



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 4, April 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

Building Student Portal Management System using Docker

K.Venkatesh¹, D. Humera Sulthana², A. Chandu Meghana³, S. Ajay Kumar⁴,

G. Chinna Meghana⁵

Assistant Professor, Department of Computer Science, Srinivasa Ramanujan Institute of Technology-Anantapur,

Andhra Pradesh, India¹

Student, Department of Computer Science, Srinivasa Ramanujan Institute of Technology-Anantapur,

Andhra Pradesh, India^{2,3,4,5}

ABSTRACT: Virtual machines are an abstraction of physical hardware turning one server into many servers. Being an old concept it uses the hypervisor, which allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries. Whereas open-source software are gaining popularity over licensed software and Docker is one of them. Docker has many advantages when compared with Virtual Machines environment. The demand in the market for docker containers is rapidly increasing. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. It is an open platform for developers and sys admins to build, ship, and run distributed applications. The overall objective of our work is to make the whole process of managing our applications and deploying them to live servers much easier and safer. We can edit a docker app, push the new docker image to a repository using the docker images command, then instantly run that application anywhere. Key Words: Virtual Machine, Hypervisor, Docker, Containerization.

I.INTRODUCTION

Initially our project is based on web application which is deployed in docker containers. Docker is an open-source platform that run applications and makes the process easier to develop, distribute. The applications that are built in the docker are packaged with all the supporting dependencies into a standard form called a container. These containers keep running in an isolated way on top of the operating system's kernel. The extra layer of abstraction might effect in terms of performance. This attracts developers for building micro services that will run over containers. The demand in the market for docker containers is rapidly increasing. Light weight and speed are the factors due to which docker outperforms virtual machine environment. The time required for starting and running a docker is very less when compare to virtual machine. Even though, the technologies of the container have been around for over 10 years, but docker, a generally new hopeful is right now a standout amongst the best innovations, since it accompanies new capacities that prior technologies did not have. Initially, it gives the facility to create and control containers. Besides that, applications can easily be packed into lightweight docker containers by the developer. These virtualized applications can easily be worked anywhere without any alteration. Moreover, docker can convey more virtual situations than different innovations, on the same equipment. To wrap things up, docker can easily coordinate with third-party instruments, which help to easily deploy and manage docker containers. Docker containers can easily be deployed into the cloud-based environment.

These virtualized applications can easily be worked anywhere without any alteration. Moreover, docker can convey more virtual situations than different innovations, on the same equipment. To wrap things up, docker can easily coordinate with third-party instruments, which help to easily deploy and manage docker containers. Docker containers can easily be deployed into the cloud-based environment.

Docker

Docker is an open-source platform for building, deploying, and managing containerized applications. Docker provides a facility to automates the applications when they are deployed into containers. In a Container environment where the applications are virtualized and executed, docker adds up an extra layer of deployment engine on top of it. The way that docker is designed is to give a quick and a lightweight environment where code can be run efficiently and moreover it provides an extra facility of the proficient work process to take the code from the computer for testing before

production [1]. In the year 2015, Russell confirms that it is possible to test your code and deploy it to the production working environment, Turnbull comes with the statement that anyone can begin with a docker with very easy configurations and docker binary with very easy configurations and docker binary with the linux kernel.

1. Docker Inside

There are four main components provided by docker and these are docker client and server, docker registries, docker images and docker containers [5].

1. Docker Client and Server

Docker can be explained as a client and server based application, as depicted in Figure 1.

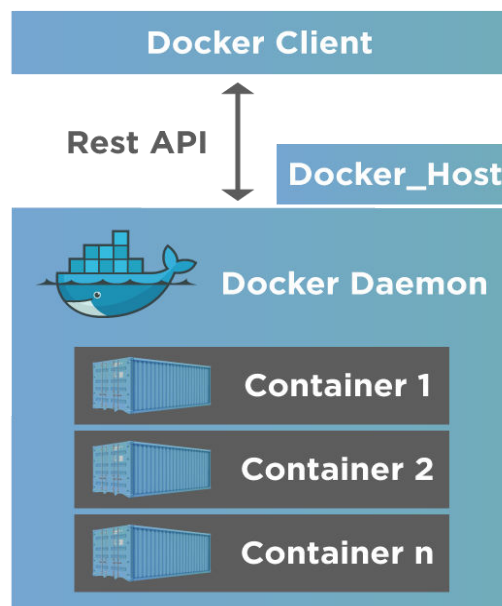


Fig 1. Docker Engine on Linux Operating System

The docker server gets the request from the docker client and then process it accordingly. The complete RESTful (Representational state transfer) API and a command line client binary are shipped by docker. Docker daemon/server and docker client can be run on the same machine or a local docker client can be connected with a remote server or daemon, which is running on another machine [1].

