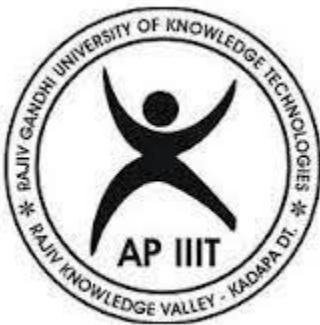


“WRITING WORLD”
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING



RGUKT
Rajiv Gandhi University of Knowledge Technologies
R.K.VALLEY

Submitted by
J Sree Manjunatha -- R161326
A Thirumalesh -- R161083

Under the Esteemed guidance of
Mr. SATHYANANDHARAM N
RGUKT RK Valley.

DECLARATION

We hereby declare that the report of the B.Tech Minor Project Work entitled "**WRITING WORLD**" which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

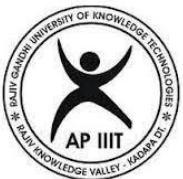
J Sree Manjunatha -- R161326

G N A N Murthy -- R161754

A Thirumalesh -- R161083

Dept. Of Computer Science and Engineering.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES



RGUKT

(A.P.Government Act 18 of 2008)

IIIT RK VALLEY, RGUKT-AP

Department of Computer Science and Engineering

CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled “**WRITING WORLD**” submitted by **J Sree Manjunatha(R161326),G N A N Murhty(R161754), A Thirumalesh(R161083)**,under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester -1 2020-2021 at IIIT ,RK VALLEY RGUKT-AP.To the best of my knowledge, the result embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide

Mr.N.Satya Nandaram
Assistant Professor
IIIT,RGUKT-AP,RK Valley

Head of the Department

Mr.T.Sandeep Kumar Reddy
HOD Of CSE
IIIT,RGUKT-AP,RK Valley

INDEX

1. Abstract
2. Introduction
 - a. Objective
 - b. StakeHolders
 - c. Context Diagram
 - d. Functional Requirements
 - e. Non-Functional Requirements
 - f. Use-Case Diagram
 - g. ER-Diagram
 - h. Sequence Diagrams
 - i. System Requirements
3. Web Frameworks
 - a. Hosting Files
 - b. HTTP Communication
4. Tools And Frameworks
5. Software Development Life Cycle
 - a. Classical Waterfall Model
 - b. Agile
6. Devops
 - a. Continuous Integration & Continuous Delivery(CI/CD)
 - b. Continuous Development
 - c. Continuous Testing
 - d. Continuous Monitoring
 - e. Continuous Deployment
7. Docker
8. Virtual Machine vs Containers
9. Cloud Computing
10. Cloud Service Providers

11. Amazon Cloud Service

- a. Amazon Elastic Cloud Compute(EC2)
- b. Amazon S3(Simple Storage Service)
- c. Amazon Relational Database Service(RDS)

12. Google Cloud

- a. Compute Engine
- b. Cloud Storage
- c. Database Services

13. Project Details

- a. Docker pages
- b. Jenkins pages
- c. Web pages
- d. Backend Pages
- e. Firebase pages
- f. Github Repositories
- g. ApplicationUrls

References

ABSTRACT

It is a Web Application useful for Writers. In this application writers can post their writings or if they want to write the story here we provide a writing pad. So they can write and save their writings and publish the writings. Here they can save their writings as drafts, come at any time and edit it. They can delete their story if they don't want to read. The writers can read all the stories published by other writers. The Readers are read the stories published by writers. They can search the stories based on genre/author name. The Readers like the story and comment under the story for writer feedback. Comments are public any Reader or writer can see the comments.

Introduction

Objective :-

Writing World Software is used by Writers and Readers .By using this software, writers,poets etc can publish their Stories and Readers read the stories and give their feedback.

Introduction :-

This document consists of requirements and specifications of Writing World Software. Writing World Software is used by Writers and Readers .By using this software, writers,poets etc can publish their Stories and Readers read the stories and give their feedback.

Purpose :-

The purpose of this document is to gather the requirements that are needed for implementing Writing World Software.It focuses on Publishing the Stories and Poems.

Intended Audience :-

The intended audience will be the Writers who want to publish their own stories and Readers having the habit of Reading the Stories.

Stakeholders :-

Client :- Writer Association

Users :- Writers , Readers

Software vision

Vision Statement :-

The software vision is to develop the Writing World Software , which is user friendly and easily accessible.This software helps the Writers and Readers

to publish the writings and Reading the published stories. The group of writers combine and place the competition among the writers.

