

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY BANGALORE

SOFTWARE PRODUCTION ENGINEERING CS 816

Lender-Borrower interaction with Credit Risk Analysis

Abhinav Dhoka
MT2023062
abhinav.Dhoka@iiitb.ac.in

Shivam Jaiswal
MT2023112
shivam.Jaiswal@iiitb.ac.in



Lender-Borrower interaction with Credit Risk Analysis

Introduction	2
Features	2
Devops	3-4
Tools used	4
Installation	5-10
Steps to Follow	11
Jenkins	11
Jenkins Pipeline	12-14
Docker	15-18
Ansible	19
Pipeline Build	19
Application	20
Summary	26
References	27
Github Repository	27
DockerHub Repository	27

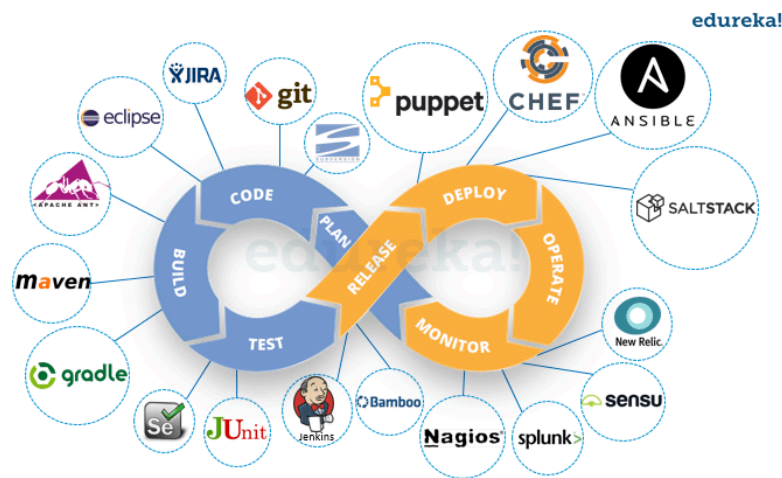
Introduction

At FairTrade, we offer a platform where visitors have the opportunity to register as either lenders or borrowers. To ensure trust and security, FairTrade conducts a thorough verification process for all registered borrowers. This process evaluates a wide range of information, including personal, financial, and professional details, to generate a Repayment Score. This score is a reflection of a borrower's creditworthiness and repayment capacity, providing lenders with a reliable metric to assess potential risks before extending a loan. Consequently, lenders can easily identify and engage with borrowers who meet their criteria for credit reliability, facilitating a smooth and secure lending process.

Features

- LogIn
- SignUp
- Dashboard
- About Page
- Contact us

DevOps



DevOps (Development and Operations) is a set of practices that combines software development (Dev) and IT operations (Ops) to enhance collaboration, streamline processes, and improve the overall efficiency and

quality of software development and deployment. While DevOps is not exclusive to any particular type of website or application, it can bring several benefits when developing a crowdfunding website:

- 1. Faster and Continuous Delivery:** DevOps promotes the use of automation, continuous integration, and continuous delivery (CI/CD) practices. This allows for faster and more frequent releases of new features, bug fixes, and updates to the crowdfunding website, ensuring a smooth and seamless user experience.
- 2. Scalability and Performance:** Crowdfunding websites often experience high traffic and unpredictable spikes in user activity, especially during campaigns or fundraising events. DevOps principles help ensure that the website infrastructure is scalable, allowing it to handle increased traffic and maintain optimal performance.
- 3. Collaboration and Communication:** DevOps fosters better collaboration and communication between development teams, operations teams, and other stakeholders involved in the crowdfunding website's development and maintenance. This helps in

aligning goals, resolving issues more efficiently, and delivering a better product.

4. **Continuous Monitoring and Feedback:** DevOps emphasizes continuous monitoring of the website's performance, user behavior, and other relevant metrics. This feedback loop helps identify issues, bottlenecks, or vulnerabilities early on, enabling proactive measures to improve the website's security, reliability, and user experience.
5. **Infrastructure as Code (IaC):** DevOps encourages the use of infrastructure as code, where the entire infrastructure, including servers, databases, and other resources, is defined and managed through code. This enables easier provisioning, configuration, and replication of the infrastructure, ensuring consistency and reducing the risk of errors or misconfigurations.
6. **Agile Development:** DevOps complements agile development methodologies by promoting iterative development, frequent deployments, and rapid feedback cycles. This enables crowdfunding website development teams to adapt to changing requirements, quickly incorporate user feedback, and deliver value to users in a timely manner.

