International Institute of Information Technology, Bangalore

Project Elective Report

Personal Library Management

Under the Guidance of
Prof. B. Thangaraju

Simran Basu (MT2020145)
Naga Sri Vaishnavi Dhulipalla (IMT2017514)

CONTENTS

1 ABSTRACT

Personal library lets people collect and donate books online. It lets readers exchange books and expand there collection online. Every user can have and manage there own library with the online application.
The architecture of our project demands two layers:

- Front end - handled by HTML, CSS, JavaScript, Django-crispy forms and Bootstrap4.
- Back end - handled by Django and sqlite3 as database.

2 SYSTEM CONFIGURATION AND TOOLS USED

This project is developed using DevOps practices. 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.

1. **Programming Language:** Python (Django)
2. **Testing:** Django testing
3. **Build:** Virtual Environment for Python(pipenv)
4. **Source Code Management:** GitHub for Git.
   https://github.com/VaishnaviDhulipalla2902/DevOps_Project
5. **Containerization:** Docker.
   https://hub.docker.com/repository/docker/vaishnavi2902/personal_library
6. **Continuous Integration:** Jenkins
7. **Continuous Deployment:** Ansible
8. **Continuous Monitoring:** ELK Stack

3 BUILD - PIPENV

We created a virtual environment for the project so that the dependencies can be exported to a "requirements.txt" file. To do that we need to:

1. **Install Pipenv**

```
~$ pip install pipenv
```

2. **Start the virtual environment**
   Set the path to the working directory of the project and run the following command.

```
~$ pipenv shell
```

3. **Create a "requirements.txt" file.**
   Now, we create a requirements.txt file and push it to github.

```
~$ pip freeze > requirements.txt
```

   Now, we can see the file created inside the working directory. Anyone who wants to run the project on their machine, they just can start the virtual environment and install dependencies using :

```
~$ pip install -r requirements.txt
```

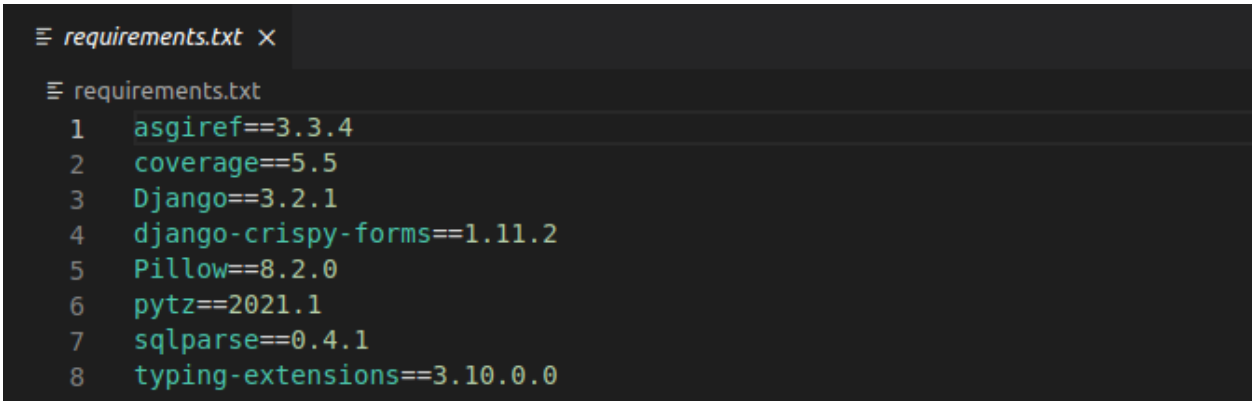


Figure 1: Requirements.txt

4 TESTING

Jenkins provides us with continuous integration which includes integrated testing, so every time we push code to github it integrates all the different modules of the project and tests their proper functioning.
For testing within python code we use TestCase and SimpleTestCase provided by django.test module. We test if the urls.py, views.py and models.py are working properly. We test

each application - library and users, separately using the following commands:

```
~$ python3 manage.py test library
```

```
~$ python3 manage.py test users
```

To test everything at once, we used a package named coverage :

```
~$ coverage run manage.py test
```

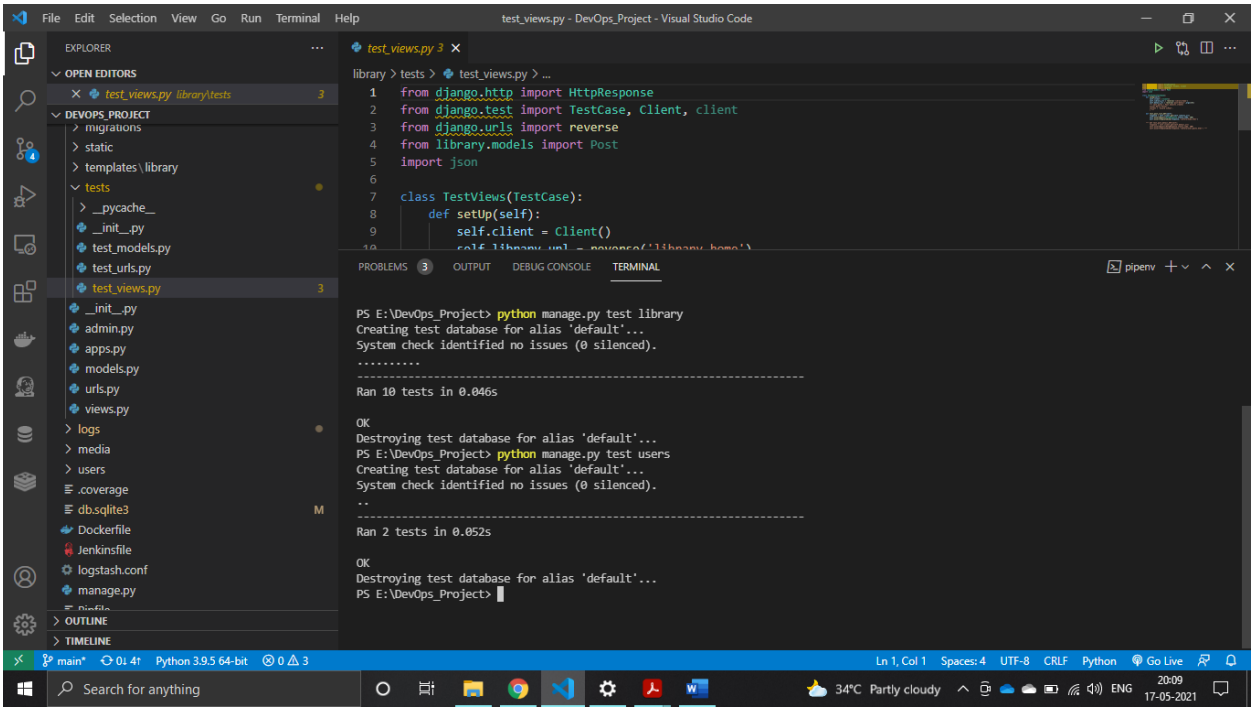


Figure 2: Testing

5 SOURCE CODE MANAGEMENT – GITHUB

Source code management (SCM) is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is also synonymous with Version control.
In this step, we create a repository on GitHub and then configure the django project on our local machine and link it to the Git repository.

```
~$ git init
```

Open the command line on the root of the Maven project and type the above command to initialize the Maven project as a Git repository.

```
~$ git remote add origin
```

This command helps to link the Git repository with local repository.

```
~$ git add .
```

This command is to Stage the Django project in Git environment. ("." command specifies to stage all the files in the current folder that are modified or created.)

```
~$ git commit –m "Commit Message"
```

This command is to commit the staged files to GitHub.

```
~$ git push –u origin main
```

This command is to push the changes to the GitHub repository.

6 CONTAINERIZATION – DOCKER

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

1. **Install docker on the local machine**

```
~$ curl –fsSL https://get.docker.com –o get-docker.sh
~$ sh get-docker.sh
```

2. **Create a repository in DockerHub**
   https://hub.docker.com/repository/docker/vaishnavi2902/personal_library

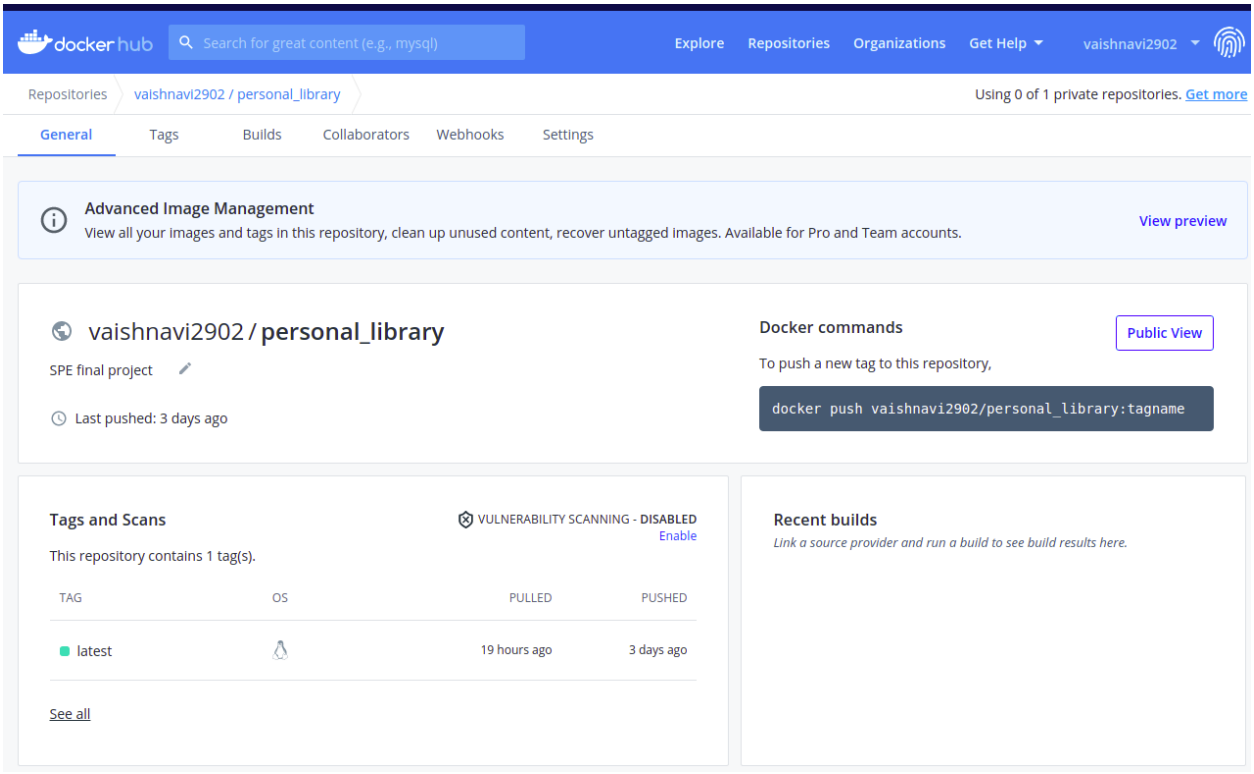


Figure 3: Docker Repository

3. **Create a Dockerfile**
   After creating a Dockerfile, push the changes to GitHub.