Technologies :-

FrontEnd :- ReactJs

BackEnd :- NodeJs, ExpressJs

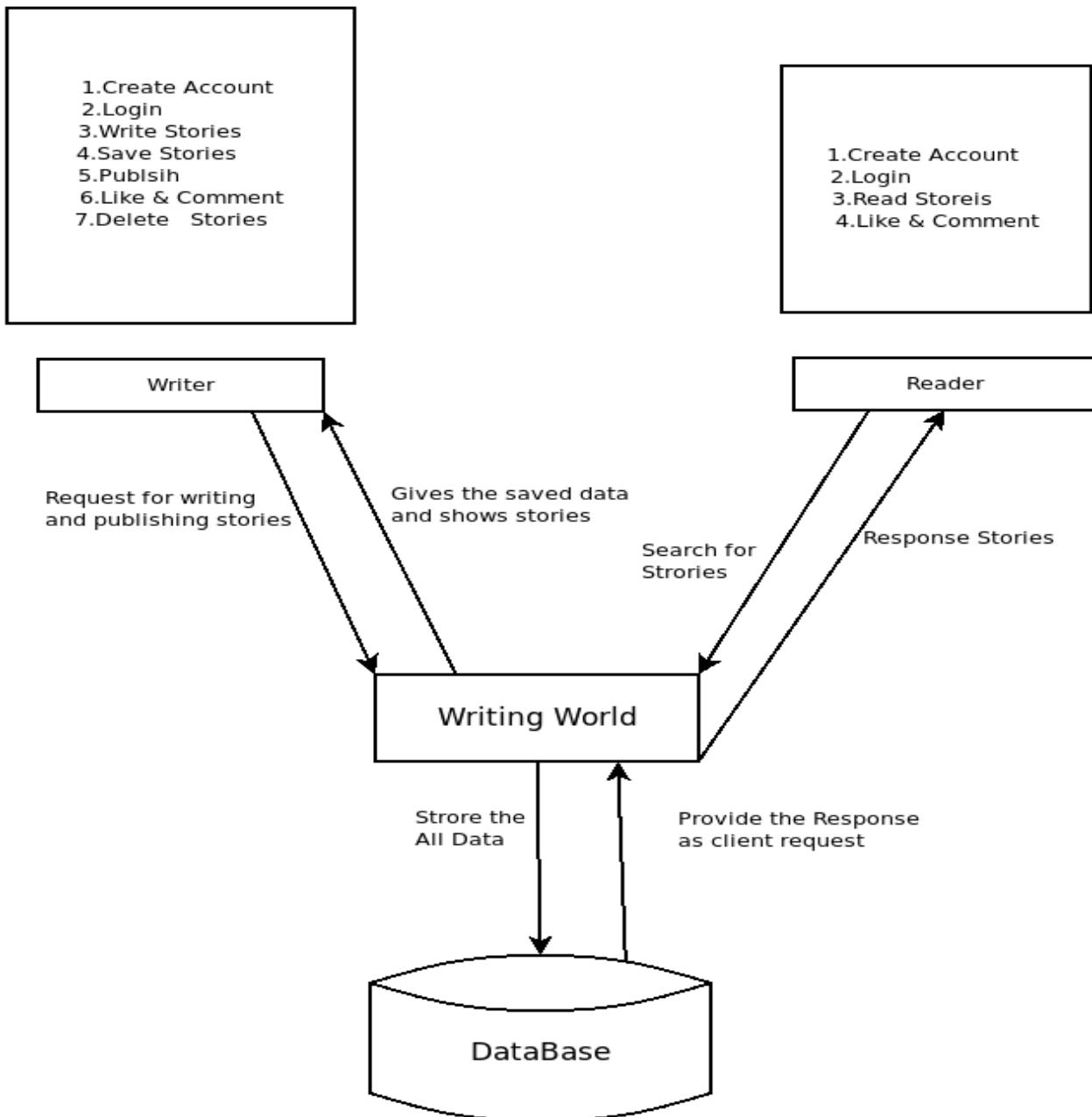
DataBase :- Firestore

Hosting :- Firebase

System in Context :-

Writing world is a software where a writer can write and publish their story and Reader can read the story based on Genre/Writer Name.

Context Diagram



Requirements :-

1. Software support two types of people
 - a. Writers :- Create account , write and publish stories
 - b. Reader :- Create an account and read the stories.
2. Writers publish the story based on following details
 - a. What type of story?
3. Privileges given by software
 - a. To Writer
 - i. Create the writer account.
 - ii. Writing the Story in the software.
 - iii. Publish the written Story or Images.
 - iv. Read other writer Stories.
 - v. Search the Stories.
 - vi. Delete the published story.
 - vii. Writing the Comments below the Story.
 - viii. Linking the Stories.
 - b. To Reader
 - i. Create the Reader account
 - ii. Read the Story based on genre/writer name
 - iii. Search the Story.
 - iv. Like the Reading Story if they like.
 - v. Comment below the Story.

Non-Functional Requirements :-

Reliability :-

Readers can read any type of stories published by users.

Usability :-

The Software has user friendly GUI.Writers can use it efficiently.

Availability :-

Writers and Readers access at any time when they want.

Accessibility :-

The software supports Writers and Readers accessing from anywhere.