Tools Used

- **Source control management tool:** Git and GitHub.
- **Editor for the Project:** VsCode
- **Building and packaging the code:** pip
- **Continuous Integration:** Jenkins
- **Containerising the code:** Docker, Docker Compose and Docker Hub
- **Configuration Management and Continuous Deployment:** Ansible
- **Monitoring:** Elasticsearch (ELK-stack)

Installation

- Code editor - VsCode
- **Jenkins (Continuous Integration) :**

Jenkins is an open-source automation server that is widely used for continuous integration and continuous delivery (CI/CD) of software projects. Jenkins is a Java-based web application that runs on a server and listens for incoming requests. When a user or a tool triggers a build, Jenkins will automatically pull the latest code from a source code repository, compile the code, run tests, and then deploy the code to a testing or staging environment. If the tests pass, Jenkins will automatically deploy the code to production.

Installation:

```
$ wget -q -O -  
https://pkg.jenkins.io/debian/jenkins.io.key |  
apt-key add -
```

```
$ sh -c 'echo deb  
http://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

```
$ sudo apt update
```

```
$ sudo apt install Jenkins
```

Starting Jenkins

```
$ service jenkins start
```

Check the status of Jenkins

```
$ service jenkins status
```

Jenkins runs on port 8080 by default, in order to use Jenkins, open `http://localhost:8080/`

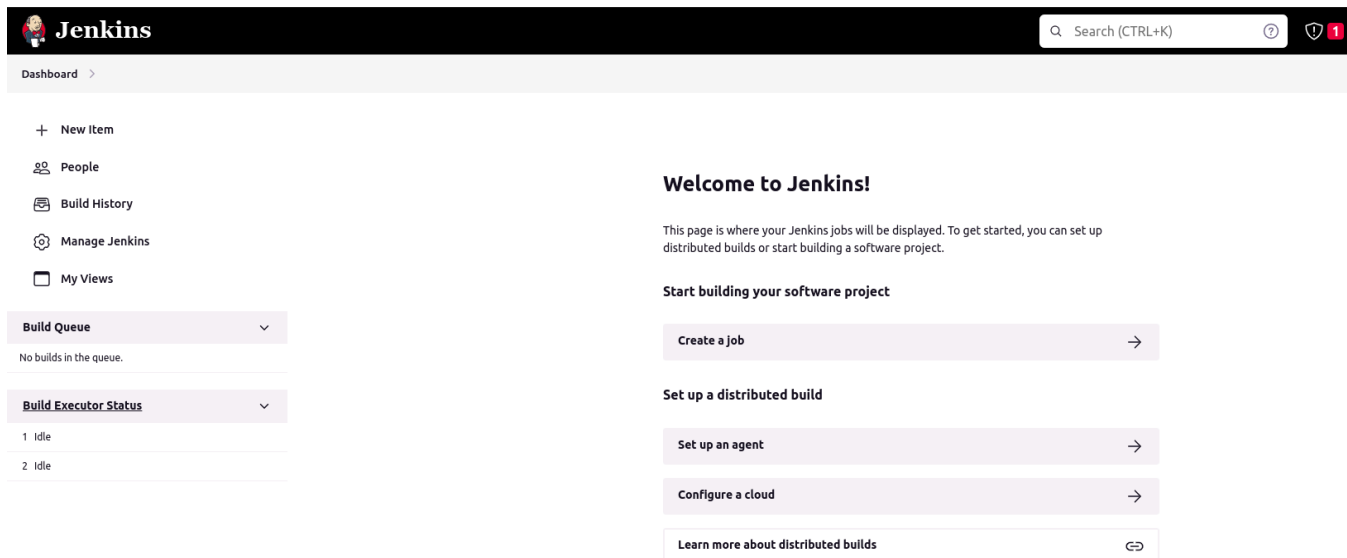
Setting up Jenkins

Use the password provided by the common below to unlock jenkins