Dockerfile

```
FROM python:3

ENV PYTHONUNBUFFERED 1
ENV PYTHONDONTWRITEBYTECODE 1

RUN mkdir /app
WORKDIR /app
ADD . /app
COPY ./requirements.txt /app/requirements.txt
RUN pip3 install -r requirements.txt

COPY . /app

EXPOSE 8000

CMD ["python","manage.py","runserver","0.0.0.0:9010"]
```

Figure 4: Docker File

7 CONTINUOUS INTEGRATION – JENKINS

1. **Add Jenkins to the Docker group**

```
    sudo apt install openssh-server
    sudo su - jenkins

```

By executing the above commands, we get logged into jenkins. Here we configure Jenkins to use Docker image via ssh.

```
mkdir .ssh
cd .ssh
ssh-keygen -t rsa
ssh-copy-id vaishnavi@localhost
ssh vaishnavi@localhost
```

After executing the last command, we get automatically directed outside the Jenkins user.
We can now start Jenkins with the command

```
sudo systemctl start jenkins
```

By default, Jenkins starts at port number 8080 so we login on to http://localhost:8080 on to the browser.

2. **Manage Plugins**
We need to install all the required plugins like Build pipeline, Docker, GitHub, Maven Integration, Ansible etc. After they are all done installing, we need to restart Jenkins and add Docker credentials. In the Jenkins dashboard we add credentials to the Docker Hub repository and we set an unique id which is equal to docker with Registry credentials id in pipeline script.

3. **Create a New Pipeline on Jenkins**
Create new item, and select pipeline.

4. **Setup a Jenkins Pipeline** Write a Pipeline Script which includes steps like Git Clone, Build, Building a Docker image, Pushing the docker image to docker hub and Ansible Deployment. After we are done setting up every aspect of our DevOps tool chain, we may now build the jenkins job. This job for now can be manually triggered but for other SCM like GitLab it can be triggered based on events such as new push or new pull but for GitHub we might have set up a webhook. This can be done via port forwarding; we have to make jenkins port 8080 open on wide internet so it can be triggered based on event. But for now we perform this job manually.

Jenkinsfile

DevOps_Project > Jenkinsfile

```
pipeline{
    environment{
        docker_image = ""
    }
    agent any
    stages{
        stage('Step 1: Git Clone'){
            steps{
                git branch: 'main', url: 'https://github.com/VaishnaviDhulipalla2902/DevOps_Project.git'
            }
        }
        stage('Step 2: Build'){
            steps{
                sh 'pip3 install --upgrade -r requirements.txt'
            }
        }
        stage('Step 3: Testing'){
            steps{
                sh 'python3 ./manage.py test library'
                sh 'python3 ./manage.py test users'
            }
        }
        stage('Step 4: Build docker image') {
            steps {
                script {
                    docker_image = docker.build "vaishnavi2902/personal_library:latest"
                }
            }
        }
        stage('Step 5: Push docker image to hub') {
            steps {
                script {
                    docker.withRegistry('', 'docker') {
                        docker_image.push()
                    }
                }
            }
        }
        stage('Step 6: Ansible Deployment'){
            steps{
                ansiblePlaybook becomeUser: null,
                colorized: true,
                credentialsId: 'docker',
                installation: 'Ansible',
                disableHostKeyChecking: true,
                inventory: 'deployment/inventory',
                playbook: 'deployment/deploy.yml',
                sudoUser: null
            }
        }
    }
}
```

Figure 5: Jenkins Pipeline script

8 CONTINUOUS DEPLOYMENT – ANSIBLE

Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, intraservice orchestration, and provisioning. Ansible is mainly used to perform a lot of tasks that otherwise are time-consuming, complex, repetitive, and can make a lot of errors or issues.

1. **Create an Ansible playbook** We make a new directory in our working directory to create the Ansible playbook. We need to be careful about giving the correct path of docker image and the correct python version for which we can do

```
~$ which python
```

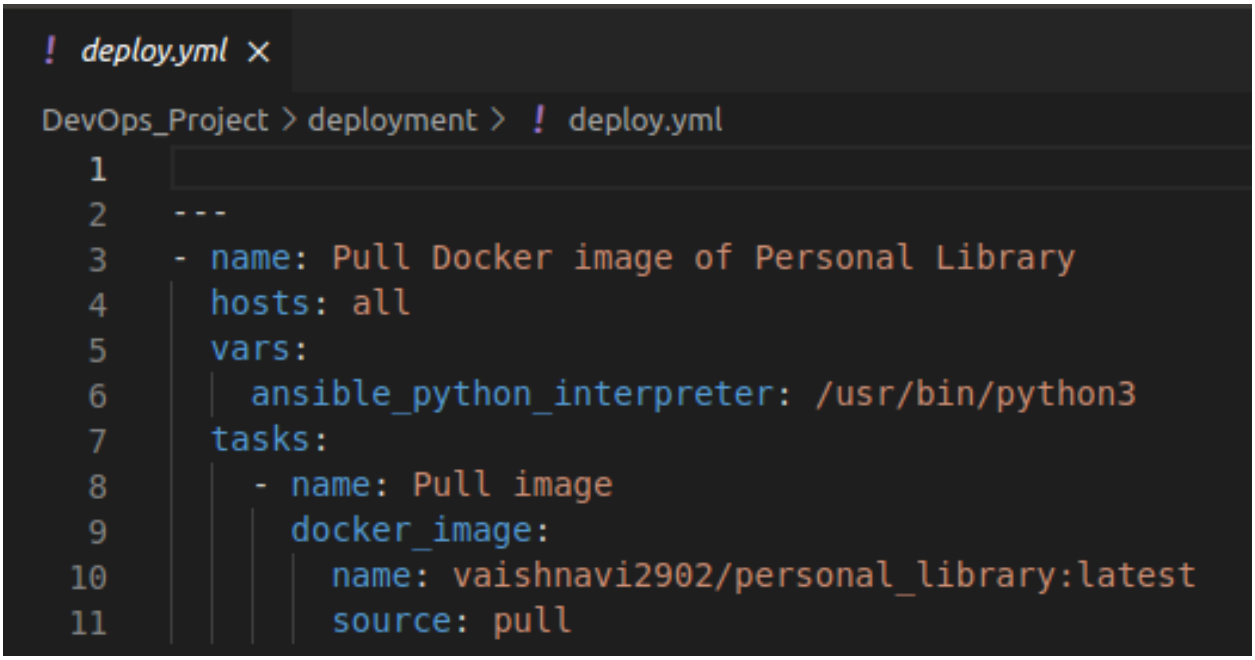


Figure 6: Ansible Playbook

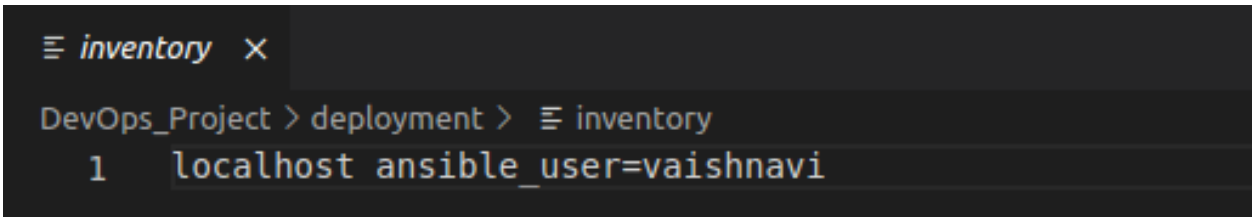


Figure 7: Inventory File

2. **Configure Ansible in Jenkins** We need to go to Jenkins -> Dashboard -> Manage Jenkins -> Global Tool Configuration and add Ansible there. Here, we also need to give the correct path to Ansible.

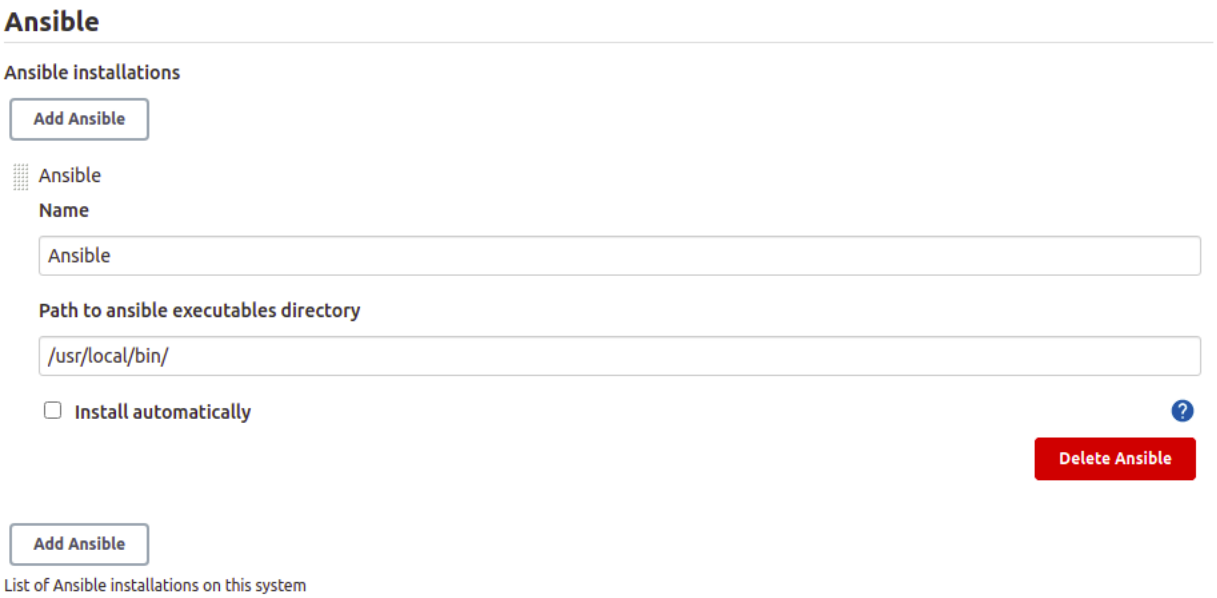


Figure 8: Ansible Configuration in Jenkins

3. **Add Ansible Stage in the Jenkins pipeline script**

```
stage('Step 5: Ansible Deployment'){
            steps{
                ansiblePlaybook becomeUser: null,
                colorized: true,
                credentialsId: 'docker',
                installation: 'Ansible'
                disableHostKeyChecking: true,
                inventory: 'deployment/inventory',
                playbook: 'deployment/deploy.yml',
                sudoUser: null
            }
        }
```

9 LOG MANAGEMENT

Logging keeps track of all the operations that were performed in the application, warnings, debugging information, errors etc. The log folder contains a file debug.log which stores all the logging information.