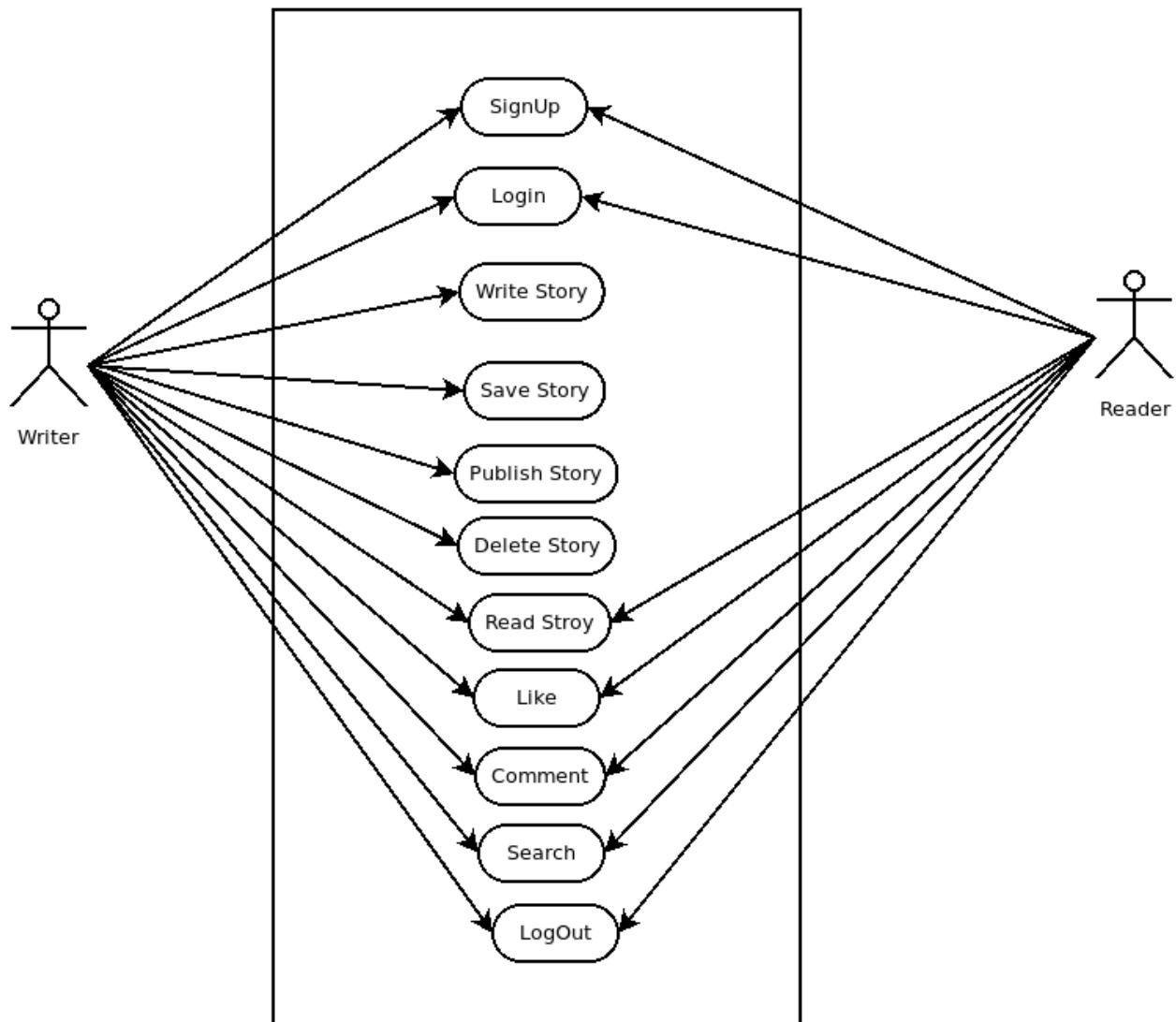
Security :-

As it is a web based application it should be more secure in order to save confidential data from hackers.

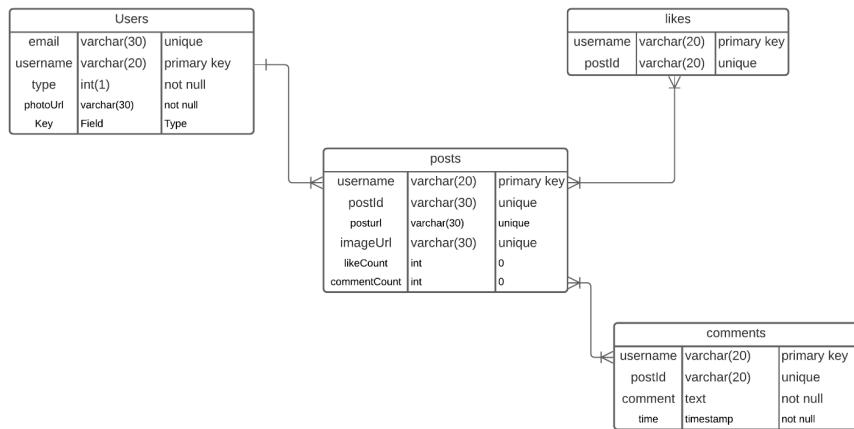
Platform Compatibility :-

This software has to work on any kind of operating System without modifying it.