1.2 Docker Images

A Docker image builds containers. We can view a Docker Image as a “source code” of the containers. Images have pre-installed software, they are portable, and we can even use pre-existing images instead of building a brand new one. All this makes deployment more straightforward and faster. Here we create a docker file. The docker file contains a list of instructions when “Docker build” command is run from the bash terminal it follows all the instructions given in the docker file and builds an image. This is an automated way of building an image.

Docker images are building blocks of docker containers and serve as templates for docker containers in simpler terms.

3. Docker Registries

Docker Registries is king of a repository used to store Docker images. It works correspondingly to source code repositories where images can be pushed and pulled from a single source. Registries can be two types (public or private). Public registries are called “Docker Hub” and are similar to online code repositories like GitHub. We can also store images privately in Docker Hub and is accessible to millions of users.

4. Docker Containers

Docker image creates a docker container. Containers hold the whole kit required for an application, so the application can be run in an isolated way. For example, suppose there is an image of Ubuntu OS with SQL SERVER,

when this image is run with docker run command, then a container will be created and SQLSERVER will be running on UbuntuOS [1].

Virtual Machine vs. Docker

Virtualization is an old concept, which has been in used in cloud computing, after IaaS has been accepted as a crucial technique for system constitution, resource provisioning, and multi-tenancy. Virtualized resources play the main role in solving the problems using the core technique of cloud computing [1]. A Virtual Machine, is created to perform tasks that if otherwise performed directly on the host environment, may prove to be risky. The Figure 2 shows the architecture of the Virtual machine.

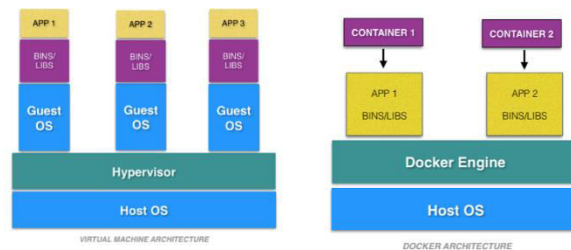


Fig 2. Virtual Machine and Docker Architecture

Hypervisor is lying between host and guest operating system inside each Virtual machine. A guest OS can be any OS, like Linux or Windows, irrespective of the host OS. Where it is a virtual platform and it handles more than one operating system in the server. It works between the operating system and CPU.

In Docker Containers host on a single physical server with a host OS which shares among them. Sharing the host OS between containers makes them light and increases the boot time. Docker containers are considered suitable to run multiple applications over a single OS kernel, whereas, virtual machines are needed if the applications or services required to run on different OS.

II. LITERATURE SURVEY

[1] Babak Bashari Rad, Harrison John Bhatti, Mohammad Ahmadi, Docker and Kubernetes work at different levels. Docker used in the IT software, makes use of the Containerization technology that helps to generate the different types of Linux containers. The Docker technology make use of the Linux kernel and the different and various features of Linux, because in Linux operating system the entire system is been separated from the other running applications, which makes the process or working of the Docker faster and accurate Kubernetes is the most popular container orchestration system and it was designed specifically with Google Cloud Platform integration in mind. The major advantage of using Kubernetes id orchestrating between many applications. It is also very useful for scaling many applications at very less time. By using this we can save our cost and time. So, it gives us management capabilities, saving cost, proper used of all resources.

[2] Jay Shah, Dushyant Dubaria, Cloud computing makes extensive use of virtual machines because they permit workloads to be isolated from one another and for the resource usage to be somewhat controlled. In this paper, we explore the performance of traditional virtual machine (VM) deployments, and contrast them with the use of Linux containers. We use KVM as a representative hypervisor and Docker as a container manager. Our results show that containers require tuning to support I/O intensive applications. We also discuss the implications of our performance results for future cloud architectures.