1. **Manage logging in settings.py** LOGGING in settings.py handles the logs and stores them in debug.log.

settings.py

DevOps_Project > django_project > settings.py > ...

```
129  STATIC_URL = '/static/'
130  DEFAULT_AUTO_FIELD='django.db.models.AutoField'
131
132  MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
133  MEDIA_URL = '/media/'
134  CRISPY_TEMPLATE_PACK = 'bootstrap4'
135
136  LOGIN_REDIRECT_URL = 'library-home'
137  LOGIN_URL = 'login'
138
139  LOGGING = {
140      'version':1,
141      'loggers':{
142          'django':{
143              'handlers':['file'],
144              'level':'DEBUG'
145          }
146      },
147      'handlers':{
148          'file':{
149              'level':'INFO',
150              'class':'logging.FileHandler',
151              'filename':'./logs/debug.log',
152              'formatter':'simpleLibrary',
153          }
154      },
155      'formatters':{
156          'simpleLibrary':{
157              'format':'{levelname} {message}',
158              'style':'{',
159          }
160      }
161  }
```

Figure 9: LOGGING

2. **Add Logger functions**
   We need to add logger functions in the main application code. Add the below line inside view.py.

```
import logging
logger = logging.getLogger(__name__)
```

```
3
```

And add some lines inside the functions to generate logs.

```
4     logger.info("Your Message");
5
```

```
@login_required
def profile(request):
    if request.method == 'POST':
        u_form = UserUpdateForm(request.POST, instance=request.user)
        p_form = ProfileUpdateForm(request.POST,
                                   request.FILES,
                                   instance=request.user.profile)
        if u_form.is_valid() and p_form.is_valid():
            u_form.save()
            p_form.save()
            messages.success(request, f'Your account has been updated
            # DEBUG
            logger.debug('debug')
            # INFO
            logger.info('Update Successful')
            # WARNING
            logger.warning('warning')
            # ERROR
            logger.error('Error Occured')
            return redirect('profile')
```

Figure 10: LOGGING

10 BUILD JENKINS PIPELINE

Now, we need to build the project on Jenkins. While we build it, we might encounter a lot of errors which we fix along the way. The full stage pipeline looks like,

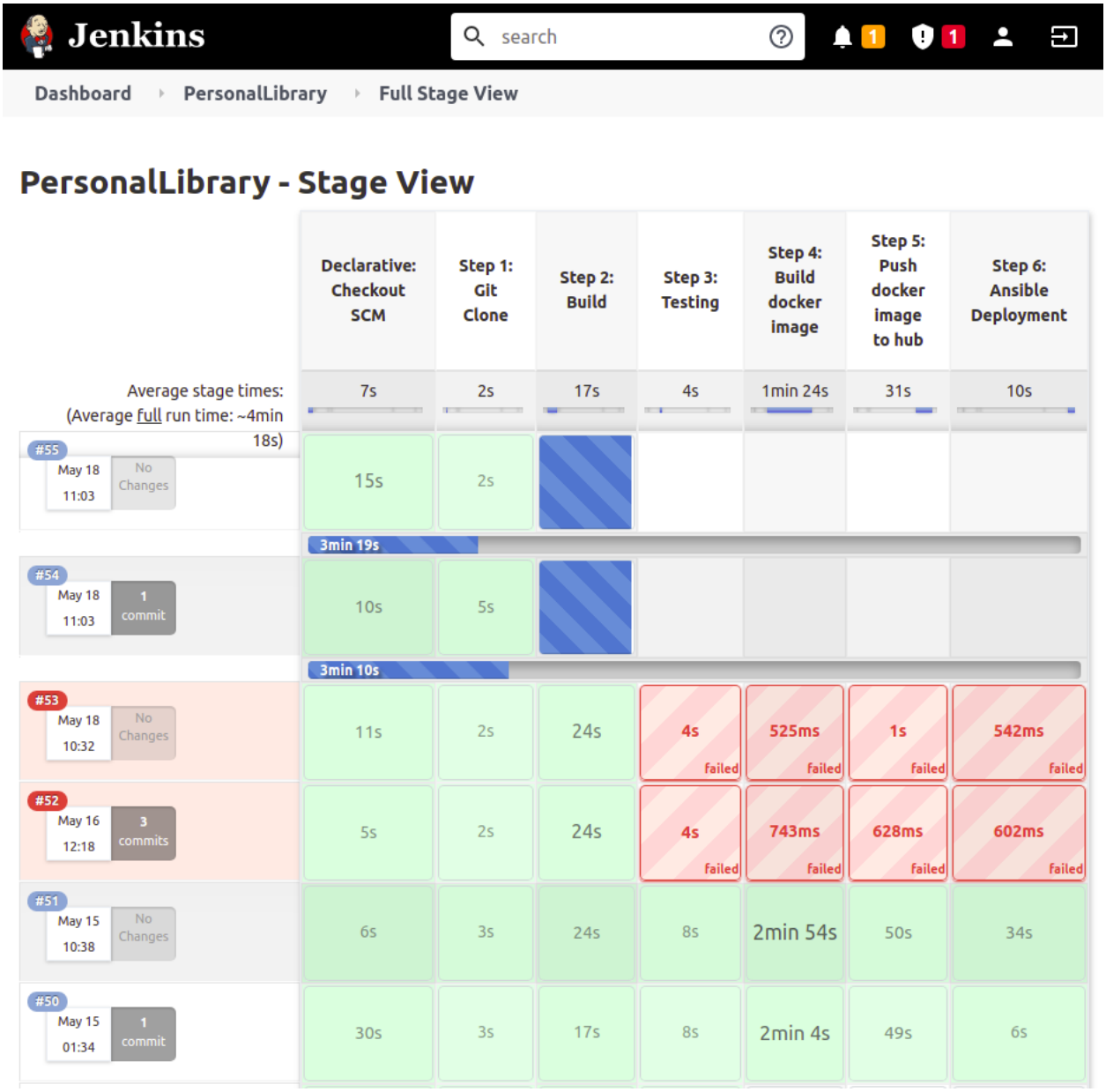


Figure 11: Full Stage Pipeline Jenkins

We may encounter some errors but we can fix them by looking up the error logs in jenkins. A successful build would look like:

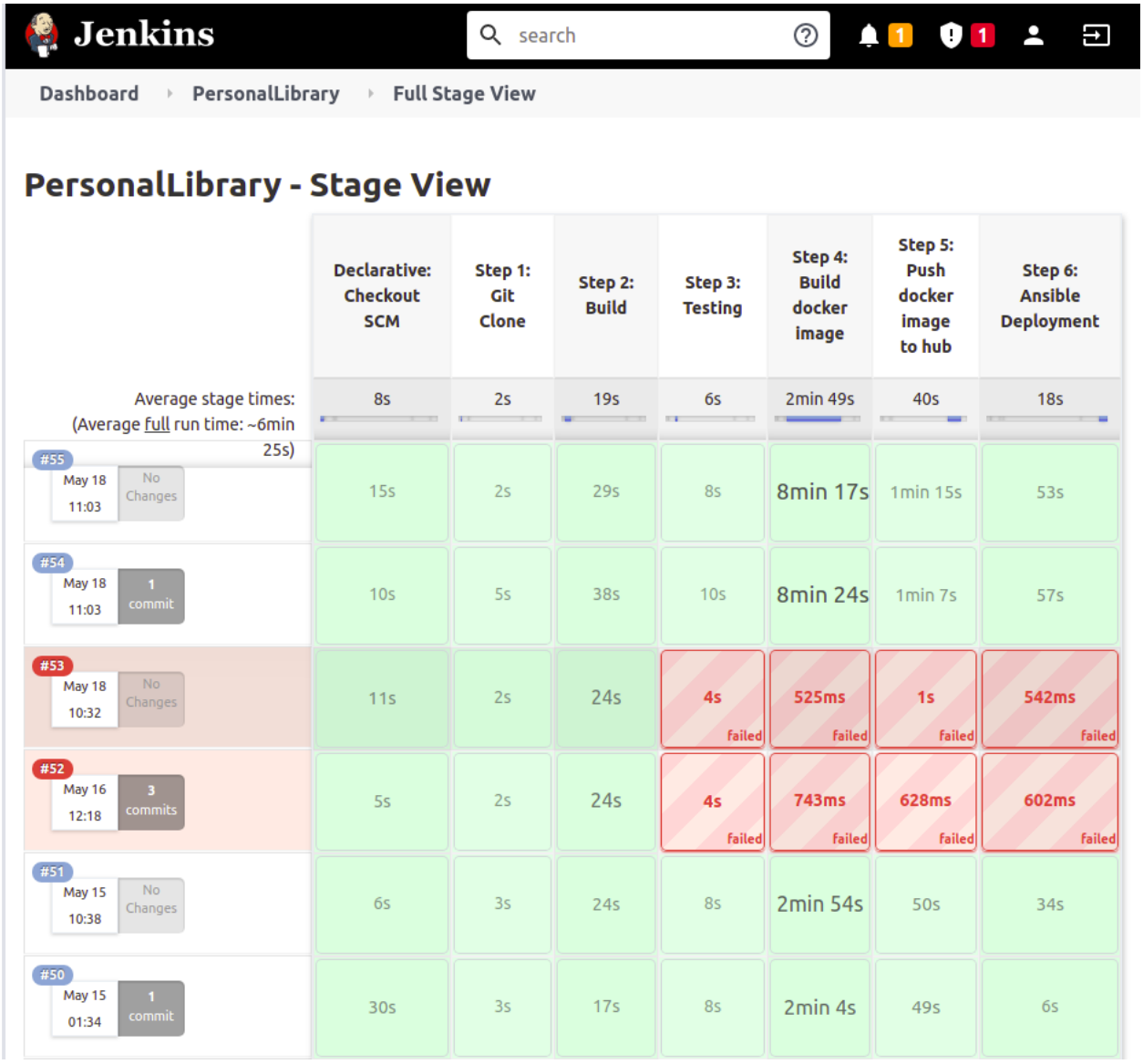


Figure 12: Full Stage Pipeline Jenkins

There are 5 stages in my Jenkins pipeline:

1. Git Clone
2. Build
3. Testing
4. Build Docker Image
5. Push Docker Image to Hub
6. Ansible Deployment

After doing all this, the docker image is pulled on to the local machine. We can run this image on the local machine to generate logs which act as input to the ELK monitoring.

```
~$ docker images
~$ docker run --name personal_library -d -p 9010:9010 vaishnavi2902/
    personal_library:latest
```

Starting and stopping the container is done as follows:

```
~$ docker ps -a
~$ docker start <container_id>
~$ docker stop <container_id>
```

Now we know that the application executes smooth and we run the application for a few times to generate the log files.