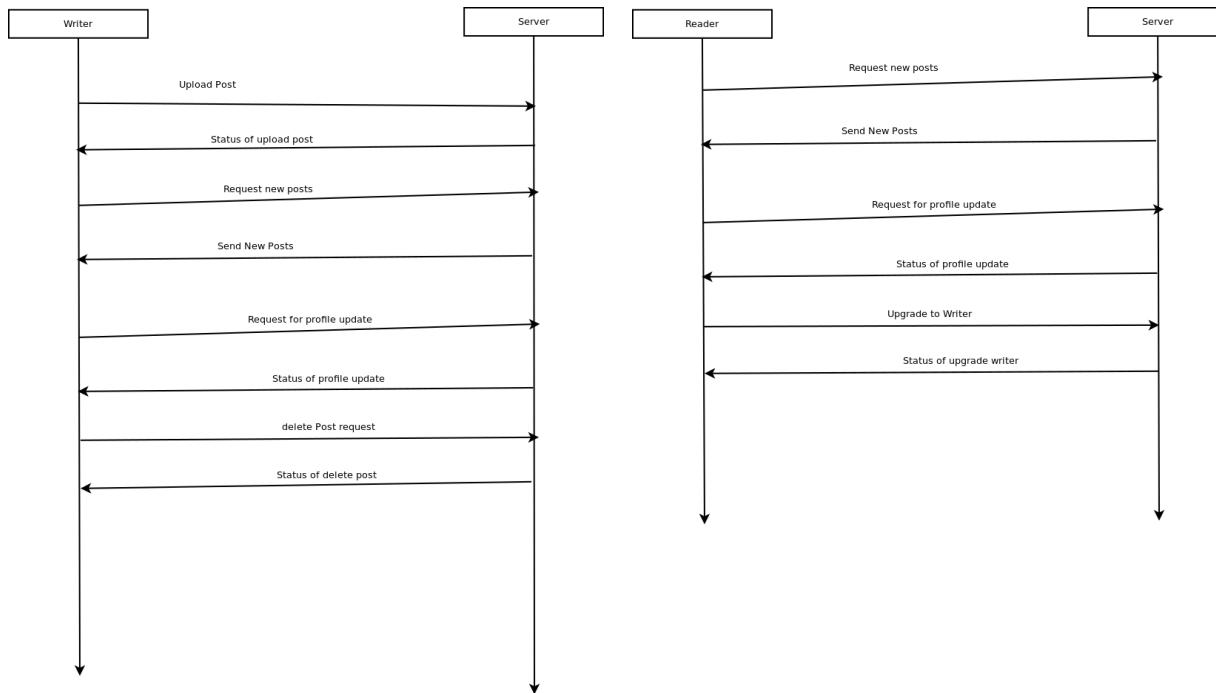
UseCase Diagram



ER Diagram



Sequence Diagrams



System Requirements

- **Hardware**

Hardware : Dual core 1.6 GHz

HDD : 100

RAM : 4GB

- **Software**

OS : ubuntu-18.04

Language : Nodejs

FrontEnd : Node Packages

IDE tools : firebase node module

WEB FRAMEWORKS

A web framework or web application framework is a software framework that is designed to support the development of web applications including web services, web resources, and web APIs. Web frameworks provide a standard way to build and deploy

Web applications on the World Wide Web. Web frameworks are playing a major role in the creation of today's most compelling web applications, because they automate many of the tedious tasks, allowing developers to instead focus on providing users with creative and powerful features.

- **Database** – Today nearly all web development frameworks are database driven. They provide support to multiple databases like MySQL, SQL ++, Oracle, and others. Every dynamic application facilitates the end-user to add, delete, and maintain records.
- **Rest APIs** - REST is an acronym for Representational State Transfer Sharing data between two or more systems has always been a fundamental requirement of software development. A REST API is a way for two computer systems to communicate over HTTP in a similar way to web browsers and servers. Similar to REST, you may encounter CORBA, SOAP, or XML-RPC e.t.c which usually establish strict messaging rules.
- **User Management** – One of the most prominent features of frameworks is user management i.e. it supports user logins with users being assigned roles and limitations placed on their access.
- **URL Mapping** – This feature allows the framework to interpret a URL with ease and simplification for the user. URL mapping simply means URL simplification.

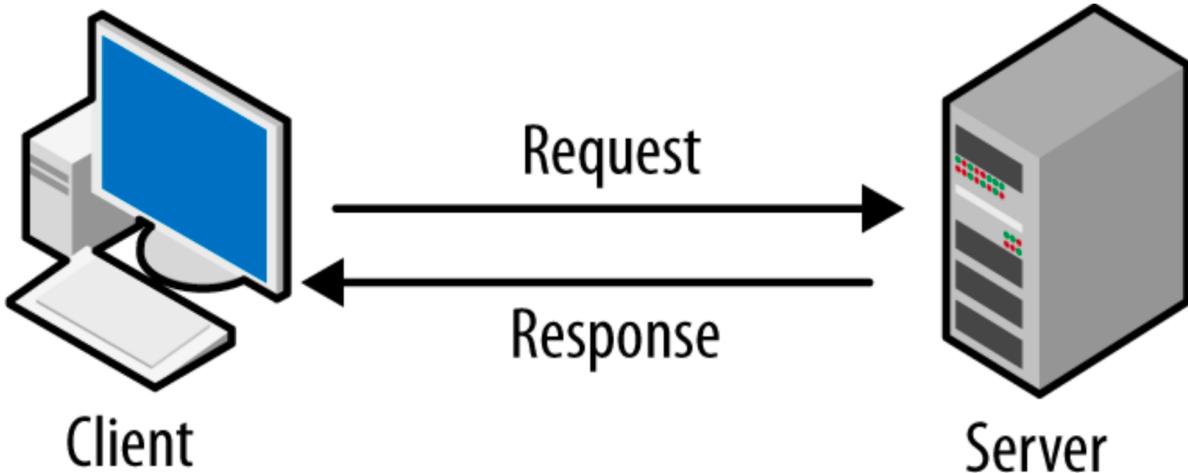
Hosting Files :-

A web server first has to store the website's files, namely all HTML documents and their related assets, including images, CSS stylesheets, JavaScript files, fonts, and videos. Technically, you could host all those files on your own computer, but it's far more convenient to store them all on a dedicated web server that

- is always up and running
- is always connected to the Internet
- has the same IP address all the time (not all ISPs provide a fixed IP address for home lines)
- is maintained by a third-party provider

HTTP communication :-

A web server provides support for HTTP (Hypertext Transfer Protocol). As its name implies, HTTP specifies how to transfer hypertext (i.e., linked web documents) between two computers.