[3] Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio, Cloud computing makes extensive use of virtual machines because they permit workloads to be isolated from one another and for the resource usage to be somewhat controlled. In this paper, we explore the performance of traditional virtual machine (VM) deployments, and contrast them with the use of Linux containers. We use KVM as a representative hypervisor and Docker as a container manager. Our results show that containers result in equal or better performance than VMs in almost all cases. Both VMs and containers require tuning to support I/O intensive applications. We also discuss the implications of our performance results for future cloud architectures.

[4] ThijsScheepers, Modern cloud infrastructure uses virtualization to isolate applications, optimize the utilization of hardware resources and provide operational flexibility. However, conventional virtualization comes at the cost of resource overhead. Container-based virtualization could be an alternative as it potentially reduces overhead and thus improves the utilization of datacenters. This paper presents the results of a macro benchmark performance comparison between the two implementations of these technologies, namely Xen and LXC, as well as a discussion on their operational flexibility.

[5] Vivek Sharma, Harsh Kumar Saxena, Akhilesh Kumar Singh, Deploying web application on docker containers are quite easy, and provides interoperability and portability as well. It is very cost effective and time saving solution in the IT world. Using docker networking, docker swarm and docker compose; it is very simple to deploy a multi-tier web application on docker. There are less number of resources consumed by docker as compared to the virtual machines, so they have better speed and performance. Deploying a multi container web application on docker is very effective and easy instead of using any other cloud computing platform or infrastructure. In case of docker, entire code of the web application can be put into a single image and pushed to the docker hub for easily download that can be configured very easily, A container gives a more prominent effectiveness, both in dealing with the managing application creating application instances, and in asset utilization. Docker provides the capabilities to automate the containerized application. It provides simple and effective workflow and an additional layer has been added in the host operating system.

III. PROBLEM DEFINITION

Now a days we are facing issues in computer environment where all the software use different framework. By using Docker, it will help you to run applications with their suitable framework. Similarly, in computers we are dividing suitable framework for suitable application. Since every application has a framework, with a suitable version, this space also can be utilized for a new software application, along with their required frameworks. By deploying any web application in software we can easily access without any delay issues in servers. In web application we design a student portal application where all students upload their certificates which are done from first year by semester wise. Which is uploaded in portal to check what are the certificates are done up to now on webinar, seminar, internships and any events attended till now. Finally, the outcome of the present work was to deploy a web application in docker container with their suitable frameworks.

IV. EXISTING SYSTEM

Virtual Machine software can run programs on operating systems, store data, connect to networks, and do other computing functions, and requires maintenances such as updates and system monitoring.

Drawbacks of Existing System:

1. Virtual Machines takes a long boot-up time.
2. It also requires entire OS to be loaded before starting the surface, so less efficient.
3. Deployment is comparatively lengthy as separate instances are responsible for execution.
4. VMs make use of the hypervisor.

V. PROPOSED SYSTEM

Proposed system overcome existing system problems, our system replaces Hypervisor by using docker containers. Docker is a tool which is used to automate the deployment of application in lightweight containers so that applications can work efficiently in different environments.

Advantages of Proposed System:

- [1] Docker takes a short boot-up time.
- [2] Containers have a better performance as they are hosted in a single docker.
- [3] Docker containers use single host operating system.
- [4] Docker containers occupy less space.

VI. METHODOLOGY

System Architecture using Python

We developed the student application using Python in Django using the MVT design pattern.

Model: The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).

View: The View is the user interface that you see in your browser when you render a website. It is represented by HTML/CSS/ Javascript and Jinja files.

Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. To check more visit – Django templates

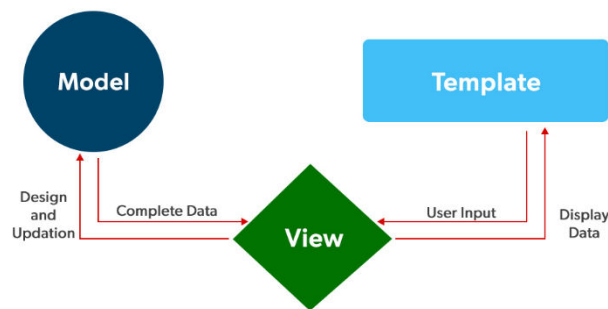


Fig 4-MVT Architecture

Implementation in Python

The Student Portal system was implemented in Python.

Python is one of the most readable and beginner-friendly programming languages for server-side web development. It's used in many frameworks and WordPress.