```
$ cat  
/var/lib/jenkins/secrets/initialAdminPassword
```

Install the suggested plugins and it's ready to use.

You'll find the dashboard as shown below



The screenshot shows the Jenkins dashboard. At the top, there is a black header with the Jenkins logo and name on the left, a search bar with the text "Search (CTRL+K)" in the center, and a shield icon with a red "1" on the right. Below the header, the main content area is divided into two columns. The left column contains a sidebar with navigation options: "+ New Item", "People", "Build History", "Manage Jenkins", and "My Views". Below these are two expandable sections: "Build Queue" (showing "No builds in the queue.") and "Build Executor Status" (showing two "Idle" executors). The right column features a "Welcome to Jenkins!" section with a brief introduction and a "Start building your software project" heading. Under this heading, there are four buttons: "Create a job", "Set up a distributed build", "Set up an agent", and "Configure a cloud", each with a right-pointing arrow. At the bottom of this section is a link "Learn more about distributed builds" with an external link icon.

- **Git(Source Code Management):**

Git is a popular distributed version control system (DVCS) used for software development. Git allows developers to track changes to their source code over time, collaborate with other developers, and manage different versions of their code. It does this by creating a repository that stores all the changes made to the code, along with information about who made the changes and when they were made.

Installation

```
$ sudo apt update
```

```
$ sudo apt install git
```

```
$ git --version
```

 We can check if it's correctly installed

Commands - To create a remote repository and push a local repository to it.

Create a new repository in GitHub

To push our code to remote repo(github)

```
$ git init
```

 Initialized empty git repo in local system

```
$ git add .
```

```
$ git remote add origin <github repo URL>
```

```
$ git commit -m "Message here"
```

 You can write a message every time you commit

```
$ git push origin master
```

 Pushes local repo to the master
branch in remote repo

- **Docker:**

Docker is a containerization platform that enables developers to package their applications and dependencies into portable containers. These containers are isolated from the host system and can be easily moved between environments, such as from a developer's laptop to a staging server, or from a staging server to a production environment.

Installation

```
$ sudo apt-get remove docker docker-engine  
docker.io
```

```
$ sudo apt-get update
```

```
$ sudo apt install docker.io
```

```
$ sudo snap install docker
```

```
$ sudo usermod -aG docker ${USER}
```

Commands

```
$ docker pull <image_name>
```

 To pull image and if not found on local machine pulls from dockerhub

```
$ docker images
```

 To list images downloaded to your computer

```
$ docker run -i -t --name <container_name>  
<image_name>    To run image
```

```
$ docker ps -a    To list active containers
```

- **DockerCompose**

```
$ sudo curl -L  
"https://github.com/docker/compose/releases/downlo  
ad/1.29.2/docker-compose-$(uname  
-s)-$(uname -m)" -o /usr/local/bin/docker-compose  
  
$ sudo chmod +x /usr/local/bin/docker-compose
```

- **Ansible:**

Ansible is an open-source automation tool that allows you to automate IT tasks such as configuration management, application deployment, and task automation. Ansible uses a declarative language to define the desired state of the system, which makes it easy to write and understand automation scripts. It also has a modular architecture that allows you to extend its functionality and integrate it with other tools. One of the main benefits of using Ansible is that it is agentless, which means that it does not require any software to be installed on the remote systems you want to manage. Instead, it uses SSH to communicate with the systems and execute the necessary tasks.

Installation

```
$ sudo apt install openssh-server
```

```
$ ssh-keygen -t rsa
```

```
$ sudo apt update
```

```
$ sudo apt install ansible
```

```
$ ansible -m setup localhost | grep  
ansible_python_version
```

Ansible Inventory

A list of hosts that Ansible will handle is defined by an inventory. These hosts can also be managed jointly and assigned to groups.

Ansible Playbook

List of all tasks

Commands

```
$ ansible all --list-hosts    To find hosts
```

```
$ ansible-playbook playbook_file_name.yml    To run the  
playbook
```

```
$ ansible-playbook --syntax-check  
playbook_file_name.yml    To verify the playbook
```