11 CONTINUOUS MONITORING – ELK

"ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. ELK stack gives us the ability to aggregate logs from all the systems and applications, analyze these logs, and create visualizations for application and infrastructure monitoring, faster troubleshooting, security analytics, and more.
First, we run ElasticSearch(from the directory it is installed in) using,

```
~$ /bin/elasticsearch
```

Elastic Search runs on http://localhost:9200.
Next run Kibana(from the directory it is installed in) using,

```
~$ /bin/kibana
```

Kibana runs on http://localhost:5601.
Next run Logstash(from the directory it is installed in) using,

```
~$ /bin/logstash -f <path_conf_file>
```

<path_conf_file> is the logstash configuration file: logstash.conf. It contains various plugins for input, output and filter. Here, we have used 2 plugins: grok and date. Grok helps to define a search and extract parts of our log line into structured fields and works by combining text patterns into something that matches the logs. The date filter is used for parsing dates from fields, and then using that date or timestamp as the logstash timestamp for the event. Logstash runs on https://localhost:9600.

logstash.conf

DevOps_Project > logstash.conf

```
input {
    stdin {
        beats {
    port => 5044
    ssl => true
  }
    }
}
filter {
    plugin {
        grok {
            match => { "message" => "%{TIMESTAMP_ISO8601:timestamp}]
                    }
        }
        date {
            locale => "en"
            match => [ "timestamp", "YYYY-MM-dd HH:mm:ss"]
            target => "@timestamp"
        }
    }
}
output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "indexforlogstash"
    }
}
```

Figure 13: logstash.conf

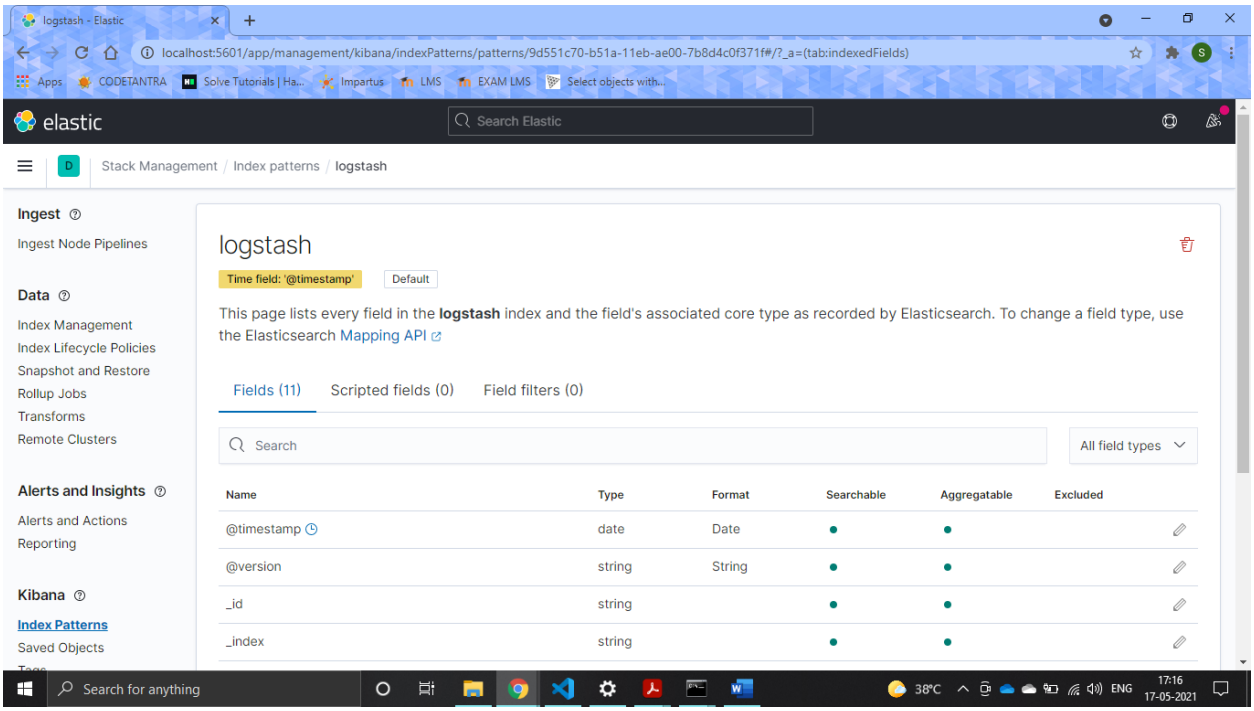
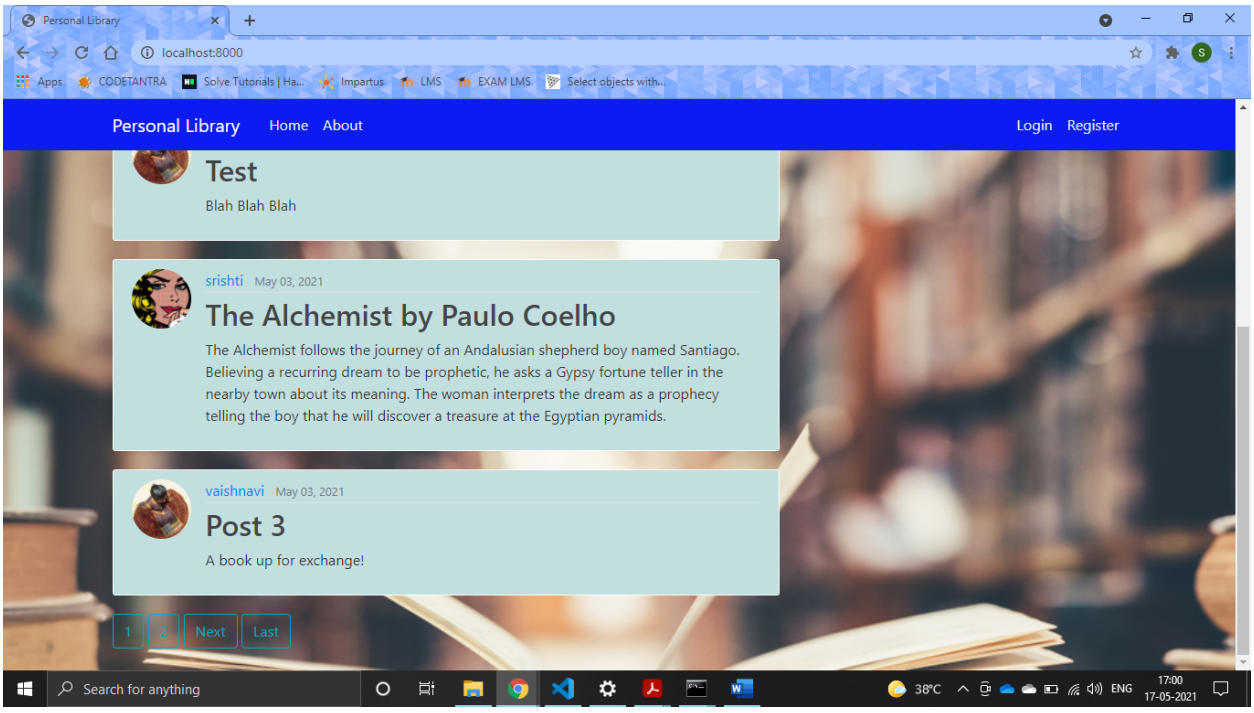
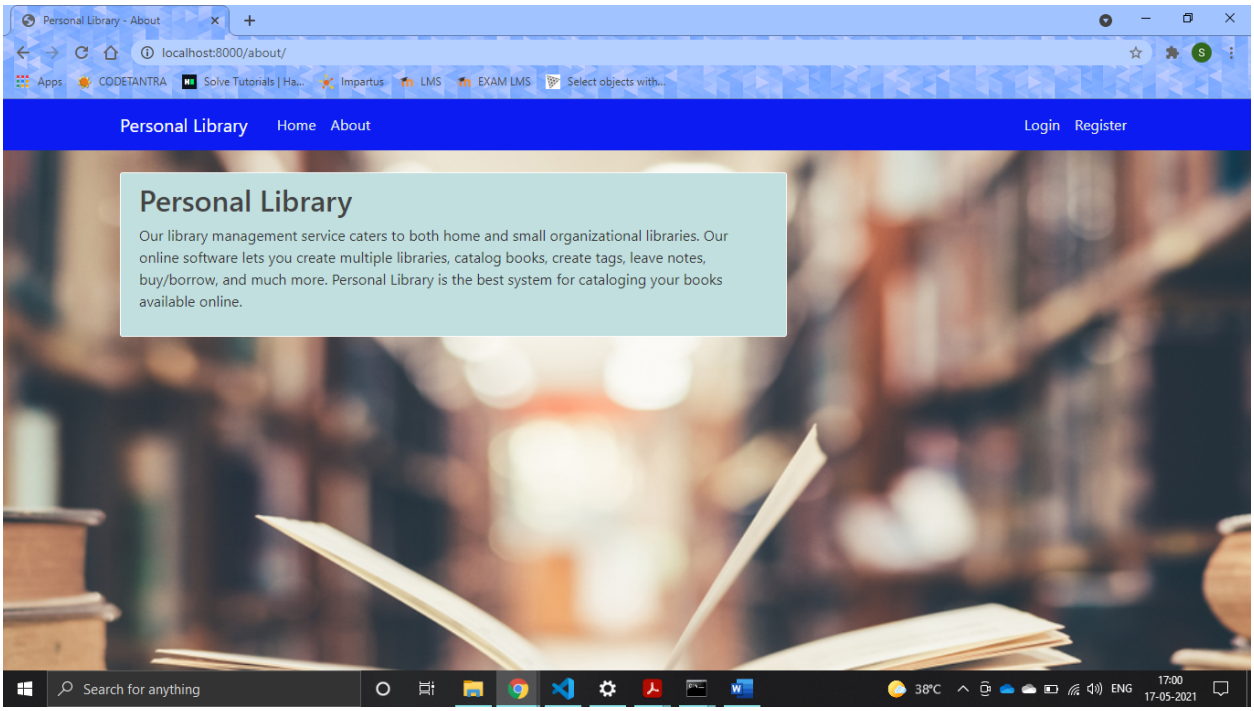
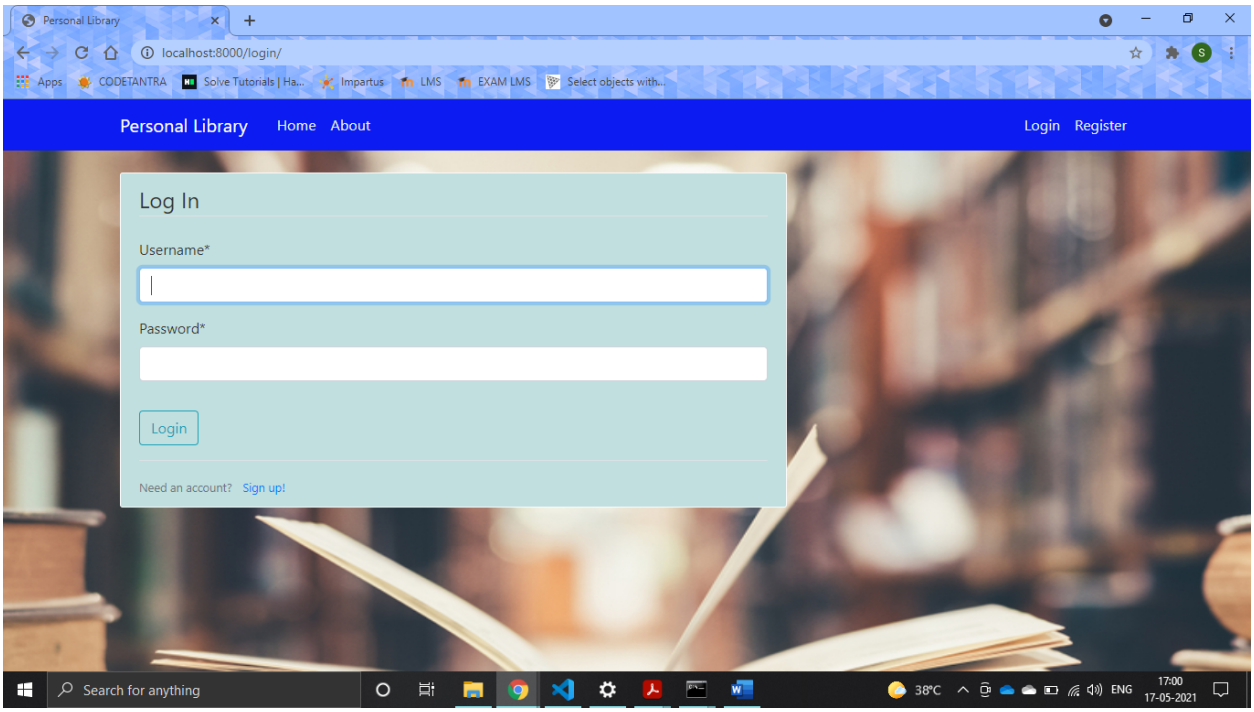


