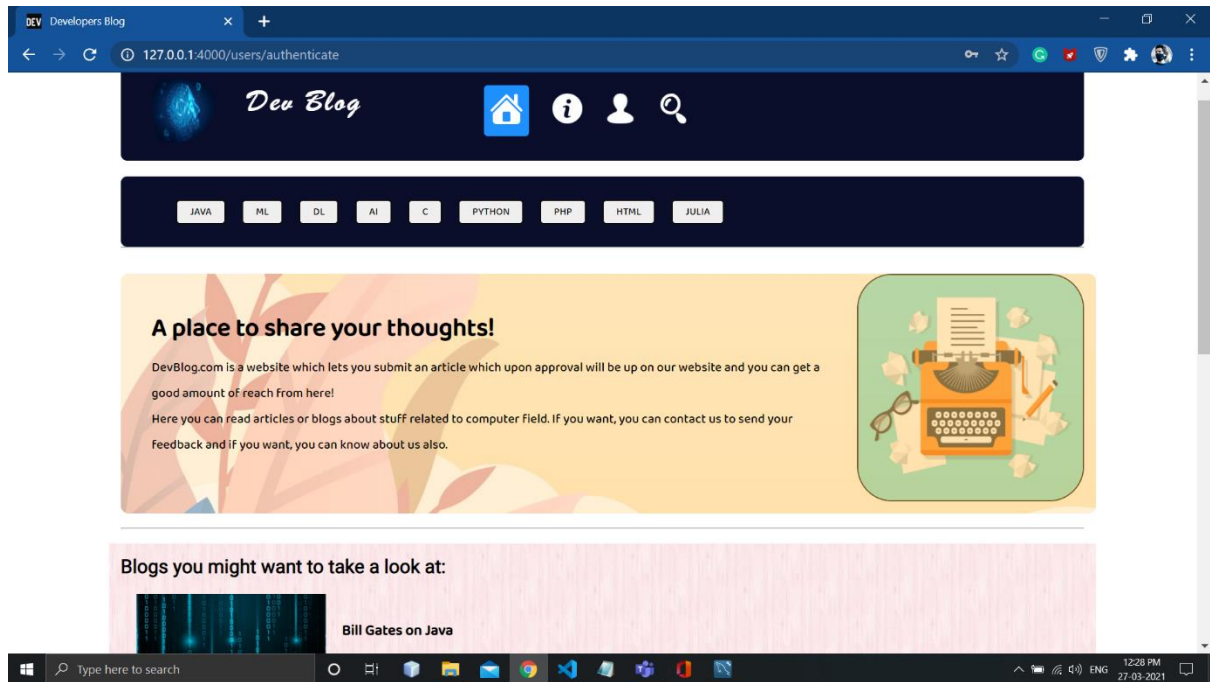


Figure (8.7) Index Page

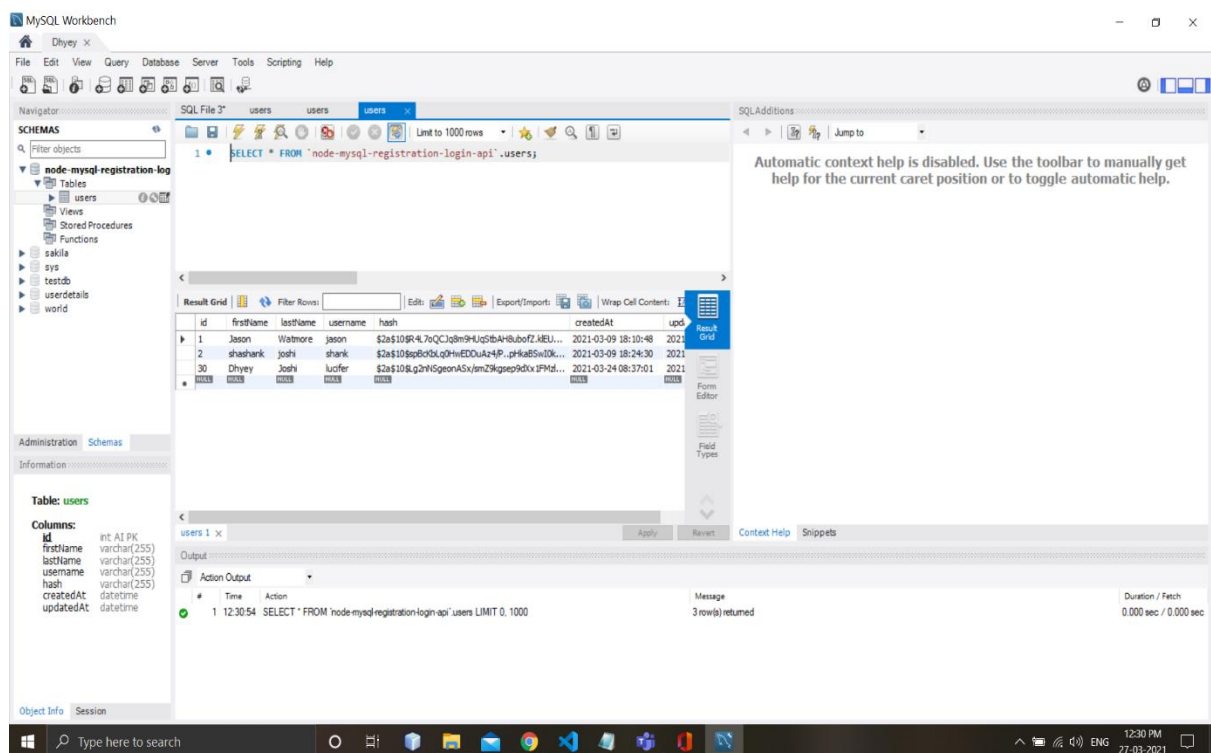


Figure (8.8) MySQL Database – Data Dictionary

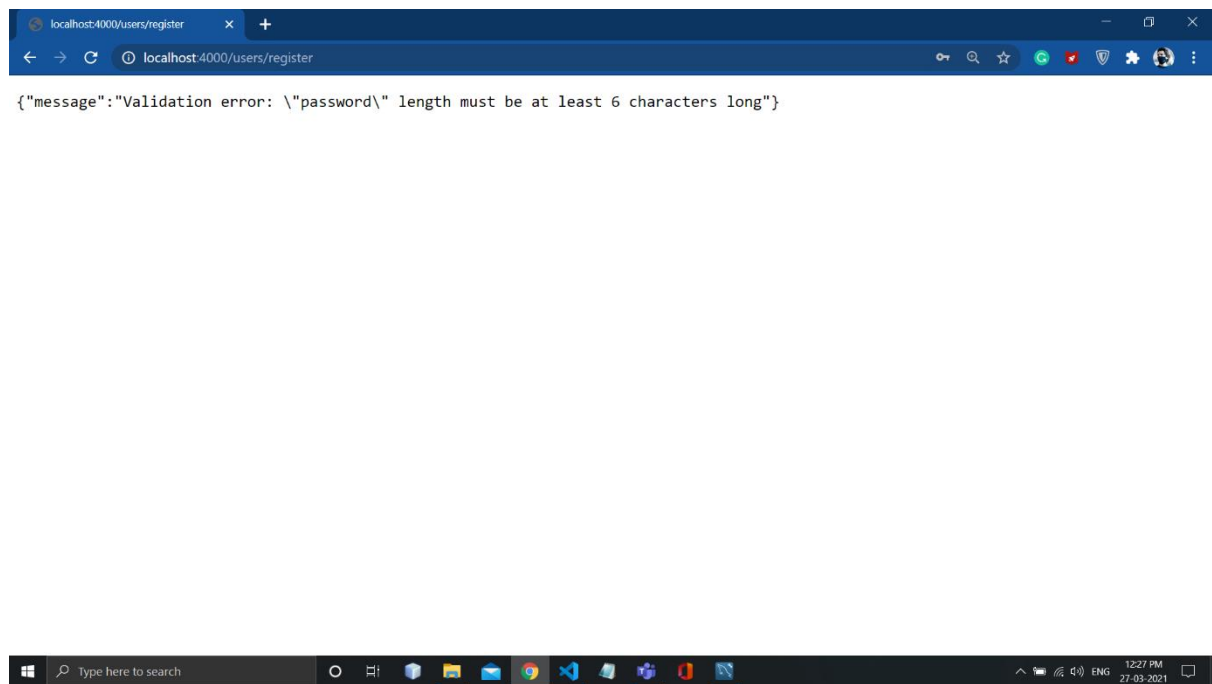


Figure (8.9) Password Validation

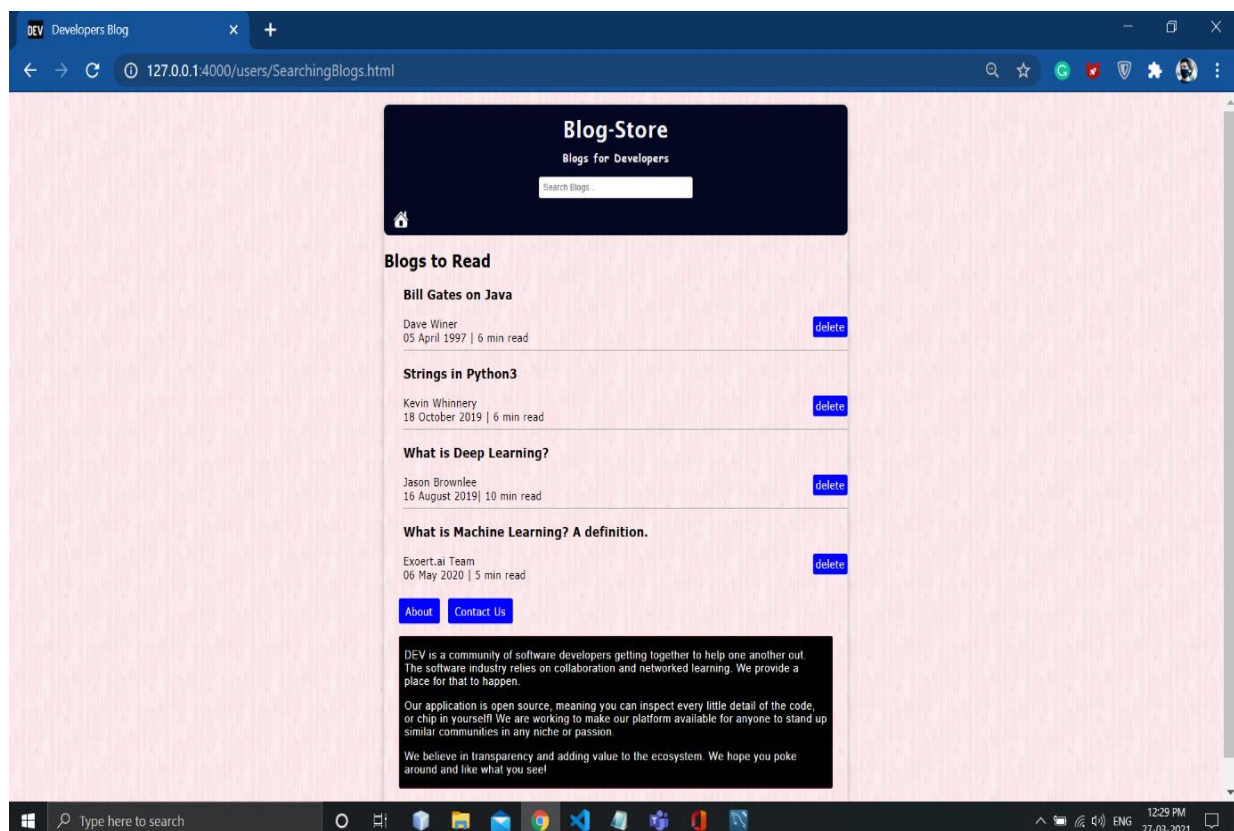


Figure (8.10) Searching Blog Page

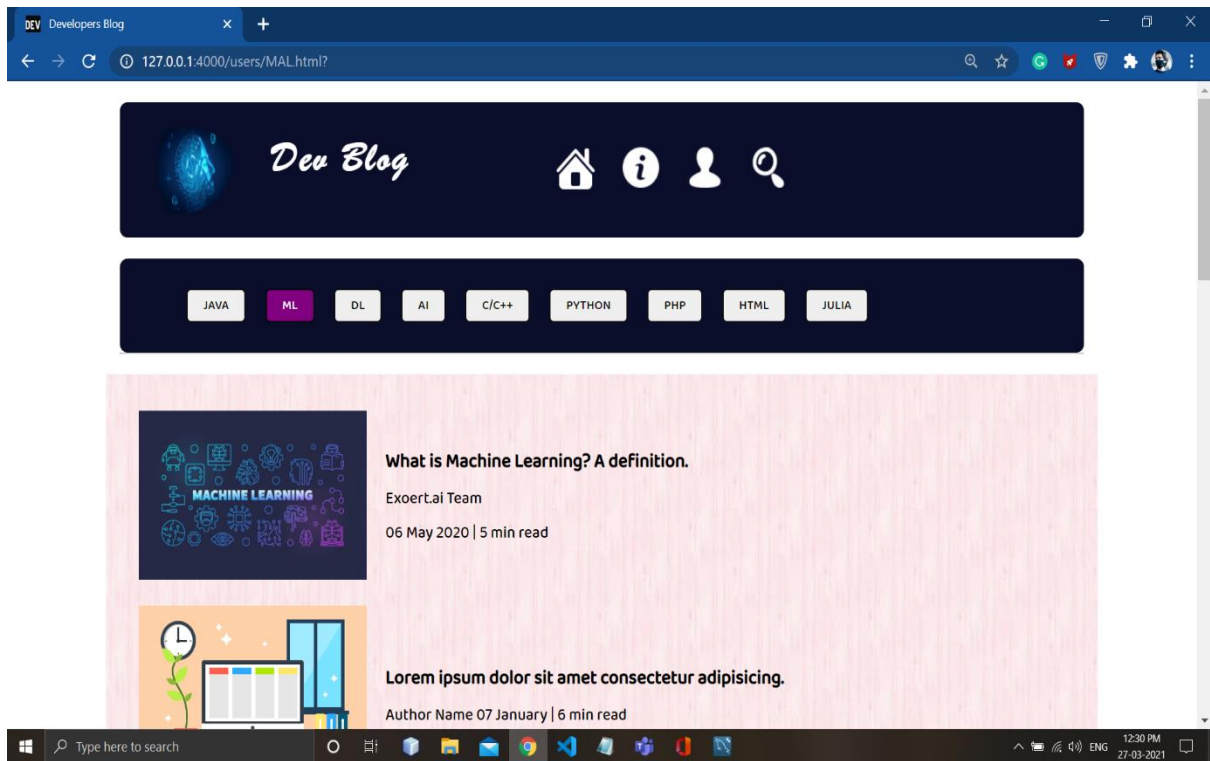


Figure (8.11) Click on any programming language displayed on screen to view related blogs.

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

COMPARISON BETWEEN BLOGGING WEBSITES - OUR WEBSITE PLAYED MUCH BETTER ROLL THEN ANY OTHER BLOGGING SITES.

COMPARISON BETWEEN BLOGGING WEBSITES				
	Blogger	WordPress.com	Medium.com	www.DevBlog.com
Software Platform	Free	Free	Free	Free
Hosting Included	Yes	Yes	No*	Yes