```
$ ansible-playbook -C playbook_file_name.yml    To do  
the dry run
```

Steps to follow

Jenkins

1. Pulling the code from github
 2. Creating a docker image for the server and client
 3. Testing our application
 4. Publishing both the images to docker hub
 5. Deleting both the images from the local machine
 6. Pulling the images of both server and client from the docker hub
 7. Deploying them to a remote server using ansible,
- are all steps in the below stated jenkins script.

After writing the pipeline, hit Apply and Save.

Jenkins Pipeline

```
1 pipeline {
2     agent any
3
4     environment {
5         DOCKERHUB_CREDENTIALS = credentials('dockerhub_credentials')
6         DOCKERHUB_REPO_FLASK = 'shivam424/flask'
7         DOCKERHUB_REPO_DJANGO = 'shivam424/django'
8     }
9
10    stages {
11        stage('Checkout') {
12            steps {
13                git branch: 'master',
14                    url: 'https://github.com/abhinav1829/FairTrade.git'
15            }
16        }
17
18        stage('Test') {
19            steps {
20                sh '''#!/bin/bash -ex
21                    python3 -m venv venv
22                    source venv/bin/activate
23                    pip install -r requirements.txt requests
24                    python3 manage.py test
25                '''
26            }
27        }
28
29        stage('Docker Login') {
30            steps {
31                script {
32                    docker.withRegistry('https://index.docker.io/v1/', 'dockerhub_credentials') {
33
34                    }
35                }
36            }
37        }
38
39        stage('Build Django Docker Image') {
40            steps {
41                script {
42                    docker.withRegistry('https://index.docker.io/v1/', 'dockerhub_credentials') {
43                        docker.build("${DOCKERHUB_REPO_DJANGO}:latest", '-f dockerFileApp .').push('latest')
44                    }
45                }
46            }
47        }
48    }
```

```

48
49     stage('Build Flask ML Docker Image') {
50         steps {
51             script {
52                 docker.withRegistry('https://index.docker.io/v1/', 'dockerhub_credentials') {
53                     docker.build("${DOCKERHUB_REPO_FLASK}:latest", '-f dockerFileRepayment .').push('latest')
54                 }
55             }
56         }
57     }
58
59     stage('Verify Ansible Installation') {
60         steps {
61             sh 'ansible --version'
62         }
63     }
64
65     stage('Run Ansible Playbook') {
66         steps {
67             ansiblePlaybook(
68                 playbook: 'deploy_docker_compose.yml',
69                 inventory: 'inventory'
70             )
71         }
72     }
73 }
74 }

```

Explanation:

1. Pipeline Definition (**pipeline {}**):

- **agent any**: This pipeline can run on any available Jenkins agent (node).

2. Environment Variables (**environment {}**):

- **DOCKERHUB_CREDENTIALS**: Refers to Jenkins credentials (`dockerhub_credentials`) storing Docker Hub credentials.
- **DOCKERHUB_REPO_FLASK**: Specifies the Docker Hub repository for the Flask application.
- **DOCKERHUB_REPO_DJANGO**: Specifies the Docker Hub repository for the Django application.

3. Stages (**stages {}**):

- **Checkout**:

- Uses `git` to clone the `master` branch of the `FairTrade` repository from GitHub.
- **Test:**
 - Sets up a Python virtual environment (`venv`).
 - Activates the virtual environment (`source venv/bin/activate`).
 - Installs Python dependencies listed in `requirements.txt` including `requests`.
 - Executes Django tests using `python3 manage.py test`.
- **Docker Login:**
 - Uses Docker credentials to log in to Docker Hub.
- **Build Django Docker Image:**
 - Uses Docker to build a Docker image (`${DOCKERHUB_REPO_DJANGO}:latest`) from the `dockerFileApp` Dockerfile.
 - Pushes the built image with tag `latest` to Docker Hub.
- **Build Flask ML Docker Image:**
 - Uses Docker to build a Docker image (`${DOCKERHUB_REPO_FLASK}:latest`) from the `dockerFileRepayment` Dockerfile.
 - Pushes the built image with tag `latest` to Docker Hub.
- **Verify Ansible Installation:**
 - Runs `ansible --version` to verify that Ansible is installed and operational.
- **Run Ansible Playbook:**
 - Executes an Ansible playbook (`deploy_docker_compose.yml`) with an inventory file (`inventory`).

Docker

DockerFileApp for App