Visual Studio is a powerful IDE, but its focus didn't align with some languages, Python being one of them. As time progressed more languages found their way to Visual Studio, like PHP and eventually Python.

The following tools were used for the implementation in Python:

1. VisualStudioCode1.75 was used for the web application development.
2. SQLiteServer8.0.32 was used for the database development.
3. Docker was used for the web server for deployment.

Deploying web application using Docker;

Here's the basic process of developing and deploying a web application with Docker:

1. Install Docker on the machines you want to use.
2. Set up a registry at Docker Hub.
3. Initiate Docker build to create your Docker Image.
4. Set up your 'Dockerized' machines.
5. Deploy your built image or application.

Usecase Diagrams

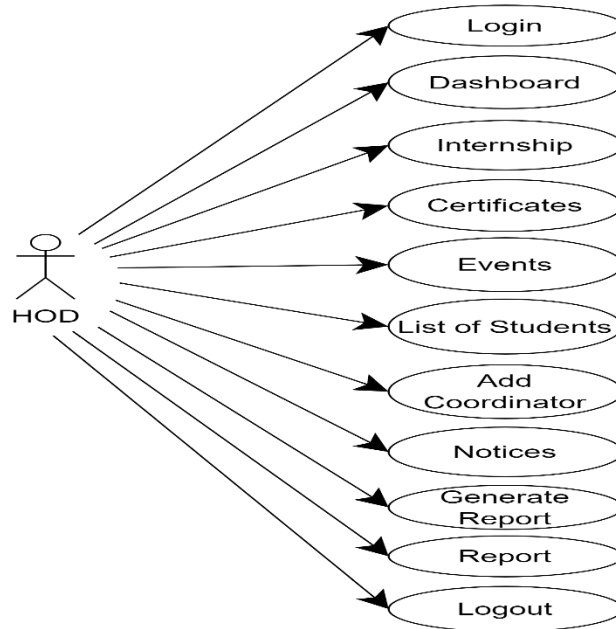


Fig 5-HOD Use Case Diagram

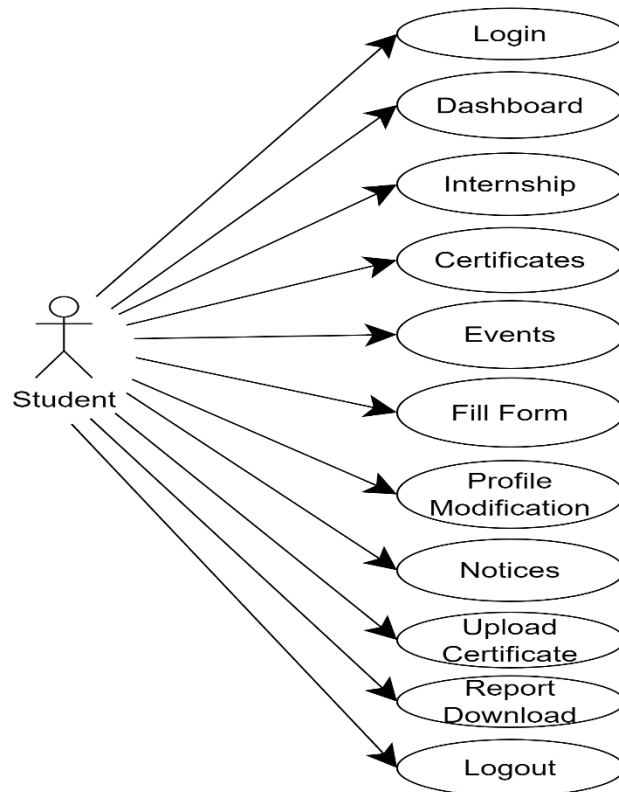


Fig6-Student Usecase Diagram

Modules

In this project, after a careful analysis the system has been identified to be presented with the following module.

The modules involved are:

Main Module: It consists of

- 1 Home: It contains the information about the web applications and pictures.
- 2 Login: It contains the login forms for admin and the faculty to perform the operations.
- 3 About Us: It contains the contact information of the admin and people in the team.