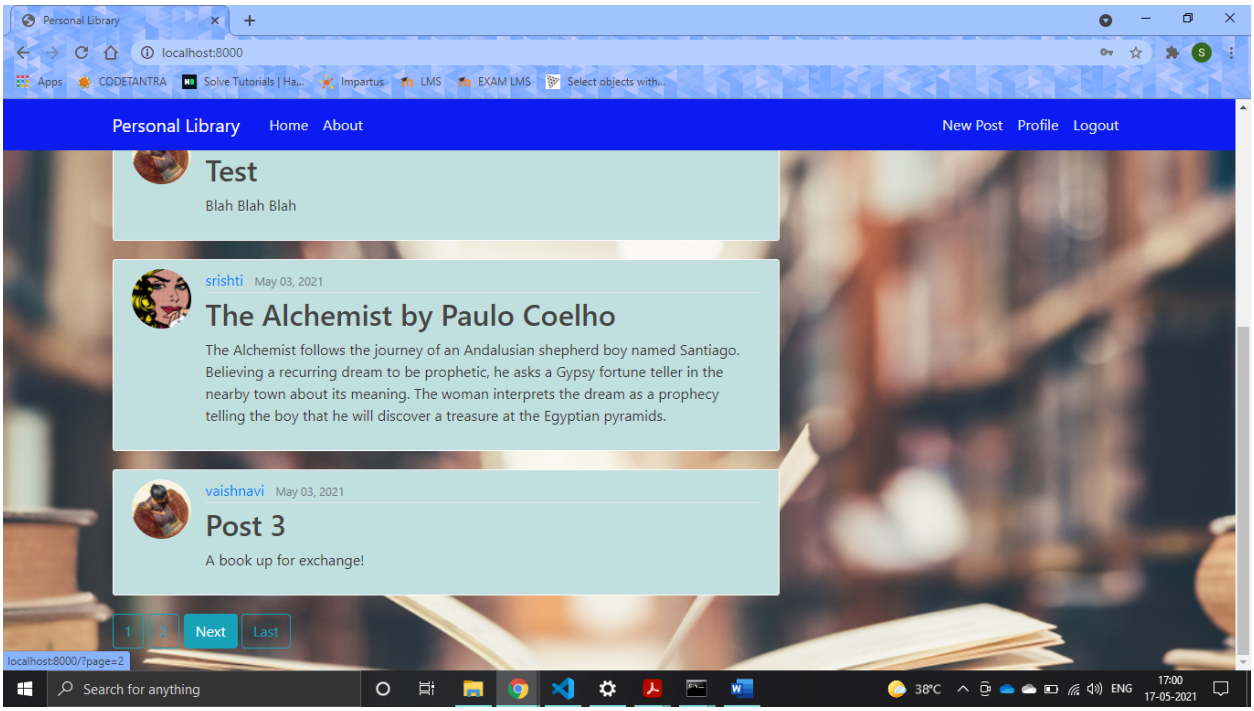
Figure 14: kibana dashboard

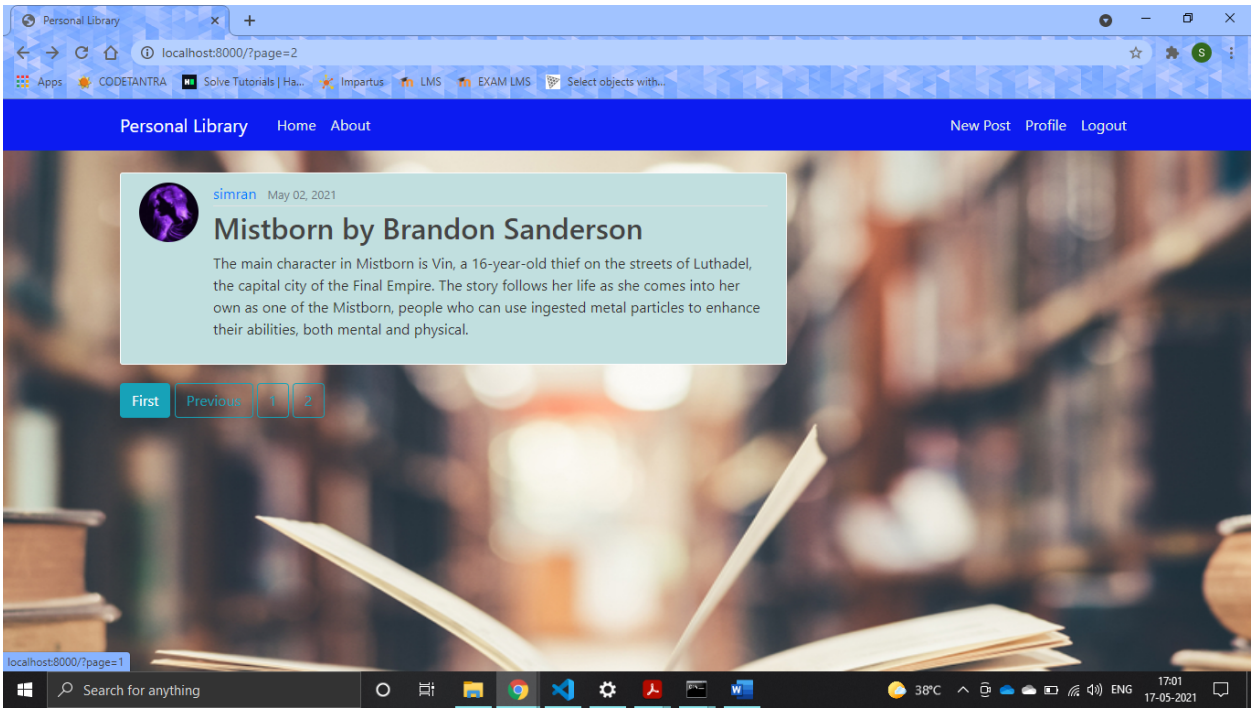
12 SCREENSHOTS OF THE RUNNING APPLICATION