```
1 FROM python:3.9
2
3 WORKDIR /app
4
5 ENV PYTHONUNBUFFERED=1
6
7 RUN pip install django requests
8
9 # create root directory for our project in the container
10 RUN mkdir /project
11
12 # Set the working directory to /music_service
13 WORKDIR /project
14
15 # Copy the current directory contents into the container at /music_service
16 # ADD . /project/
17 COPY ./assets /project/assets
18 COPY ./Auth /project/Auth
19 COPY ./Borrower /project/Borrower
20 COPY ./Lender /project/Lender
21 COPY ./FairTrade /project/FairTrade
22 COPY current.json /project
23 COPY current.py /project
24 COPY manage.py /project
25 COPY requirements.txt /project
26 COPY db.sqlite3 /project
27 COPY tests.py /project
28
29 RUN pip install -r requirements.txt
30 RUN python manage.py makemigrations
31 RUN python manage.py migrate
32 EXPOSE 8000
33
34 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

DockerFileRepayment for flask_ML

```
1 # Base image - Choose a lightweight Python image
2 FROM python:3.9
3
4 # Create working directory
5 RUN mkdir /ml
6
7 # Set the working directory to /music_service
8 WORKDIR /ml
9
10 # Copy the current directory contents into the container at /music_service
11 # ADD . /ml/
12 COPY model.pkl /ml/
13 COPY scalar.pkl /ml/
14 COPY repayment-score.py /ml/
15
16 # Install dependencies
17 RUN pip install flask numpy scikit-learn==1.3.0
18
19 # Copy your Flask application files
20
21 # Expose the port Flask will use
22 EXPOSE 5000
23
24 # Command to start the Flask app
25 CMD ["python", "repayment-score.py", "--host=0.0.0.0"]
```