A Protocol is a set of rules for communication between two computers. HTTP is a textual, stateless protocol. HTTP provides clear rules for how a client and server communicate. Only clients can make HTTP requests, and then only to servers. Servers can only respond to a client's HTTP request.

- When requesting a file via HTTP, clients must provide the file's URL.
- The web server must answer every HTTP request, at least with an error message.

Tools and Frameworks

FrontEnd :-

ReactJs

React (also known as React.js or ReactJS) is an open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Material Ui

Material UI is an open-source, front-end framework for React components. It is built using Less. Less (stands for Leaner Style Sheets), is a backward-compatible language extension for CSS. Material UI is based on Google's Material Design to provide a high-quality, digital experience while developing front-end graphics. Material Design focuses on providing bold and crisp designs – it builds textures by focusing on how the components cast shadows and reflect light.

BackEnd:-

NodeJs

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Database

Firebase is a NoSQL document database built for automatic scaling, high performance, and ease of application development. While the Firestore interface has many of the same features as traditional databases, as a NoSQL database it differs from them in the way it describes relationships between data objects.

Software Development Life Cycle(SDLC) :-

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

Classic Water Model :-

Waterfall is a traditional, plan-driven practice for developing systems. One of the earliest software development lifecycle (SDLC) approaches, it is a practice of developing in stages: Gather and analyze software requirements, design, develop, test and deploy into operations. The output of one stage is required to initiate the next stage.

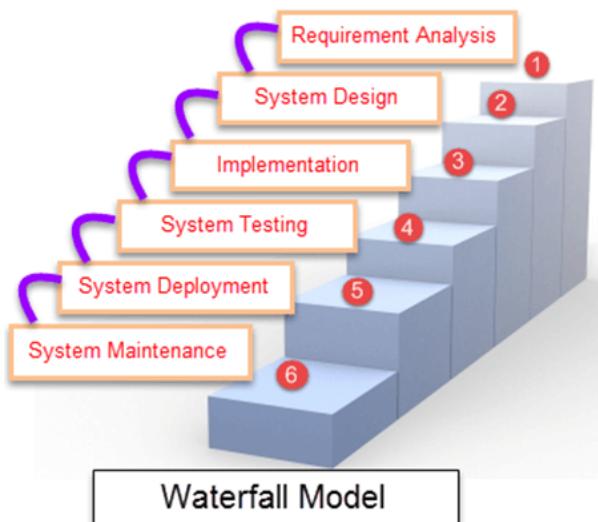
Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everything from documents till the logic.

Advantages

- Uses clear structure
- Determine the end goal correctly
- Transfers Information Well

Disadvantages

- Makes Changes difficult
- Excludes the client and/or end user
- Delays testing until after completion



Agile :-

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

Agile development process breaks the product into smaller pieces and integrates them for final testing. It can be implemented in many ways, including scrum, kanban, scrum, XP, etc.