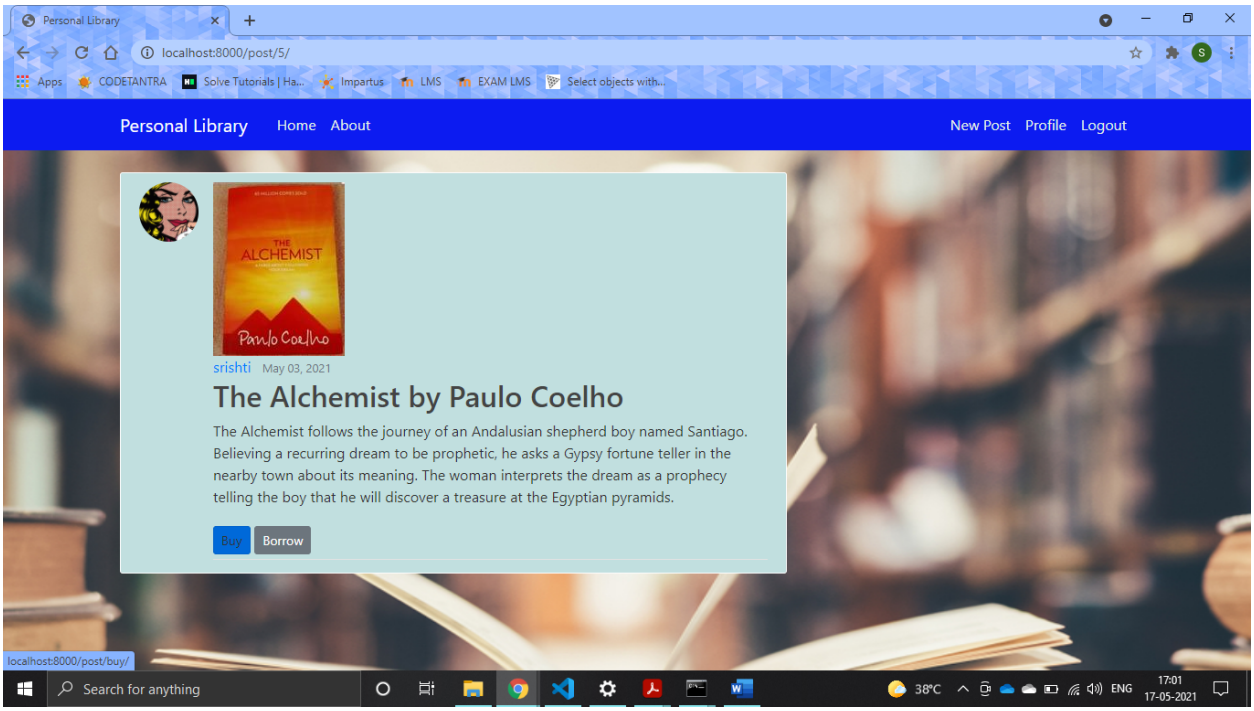


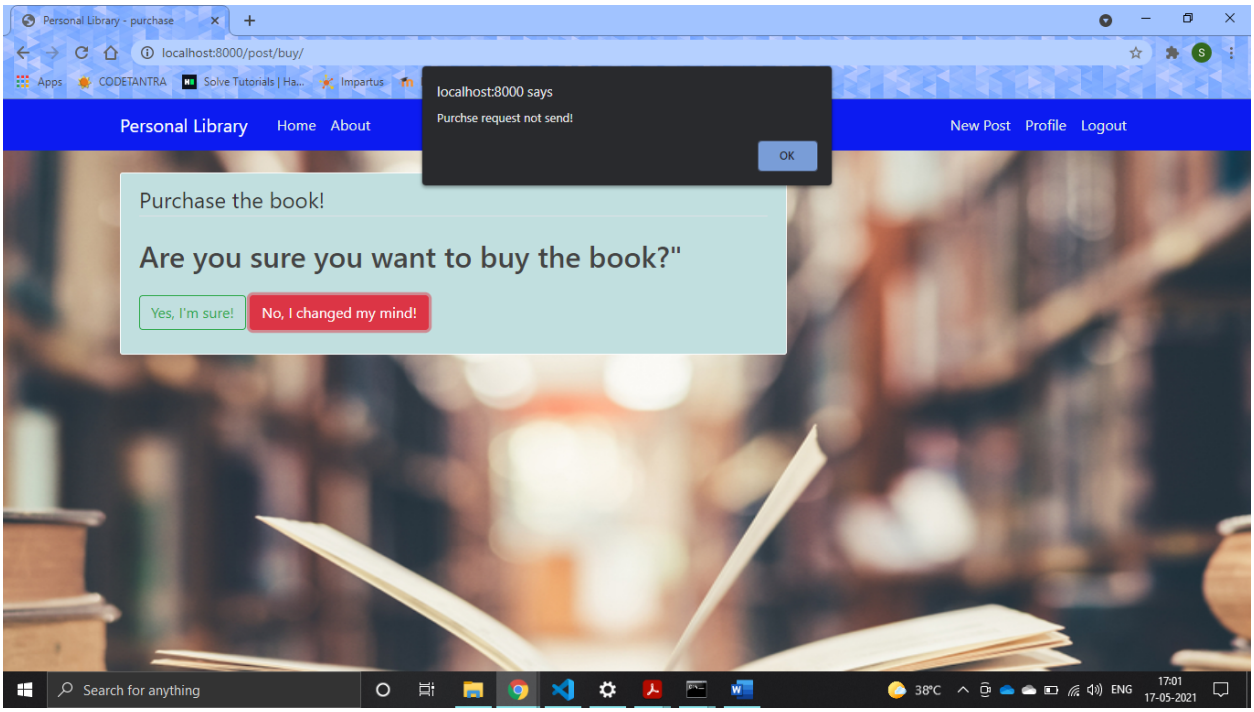

Personal Library - About
localhost:8000/about/
Personal Library
Home
About
Login
Register
Personal Library
Our library management service caters to both home and small organizational libraries. Our online software lets you create multiple libraries, catalog books, create tags, leave notes, buy/borrow, and much more. Personal Library is the best system for cataloging your books available online.
Search for anything

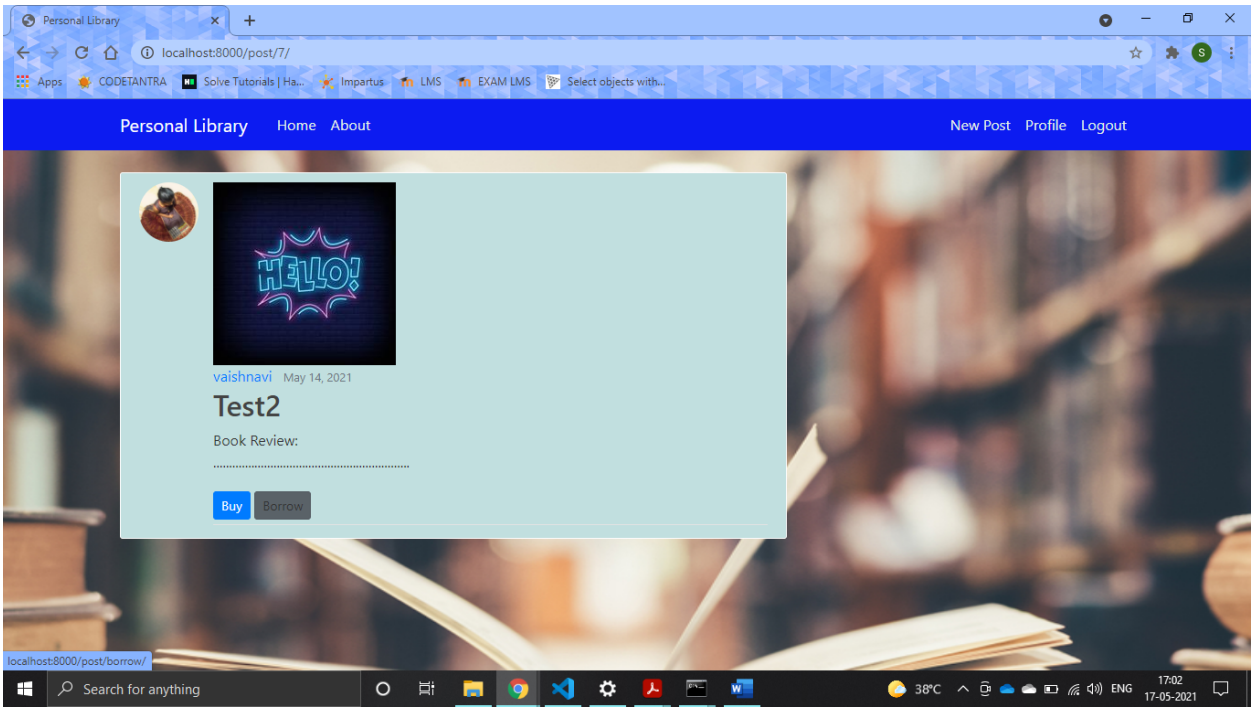

Personal Library
localhost:8000/login/
Personal Library
Home
About
Login
Register
Log In
Username*
Password*
Login
Need an account? Sign up!
Search for anything

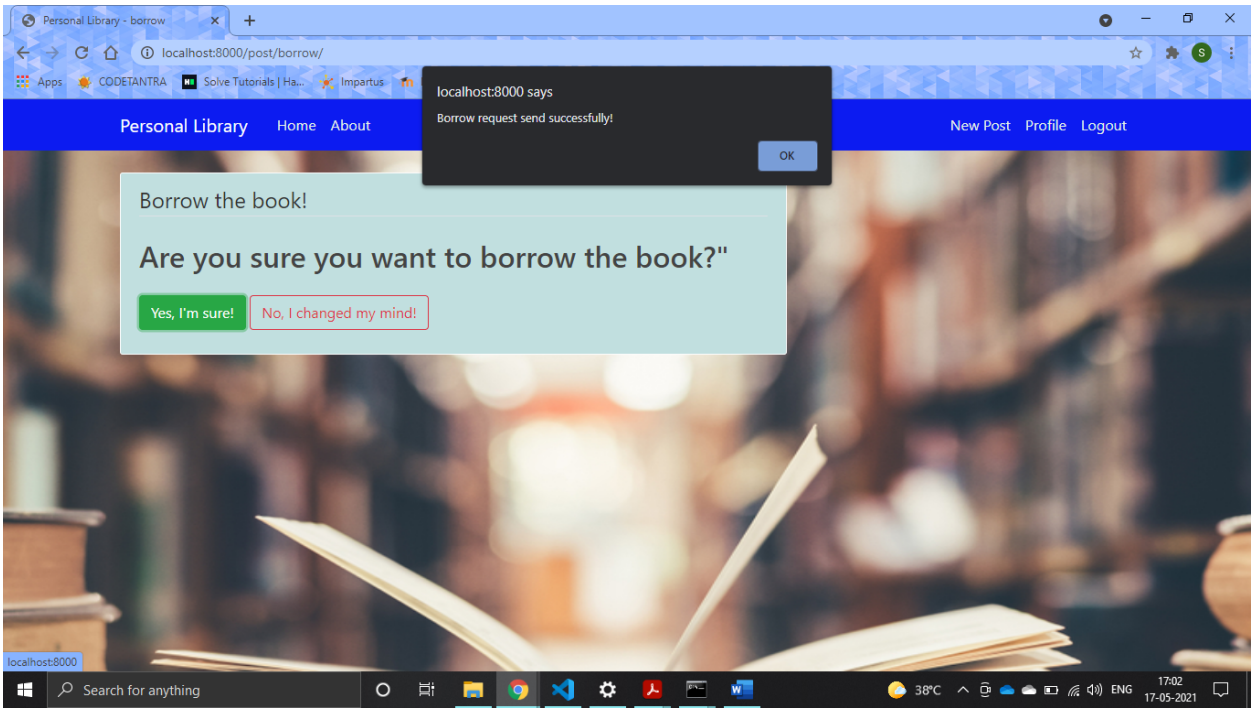

Personal Library
localhost:8000
Personal Library
Home
About
New Post
Profile
Logout
Test
Blah Blah Blah
srishti
May 03, 2021
The Alchemist by Paulo Coelho
The Alchemist follows the journey of an Andalusian shepherd boy named Santiago. Believing a recurring dream to be prophetic, he asks a Gypsy fortune teller in the nearby town about its meaning. The woman interprets the dream as a prophecy telling the boy that he will discover a treasure at the Egyptian pyramids.
vaishnavi
May 03, 2021
Post 3
A book up for exchange!
1
2
Next
Last
localhost:8000/?page=2
Search for anything
38°C
ENG
17:00
17-05-2021