Docker Compose File for App

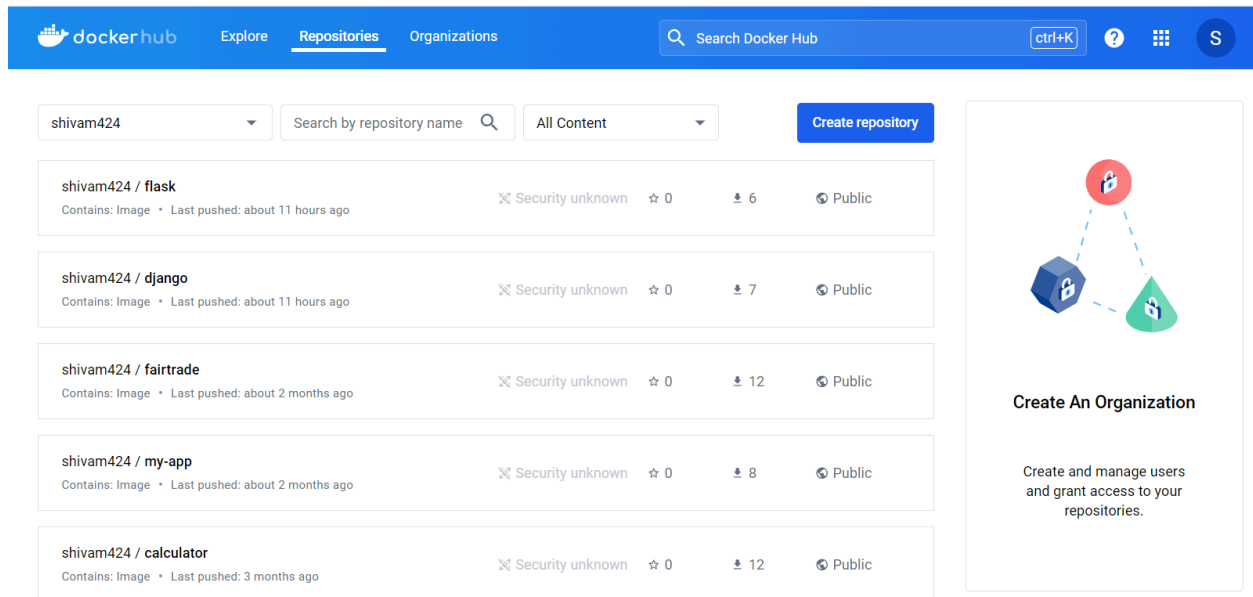
```
1  version: '3'
2
3  services:
4    flask_ml:
5      container_name: flask_ml
6      image: shivam424/flask:latest
7      ports:
8        - "5000:5000"
9      networks:
10       - elk
11
12    django:
13      container_name: django
14      image: shivam424/django:latest
15      ports:
16        - "8000:8000"
17      volumes:
18        - sqlite_data:/data
19      depends_on:
20        - flask_ml
21      networks:
22        - elk
23
24    Elasticsearch:
25      image: elasticsearch:7.16.2
26      container_name: elasticsearch
27      restart: always
28      volumes:
29        - elastic_data:/usr/share/elasticsearch/data/
30      environment:
31        ES_JAVA_OPTS: "-Xmx256m -Xms256m"
32        discovery.type: single-node
33      ports:
34        - "9200:9200"
35        - "9300:9300"
36      networks:
37        - elk
38
39    Logstash:
40      image: logstash:7.16.2
41      container_name: logstash
42      restart: always
43      volumes:
44        - ./logstash:/logstash_dir
45      command: logstash -f /logstash_dir/logstash.conf
46      depends_on:
47        - Elasticsearch
48      ports:
49        - "9600:9600"
50      environment:
51        LS_JAVA_OPTS: "-Xmx256m -Xms256m"
```

```

51     LS_JAVA_OPTS: "-Xmx256m -Xms256m"
52     networks:
53     - elk
54
55     kibana:
56     image: kibana:7.16.2
57     container_name: kibana
58     restart: always
59     ports:
60     - "5601:5601"
61     environment:
62     - ELASTICSEARCH_URL=http://elasticsearch:9200
63     depends_on:
64     - Elasticsearch
65     networks:
66     - elk
67
68     networks:
69     elk:
70
71     volumes:
72     sqlite_data:
73     elastic_data: {}

```

We can see that our images are available on the dockerhub after the pipeline has been built.



The screenshot shows the Docker Hub interface for the user 'shivam424'. The 'Repos' tab is active, displaying a list of repositories. The repositories listed are:

- shivam424 / flask**: Contains: Image • Last pushed: about 11 hours ago. Security unknown, 0 stars, 6 downloads, Public.
- shivam424 / django**: Contains: Image • Last pushed: about 11 hours ago. Security unknown, 0 stars, 7 downloads, Public.
- shivam424 / fairtrade**: Contains: Image • Last pushed: about 2 months ago. Security unknown, 0 stars, 12 downloads, Public.
- shivam424 / my-app**: Contains: Image • Last pushed: about 2 months ago. Security unknown, 0 stars, 8 downloads, Public.
- shivam424 / calculator**: Contains: Image • Last pushed: 3 months ago. Security unknown, 0 stars, 12 downloads, Public.

On the right side of the page, there is a 'Create An Organization' banner. It features a diagram of three interconnected nodes (a red circle, a blue cube, and a green triangle) and the text: 'Create and manage users and grant access to your repositories.'