Advantages

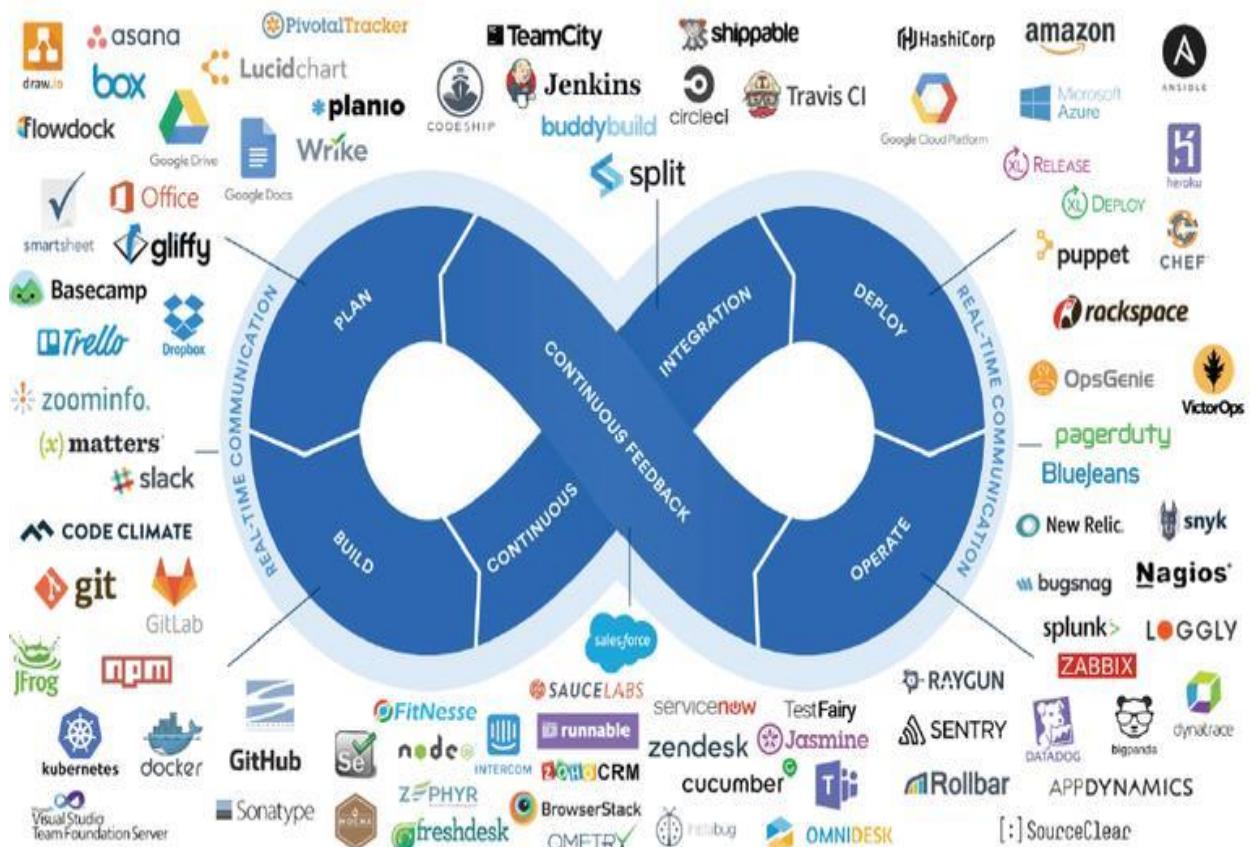
- Responding to Change
- Accepting Uncertainty
- Faster Review Cycles
- Greater Flexibility in Releasing Features
- Less Up-Front Work

Disadvantages

- Lack of Understanding
- Flexibility Can Lead to Bad Behaviors
- Culture Fit
- Lack of Predictability
- Challenges at Scale

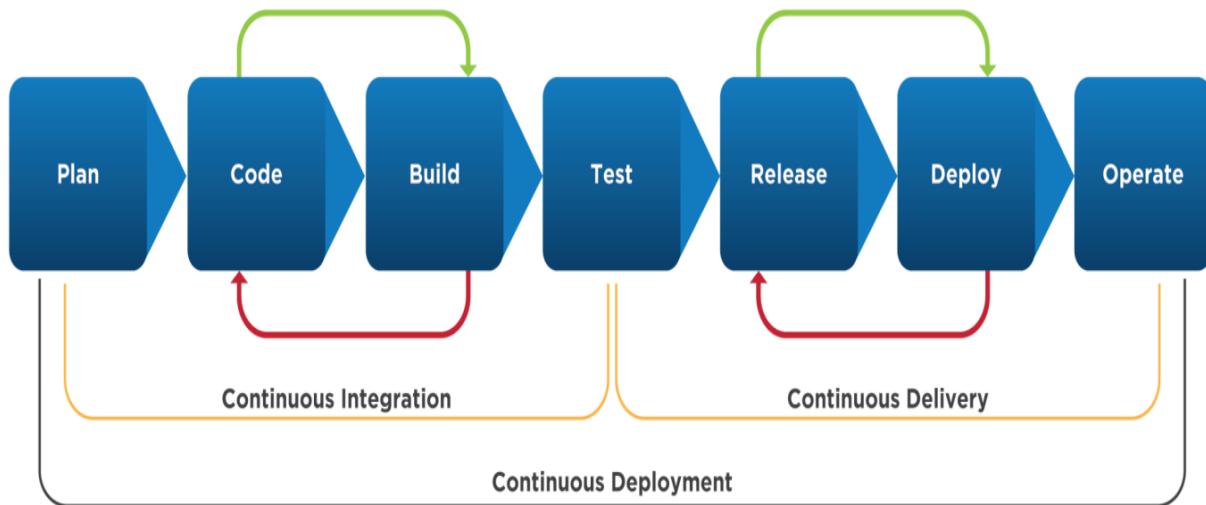
DevOps :-

DevOps is a software development strategy that bridges the gap between the developers and the IT staff. With DevOps, organizations can release small features very quickly and incorporate the feedback which they receive immediately. The DevOps process involves a lot of development, testing, and deployment of technologies for developing automated CI/ CD pipelines.



Continuous Integration & Continuous Deployment(CI/CD) :-

CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development.



| Planning | Code | Build | Test | Release | Deploy | Operate |
|--|---|--|---|--|--|--|
| <ul style="list-style-type: none">Requirement finalizationUpdates & new changesArchitecture & designTask assignmentTimeline finalization | <ul style="list-style-type: none">DevelopmentConfiguration finalizationCheck-in source codeStatic-code analysisAutomated review & peer review | <ul style="list-style-type: none">Compile codeUnit testingCode-metricsBuild container images or packagePreparation or update in deployment templatedCreate or update monitor dashboards | <ul style="list-style-type: none">Integration test with other componentLoad & stress testUI testingPenetration testingRequirement testing | <ul style="list-style-type: none">Preparing release notesVersion taggingCode freezeFeature freeze | <ul style="list-style-type: none">Updating the infrastructure i.e staging, productionVerification on deployment i.e smoke tests | <ul style="list-style-type: none">Monitor designed dashboardAlarm triggersAutomatic critical events handlerMonitor error logs |