Domain Required	No, default domain included e.g. yourblog.blogspot.com. You can buy your own domain from \$10/yr e.g. yourblog.com	No, default domain included e.g. yourblog.wordpress.com. You can pay an additional \$13/yr to use your own domain** e.g. yourblog.com	Yes, you need to buy your own domain e.g. yourblog.com	Yes, default domain included e.g. Yourblog.ngrok.com * No Charges for your Domain Name *
Setup	Only requires an account. You can have multiple blogs under the same account.	Only requires an account. You can have multiple blogs under the same account.	You need to install medium software on your web host.	Only requires an account. You can have multiple blogs under the same account.
Software Updates	Automatic Blogger updates.	Automatic WordPress updates.	Automatic minor medium updates. Manual Theme and Plugin updates required.	Automatic DevBlog updates.
Customization	Limited compared to WordPress.	Depends on which plan you chose.	You have complete control over the look and feel of your blog.	You have complete control over the look and feel of your blog. Security : Secured Blogging *Less rick of being hacked as virtualization / containerization technology is used as backend.*

9.2 Future Scope

- ❖ We will try to make our website more better by adding new features like Create your own blog , View your post , Like , Comment , Share , Profile view .
- ❖ Also add machine learning stuffs like Recommendation System in backend.
- ❖ We will also try to make our website more dynamic by fulfilling necessities like Continuous Integration , Continuous Delivery , Continuous Feedback.

Chapter 10

References

- [1] Docker Tutorial For Beginners | What Is Docker? | Docker Explained | DevOps Tools | Simplilearn :
You-tube channels.....
[1.1] <https://www.youtube.com/watch?v=JKRAIolBNWk>

- [2] How to build Node.js Application with Docker | Node.js with Docker:
You-tube channels.....
[2.1] <https://www.youtube.com/watch?v=B7HNhe-dcFk>

- [3] How to Safely Push Docker Image to Docker Hub Remote Registry:
You-tube channels.....
[3.1] <https://www.youtube.com/watch?v=fdQ7MmQNTa0&t=186s>

- [4] What Is Kubernetes | Kubernetes Introduction | Kubernetes Tutorial For Beginners | Edureka:
You-tube channels.....
[4.1] https://www.youtube.com/watch?v=F-p_7XaEC84
- [5] Deploying WordPress on Kubernetes, Step-by-Step
You-tube channels.....
[5.1] <https://www.youtube.com/watch?v=-Hn7vFAR-9s>
- [6] Kubernetes - K8s
Training Website.....
[6.1] <https://learnk8s.io/first-steps>