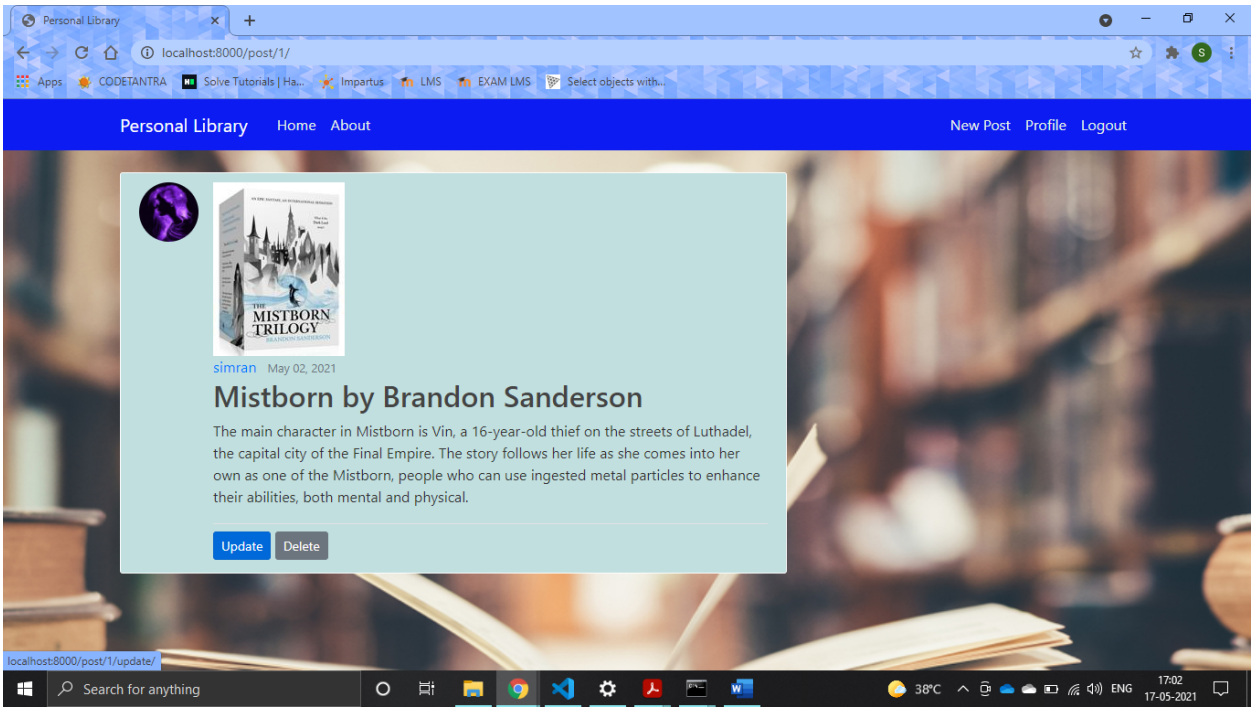

Personal Library
localhost:8000/?page=2
Personal Library
Home
About
New Post
Profile
Logout
simran
May 02, 2021
Mistborn by Brandon Sanderson
The main character in Mistborn is Vin, a 16-year-old thief on the streets of Luthadel, the capital city of the Final Empire. The story follows her life as she comes into her own as one of the Mistborn, people who can use ingested metal particles to enhance their abilities, both mental and physical.
First
Previous
1
2
localhost:8000/?page=1
Search for anything
38°C
ENG
17:01
17-05-2021

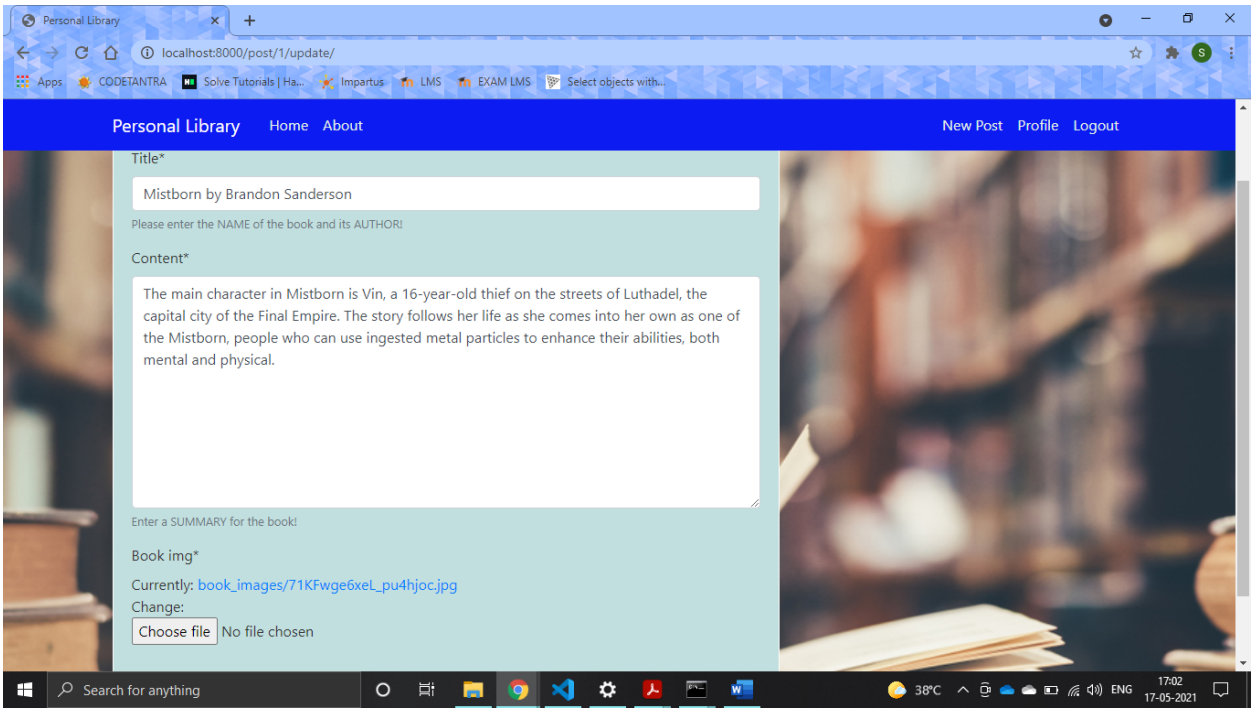

Personal Library
localhost:8000/post/5/
Personal Library
Home
About
New Post
Profile
Logout
srishti
May 03, 2021
The Alchemist by Paulo Coelho
The Alchemist follows the journey of an Andalusian shepherd boy named Santiago. Believing a recurring dream to be prophetic, he asks a Gypsy fortune teller in the nearby town about its meaning. The woman interprets the dream as a prophecy telling the boy that he will discover a treasure at the Egyptian pyramids.
Buy
Borrow
localhost:8000/post/buy/
Search for anything
38°C
ENG
17:01
17-05-2021

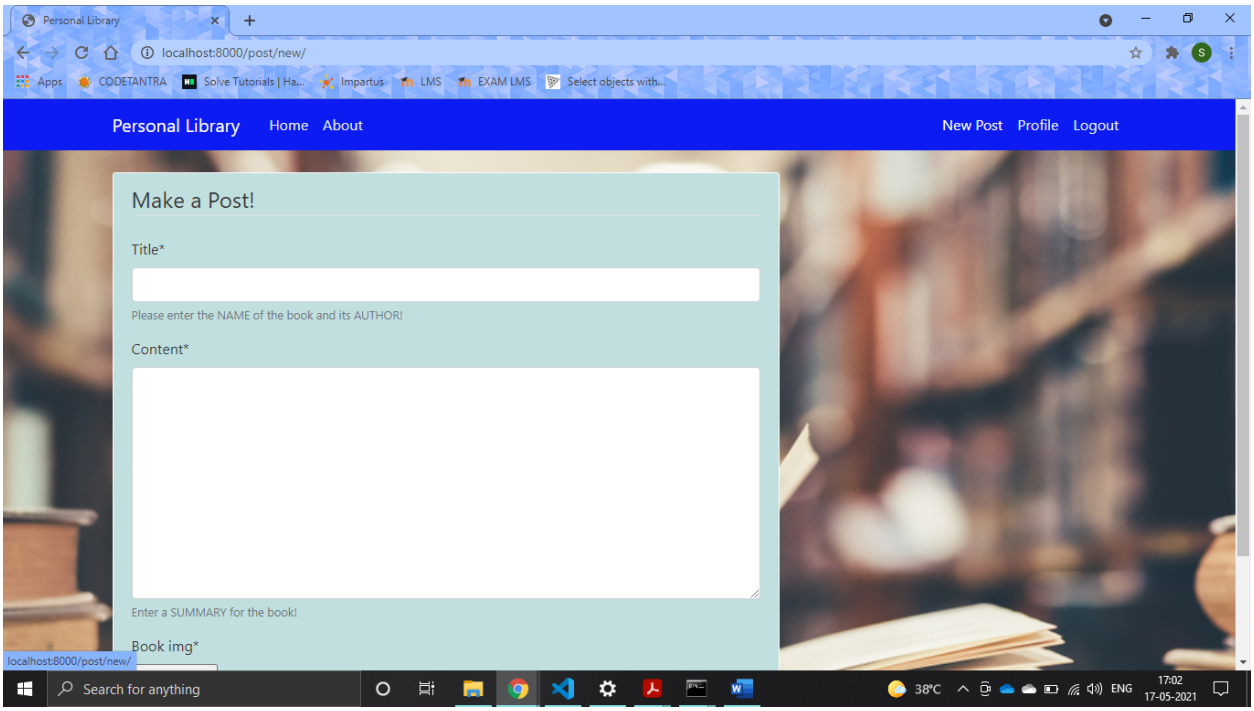

Personal Library - purchase
localhost:8000/post/buy/
localhost:8000 says
Purchse request not send!
OK
Personal Library
Home
About
New Post
Profile
Logout
Purchase the book!
Are you sure you want to buy the book?"
Yes, I'm sure!
No, I changed my mind!
Search for anything
38°C
ENG
17:01
17-05-2021