Faculty Module: It Consist of

- 1 View faculty Details: The faculty can monitor the work history of every faculty.
- 2 View Certificates: The faculty can see the list of faculty names who had done their certificates in the respective months.
- 3 View Issues: The faculty can see the complaints provided by the faculty on students or the any improvements required on teaching methodology.
- 4 Notices: The faculty can post the notices about internships, certificate courses and any further events happening.
- 5 View Internships: The admin can see the list of faculty names who are given virtual talks on the Emerging Technologies or Particular topics.
- 6 View Certificates: The admin can see the list of faculty names who are given live sessions on Emerging Technologies or Particular topics.
- 7 Reports: It contains the all the student yearly report.

Student Module: It consists of 1 Internships: The students can upload their certificates after completion and also any new internship registered status.

- 2 Certifications: The student can upload the certifications completed in respective subjects and enter the details of the certificates.
- 3 Events: The student have to upload any competitions they have participated in it.
- 4 Fill Form: Students have to fill form regarding completion of their work like internship, certificates, and events show their status.
- 5 Upload link: Students have to upload their certificates issued by organizations.

HOD Module: It Consist of

- 1 Add Faculty: The hod can add the faculty giving the required fields, username, and default password.
- 2 View faculty Details: The hod can monitor the work history of every faculty.
- 3 View Internships: The hod can see the list of students who completed and uploaded the certificate in respective months.
- 4 View Certificates: The hod can see the list of students who had done their certificates in the respective organization in particular months.
- 5 View Issues: The hod can see the complaints provided by the faculty on students or the any improvements required on teaching methodology.
- 6 Notices: The hod can post the notices about internships, certificate courses and any further events happening.
- 7 View Events: The hod can see the list of students who are given virtual talks on the Emerging Technologies or Particular topics or any events participated by student outside the state.
- 8 Reports: It contains the all the student's yearly report.

VI. RESULTS AND DISCUSSION

Student Portal Management System was developed using Python Django and SQLite database as the server- side and HTML, CSS, Java Script as the client-side. The database used in the Student portal was designedwith SQLite Database Management System (DBMS). Allthe information related to the student portal are stored in the system database.

Below figures shows the home page of the student portal system

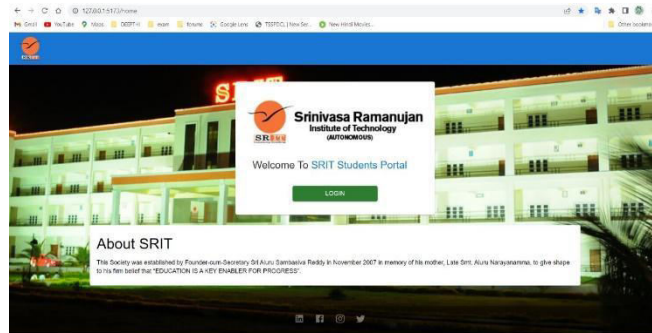


Fig 7. Home page

After clicking login you will be directed to login page where you can login through student or faculty/HOD profiles

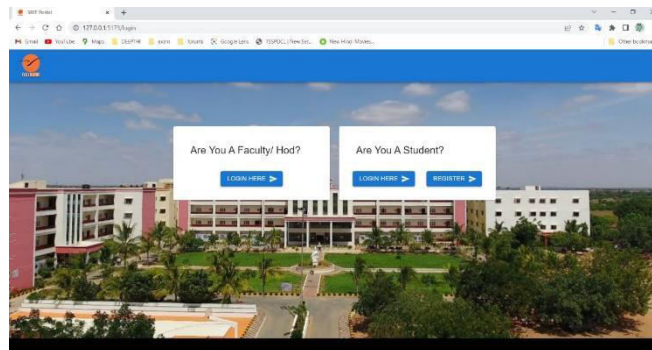


Fig 8. Login pages for both student and HOD

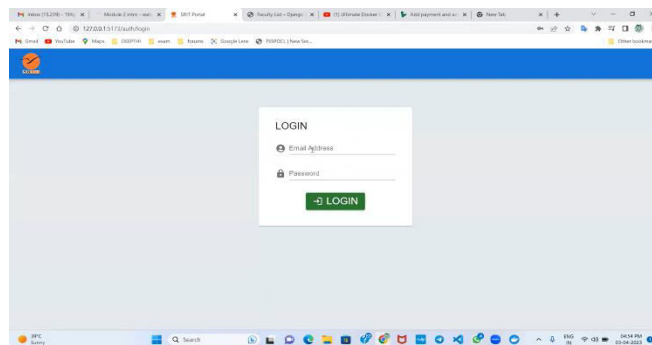


Fig 9. Login page

After entering HOD credentials there can access the respective Dashboard.