Stages :-

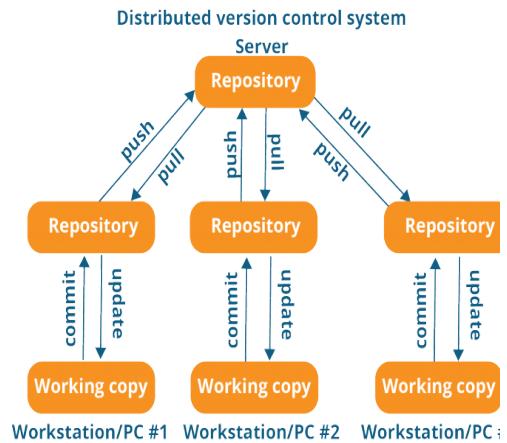
Continuous Development :-

Tools :- Git, SVN, CVS, Mercurial

Git :-

Git is a free and open source **distributed version control system** designed to handle everything from small to very large projects with speed and efficiency.

Distributed VCS, every contributor has a local copy or “clone” of the main repository i.e. everyone maintains a local repository of their own which contains all the files and metadata present in the main repository.



Basic Commands :-

| | |
|---|---|
| git init | To initialize the git repository in local environment |
| git status | To remind you of where you left off. See a summary of local changes, remote commits, and untracked files. |
| git add | To stage changes to your tracked and untracked files. Use -u , -a , and . strategically. |
| git commit | To create a new commit with changes previously added. Use -m and add a meaningful commit message. |
| git push | To send changes to your configured remote repository, most commonly GitLab or GitHub. |
| git remote origin “repository url” | Connecting our local repository to the github repository |
| git pull origin master | getting all the contents in the repository after linking to the local repository. |

Continuous Integration :-

Continuous Integration is a process of integrating code changes from multiple developers in a single project many times. The software is tested immediately after a code commit. With each code commit, code is built and tested. If the test is passed, the build is tested for deployment. If the deployment is successful, the code is pushed to production.

Continuous Delivery :-

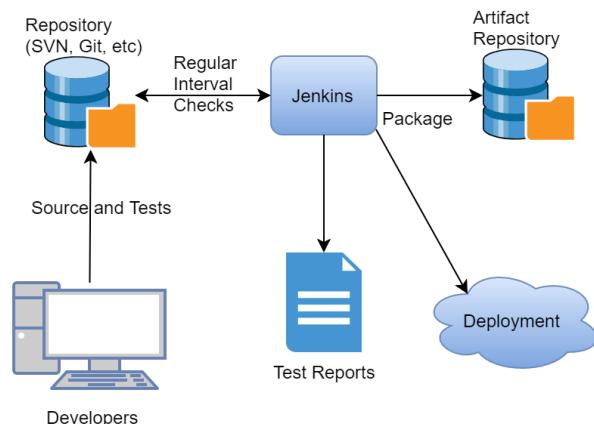
In Continuous Delivery(CD), merged code changes are automatically built and deployed to a testing environment. Then, automated tests are executed against the deployed code to identify any bugs and allow the developers to fix them in advance.

Tools :- Jenkins, Travis, Bamboo, CruiseControl, TeamCity

Jenkins :-

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Jenkins is a highly extensible product whose functionality can be extended through the installation of plugins.



Jenkins Pipeline :-

Jenkins Pipeline (or simply "Pipeline") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.

Continuous Testing :-

Continuous testing is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate.

Tools :- Junit, Selenium

Selenium :-

Selenium is an open-source tool that is used for automating the tests carried out on web browsers (Web applications are tested using any

web browser).

Junit :-

It is an open-source testing framework for java programmers. The java programmer can create test cases and test his/her own code. It is one of the unit testing frameworks.

To perform unit testing, we need to create test cases. The unit test case is a code which ensures that the program logic works as expected.

Continuous Monitoring :-

The application, infrastructure, middleware, and networks will be monitored continuously for their performance, any defects, or security and compliance. To identify issues, most companies monitor metrics such as:

- CPU and Memory Usage
- Disk Space
- Custom Policies

By Putting continuous monitoring into practice, you always be alerted about any issues in environments from testing to production, helping you ensure high availability.

Tools :- Nagios, Sensu, Prometheus.

Continuous Deployment :-

The final stage of a mature CI/CD pipeline is continuous deployment. As an extension of continuous delivery, which automates the release of a production-ready build to a code repository, continuous deployment automates releasing an app to production. Because there is no manual gate at the stage of the pipeline before production, continuous deployment relies heavily on well-designed test automation.