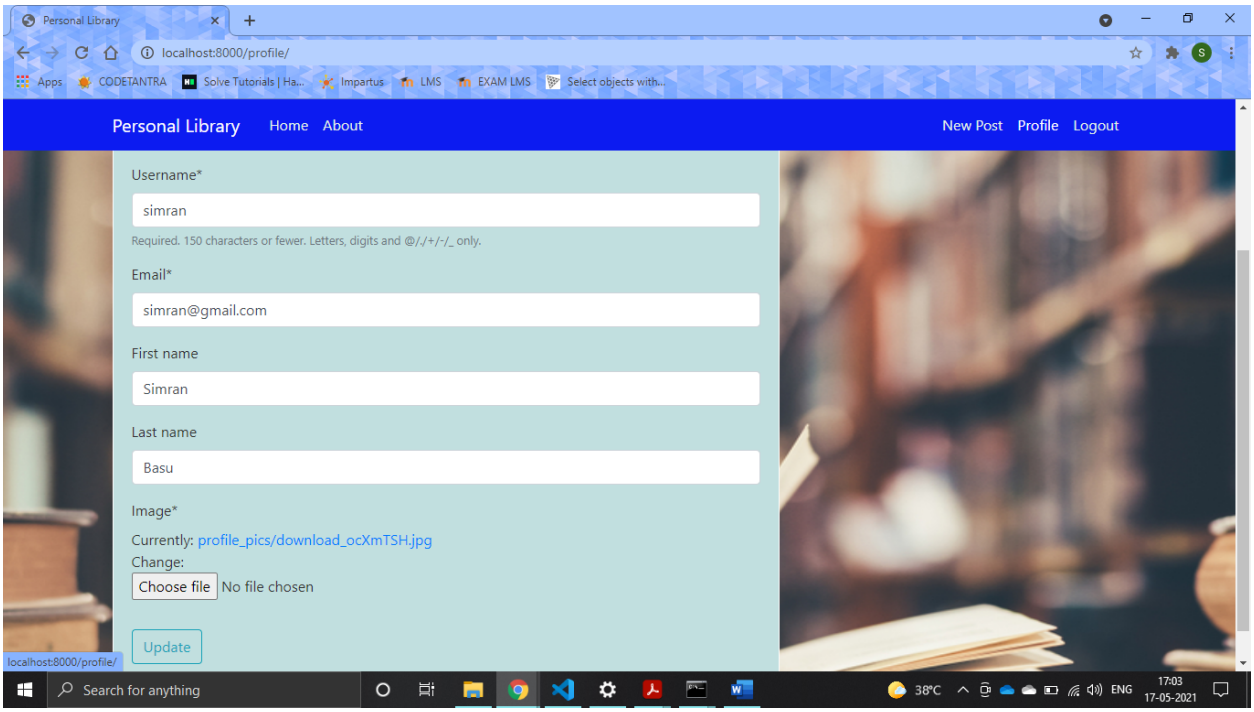

Personal Library
localhost:8000/post/7/
Personal Library
Home
About
New Post
Profile
Logout
HELLO!
vaishnavi
May 14, 2021
Test2
Book Review:
Buy
Borrow
localhost:8000/post/borrow/
Search for anything
38°C
ENG
17:02
17-05-2021

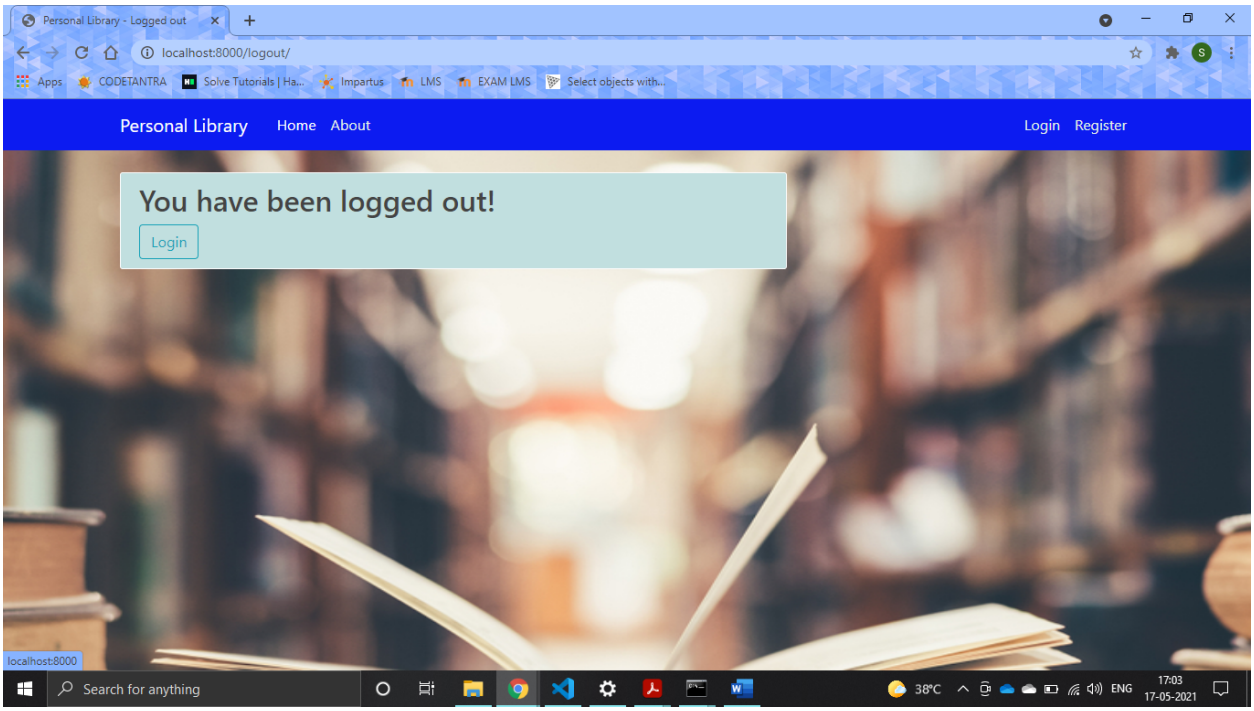

Personal Library - borrow
localhost:8000/post/borrow/
localhost:8000 says
Borrow request send successfully!
OK
Personal Library
Home
About
New Post
Profile
Logout
Borrow the book!
Are you sure you want to borrow the book?"
Yes, I'm sure!
No, I changed my mind!
localhost:8000
Search for anything
38°C
ENG
17:02
17-05-2021


Personal Library
localhost:8000/post/1/
Personal Library
Home
About
New Post
Profile
Logout
simran
May 02, 2021
Mistborn by Brandon Sanderson
The main character in Mistborn is Vin, a 16-year-old thief on the streets of Luthadel, the capital city of the Final Empire. The story follows her life as she comes into her own as one of the Mistborn, people who can use ingested metal particles to enhance their abilities, both mental and physical.
Update
Delete
localhost:8000/post/1/update/
Search for anything
38°C
ENG
17:02
17-05-2021


Personal Library
localhost:8000/post/1/update/
Personal Library
Home
About
New Post
Profile
Logout
Title*
Mistborn by Brandon Sanderson
Please enter the NAME of the book and its AUTHOR!
Content*
The main character in Mistborn is Vin, a 16-year-old thief on the streets of Luthadel, the capital city of the Final Empire. The story follows her life as she comes into her own as one of the Mistborn, people who can use ingested metal particles to enhance their abilities, both mental and physical.
Enter a SUMMARY for the book!
Book img*
Currently: book_images/71KFwge6xeL_pu4hjoc.jpg
Change:
Choose file
No file chosen
Search for anything
38°C
ENG
17:02
17-05-2021


Personal Library
localhost:8000/post/new/
Personal Library
Home
About
New Post
Profile
Logout
Make a Post!
Title*
Please enter the NAME of the book and its AUTHOR!
Content*
Enter a SUMMARY for the book!
Book img*
Search for anything
38°C
ENG
17:02
17-05-2021


Personal Library
localhost:8000/profile/
Personal Library
Home
About
New Post
Profile
Logout
Username*
simran
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Email*
simran@gmail.com
First name
Simran
Last name
Basu
Image*
Currently: profile_pics/download_ocXmTSH.jpg
Change:
Choose file
No file chosen
Update
Search for anything
38°C
ENG
17:03
17-05-2021



13 SCOPE FOR FUTURE WORK

13.1 Online Book Clubs

Book club is for a group of users who want to read and discuss a book with other readers. We would like to implement that feature using message queue under a topic.

13.2 Permission Approval

We would like to implement a permission approval system for accepting purchase or borrow request when user logs in.

14 CONCLUSION

We successfully build a online library platform for all the readers who want to stay connected to reader community using DevOps tools. The tools we used are: GitHub, Jenkins, Docker, Rundeck. These tools are integrated using Jenkins. The entire pipeline has been automated. For deployment of the entire project , it's taking a minimum of 19 minutes and maximum of half an hour.
The Devops methodology and lifecycle tools prove to be better than the Agile methodology in terms of technical, cultural and business benefits. By minimizing friction between independent teams, DevOps enables a collaborative approach for enterprise software

development and delivery that reflects the needs of the entire application lifecycle for today's modern enterprises.
Thus, we can develop, test, deploy and monitor the application easily.

REFERENCES

[1] https://docs.djangoproject.com/en/3.2/intro/reusable-apps/.

[2] https://docs.djangoproject.com/en/3.2/topics/logging/.

[3] https://semaphoreci.com/community/tutorials/dockerizing-a-python-django-web-application.

[4] https://pypi.org/project/django-ansible/.

[5] https://docs.djangoproject.com/en/3.2/topics/testing/.