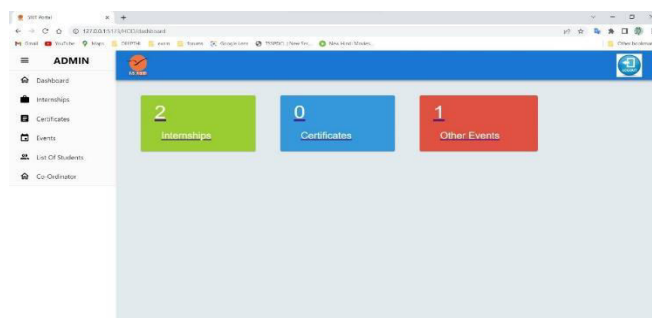


Fig 10. HOD dashboard

After entering student credentials there can access their respective Dashboard.

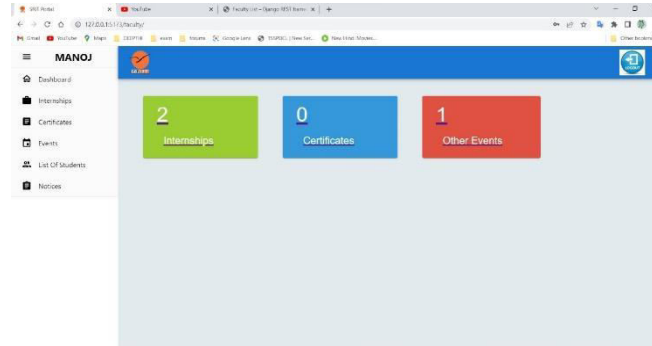


Fig 11. Student dashboard

VII. CONCLUSION

In the existing student portal system most of the work requires human attention and intervention. The major project is to build a web application for students and HOD/faculty to update and upload their certificates, prizes and awards which are earned in internships, certification courses and any other events. A user can access the website through any device. This system reduces burden to students for applying major certification courses. Here faculty updates internship programs which have to be done by student for their resumes. Creating ease in the recruitment process is major aim. After creating the student portal we are deploying the application into docker containers. Where docker is open-source software and has many advantages when compared with virtual machines environment. This attracts developers for building micro services that will run over docker containers.

In addition, a container gives a more prominent effectiveness, both in dealing with the managing application creating application instances, and in asset utilization. Basic adjustment and easy deployment of application is one of the most important principal of container services like Docker and which is completely in the DevOps soul.

REFERENCES

1. Babak Bashari Rad, Harrison john Bhatti, Mohammad Ahmadi, "An introduction to docker and analysis of its performance.pdf", IJCSNS International Journal of Computer Science and Network Security, VOL.17, March 2017.
2. JayShah, Dushyant Dubaria, "building modern clouds.pdf", 2019 IEEE.
3. Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio, "An Updated Performance Comparison of Virtual Machines and Linux Containers", 2015 IEEE.
4. Thijs Scheepers, "Virtualization and Containerization of Application Infrastructure: A Comparison". 2014
5. Vivek Sharma, Harsh Kumar Saxena, Akilesh Kumar Singh, "Docker for multi-container web application", ICIMIA 2020, IEEE.
6. Samrudhi Padwal, Samruddhi Ghorpade, Prof. P.R. Patil, Manasi Patil, Shraddha Biraje, Sapana Salunkhe, "E-TRAINING AND PLACEMENT MANAGEMENT SYSTEM", 2022.
7. Docker overview. (2021, April 23). Retrieved April 28, 2021, from <https://docs.docker.com/get-started/overview/>
8. <http://nnu.whdl.org/sites/default/files/resource/academic/Whitton.pdf>
9. Container networking. (2021, April 28). Retrieved April 28, 2021, from <https://docs.docker.com/config/containers/container-networking/>
10. brief history of virtualization. (2012). Retrieved April 28, 2021, from https://docs.oracle.com/cd/E26996_01/E18549/html/VMUSG1010.html.



INNO  **SPACE**
SJIF Scientific Journal Impact Factor
Impact Factor: 8.379



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 **9940 572 462**  **6381 907 438**  **ijircce@gmail.com**



www.ijircce.com

Scan to save the contact details