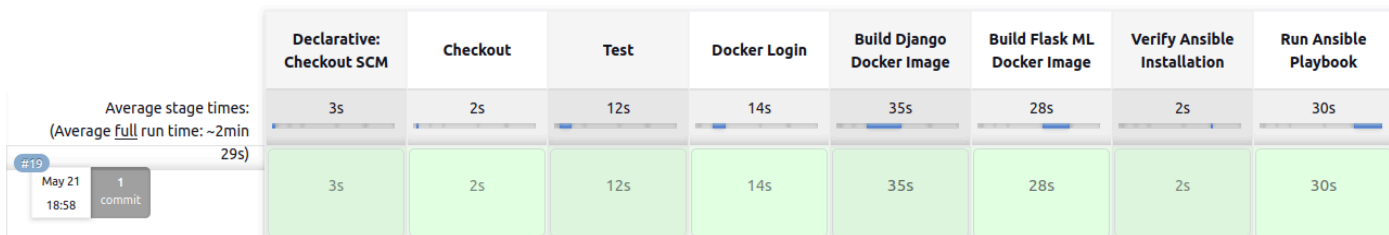
Ansible

Ansible Playbook

```
1 ---
2 - name: Run Docker Compose
3   hosts: localhost
4   gather_facts: no
5   tasks:
6     - name: Run Docker Compose
7       command: /usr/local/bin/docker-compose up -d
8       args:
9         chdir: /home/shivam/Desktop/SPE/FairTrade
10
11 # ansible-playbook deploy_docker_compose.yml --ask-become-pass
```

Pipeline build

Stage View



Application

Main Page

FairTrade [About](#) [Contact Us](#)

[Register as Lender](#) [Register as Borrower](#)

Username:

Password:

[Submit](#)

FAIRTRADE

On FairTrade, visitors, as per their requirement, can register as lenders or borrowers. FairTrade undertakes a comprehensive verification process based on personal, financial and professional information provided of all its registered Borrowers and Lenders.

LINKS

[About](#)

© 2020 Copyright: [FairTrade](#)

The image is of a website with a login form and Fair Trade logo. The website is designed to help people register as lenders or borrowers. The login form is located in the center of the page and has two input fields: username and password.

The website is designed to be easy to use and navigate. The text is clear and concise, and the layout is uncluttered. The website is also designed to be visually appealing.

When you click on **Register as Borrower**

FairTrade [About](#) [Contact Us](#)

Username

Password

First name

Last name

Dob

Pan

Education

Capital

Income

Debt

Interest

Credit score

It requests the following information:

- **Username:** The desired username for the account.
- **Password:** A password to secure the account.
- **First name:** The borrower's first name.
- **Last name:** The borrower's last name.
- **Dob:** The borrower's date of birth.
- **Pan:** This likely refers to the Permanent Account Number, a unique 10-digit alphanumeric identifier issued by the Income Tax Department in India.

- **Education:** The borrower's level of education (options not shown in the image).
- **Capital:** The borrower's current capital (possibly in monetary terms).
- **Income:** The borrower's income (possibly annually or monthly).
- **Debt:** The borrower's existing debt.
- **Interest:** The borrower's expected interest rate (or possibly a preferred interest rate for borrowing).
- **Credit score:** The borrower's credit score.

The form also includes a "Submit" button to complete the registration process.

When you click on **Register as Lender**

[FairTrade](#) [About](#) [Contact Us](#)

Username

Password

First name

Last name

Dob

Pan

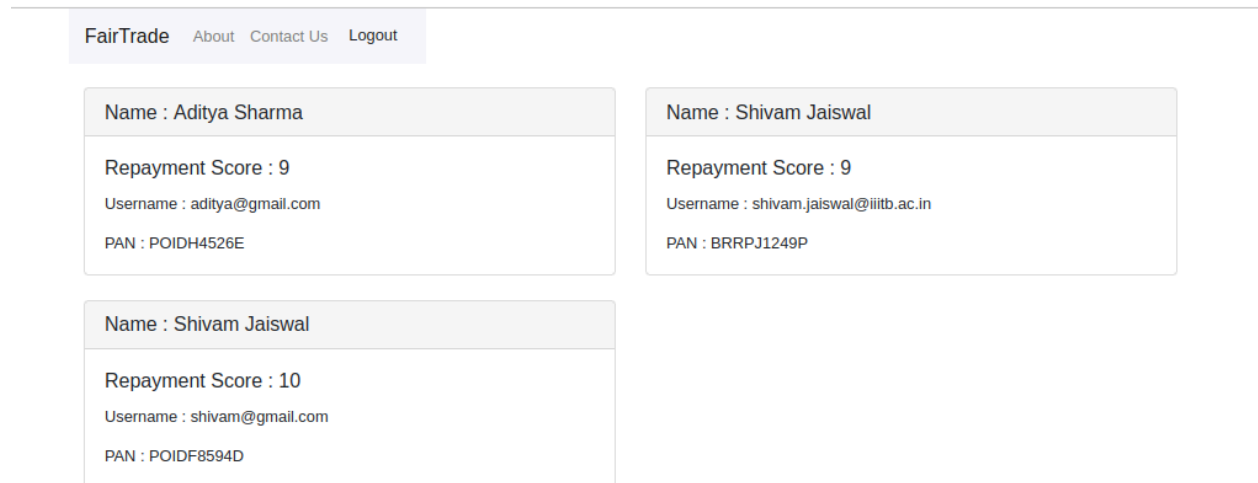
Wallet credit

The form requests the following information:

- **Username:** The desired username for the account.
- **Password:** A password to secure the account.
- **First name:** The lender's first name.
- **Last name:** The lender's last name.
- **Dob:** The lender's date of birth.
- **Pan:** This likely refers to the Permanent Account Number, a unique 10-digit alphanumeric identifier issued by the Income Tax Department in India.
- **Wallet credit:** The amount of money the lender wishes to make available for lending on the platform.

The form also includes a "Submit" button to complete the registration process.

When you Login as Lender



The image displays information about three potential borrowers on the Fair Trade platform. For each borrower, the following details are provided:

- **Name:** The full name of the borrower.
- **Repayment Score:** A numerical score (out of 10) indicating the borrower's creditworthiness or likelihood of repaying a loan.
- **Username:** The borrower's username on the platform (usually an email address).
- **PAN:** The borrower's Permanent Account Number, a unique identifier issued by the Indian Income Tax Department.

This information would be valuable for a lender on the Fair Trade platform to assess the risk associated with lending to each individual and to make informed decisions about who to lend to.

When you Login as Borrower

FairTrade About Contact Us Logout	
<p>Name : Abhinav Dhoka</p> <p>Credit available : 90000000.0</p> <p>Username : abhinavdhoka@gmail.com</p> <p>PAN : DTYPG4859D</p>	<p>Name : shivam Jaiswal</p> <p>Credit available : 5000000.0</p> <p>Username : shivam424.sj@gmail.com</p> <p>PAN : BRRPJ1249P</p>

The image shows a list of two lenders who are available in the database of the Fair Trade platform. The information provided for each lender includes:

- **Name:** The full name of the lender.
- **Credit Available:** The amount of funds the lender has available to lend to borrowers.
- **Username:** The lender's email address, which serves as their username on the platform.
- **PAN:** The lender's Permanent Account Number, a unique identifier issued by the Indian Income Tax Department.

This information would be useful for a borrower on the Fair Trade platform to identify potential lenders who have sufficient funds available to meet their borrowing needs. It may also be useful for the borrower to verify the lender's identity through their PAN.

Summary

Fair Trade is a platform designed to facilitate peer-to-peer lending. It allows individuals to register as either borrowers or lenders.

Borrowers:

- Borrowers can create profiles providing personal and financial information, including their PAN (Permanent Account Number), education level, income, and existing debts.
- They can view a list of potential lenders, including their names, the amount of credit available, and their PAN.
- Borrowers likely use this information to identify suitable lenders and potentially negotiate loan terms directly with them.

Lenders:

- Lenders create profiles indicating the amount of money they are willing to lend.
- They can view a list of potential borrowers, including their names, a repayment score, and their PAN.
- Lenders use this information to assess the creditworthiness of potential borrowers and decide who to lend to.

Overall:

- The app aims to create a more transparent and accessible lending marketplace, bypassing traditional financial institutions.
- It utilizes a scoring system to assess borrowers' creditworthiness, helping lenders make informed decisions.
- The use of PAN for identification adds a layer of security and trust to the platform.

References

- <https://idroot.us/install-elk-stack-ubuntu-22-04/>
- <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-22-04>
- https://www.youtube.com/watch?feature=shared&v=roJQ-7F_Vgg

Github Repository - <https://github.com/abhinav1829/FairTrade>

DockerHub Repository -

FrontEnd -

<https://hub.docker.com/repository/docker/shivam424/django/general>

BackEnd - <https://hub.docker.com/repository/docker/shivam424/flask>