Tools :- Puppet, Ansible, Docker, Chef, Kubernetes, AWSCodeDeploy

Docker :-

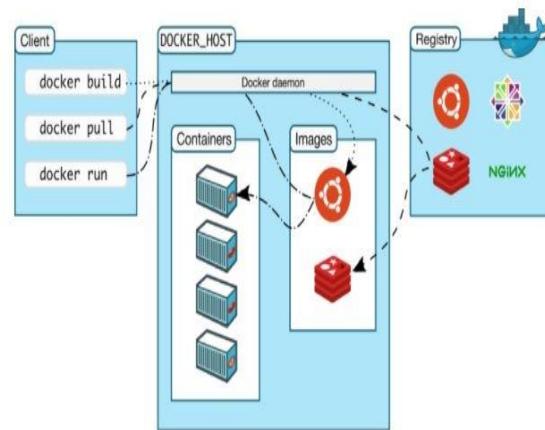
Docker is an open source project that offers a software development solution known as containers.

Container :-

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

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.

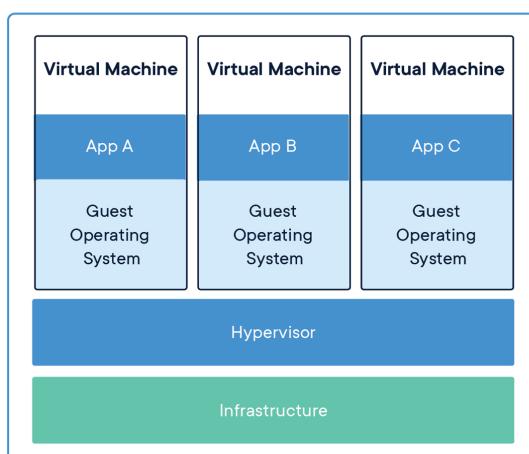
Docker images become containers when they run on Docker Engine



Architecture

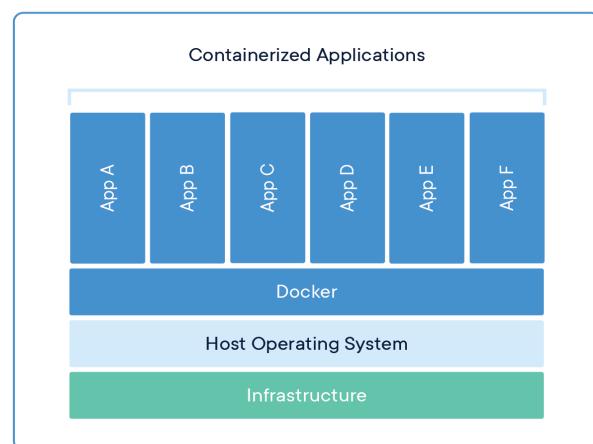
Virtual Machine

Virtualization machines (VMs) are an abstraction of physical hardware turning one server into many servers into many servers. The hypervisor 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 taking up tens of GBs.



Containers

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with the other containers, each running as an isolated process in user space.



Cloud Computing :-

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider.

Benefits using Cloud Computing :-

- ❖ **Agility**

The cloud gives you easy access to a broad range of technologies so that you can innovate faster and build nearly anything that you can imagine.

- ❖ **Elasticity**

With cloud computing, you don't have to over-provision resources up front to handle peak levels of business activity in the future. Instead, you provision the amount of resources that you actually need. You can scale these resources up or down to instantly grow and shrink capacity as your business needs change.

- ❖ **Cost Savings**

The cloud allows you to trade capital expenses (such as data centers and physical servers) for variable expenses, and only pay for IT as you consume it

Types of Cloud Computing :-

> Infrastructure as a Code (IAAS) :-

IaaS contains the basic building blocks for cloud IT. It typically provides access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS gives you the highest level of flexibility and management control over your IT resources.

> Platform as a Service (PAAS) :-

PaaS removes the need for you to manage underlying infrastructure (usually hardware and operating systems), and allows you to focus on the deployment and management of your applications.

> Software as a Service (SAAS) :-

SaaS provides you with a complete product that is run and managed by the service provider. In most cases, people referring to SaaS are referring to end-user applications (such as web-based email).

Cloud Service Providers

- **Amazon Web Services(AWS)**
- **Google Cloud**
- **Microsoft Azure**
- **IBM Cloud**
- **Alibaba Cloud**



Amazon Web Services

Services :-

Amazon Elastic Cloud Compute (EC2)

Amazon S3(Simple Storage Service)

Amazon Relational Database Service(RDS)

Amazon Elastic Cloud Compute(EC2) :-

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

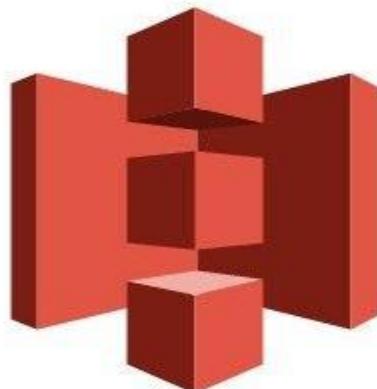
Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model.



Amazon EC2

Amazon S3(Simple Storage Service) :-

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.



Amazon S3

Amazon Relational Database Service (RDS) :-

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server.



Amazon RDS



Google Cloud

Google Cloud

Compute Engine :-

Compute Engine is a computing and hosting service that lets you create and run virtual machines on Google infrastructure. Compute Engine offers scale, performance, and value that lets you easily launch large compute clusters on Google's infrastructure.



Cloud Storage :-

Cloud Storage is a service for storing your objects in Google Cloud. An object is an immutable piece of data consisting of a file of any format. You store objects in containers called buckets. All buckets are associated with a project, and you can group your projects under an organization.

Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download.

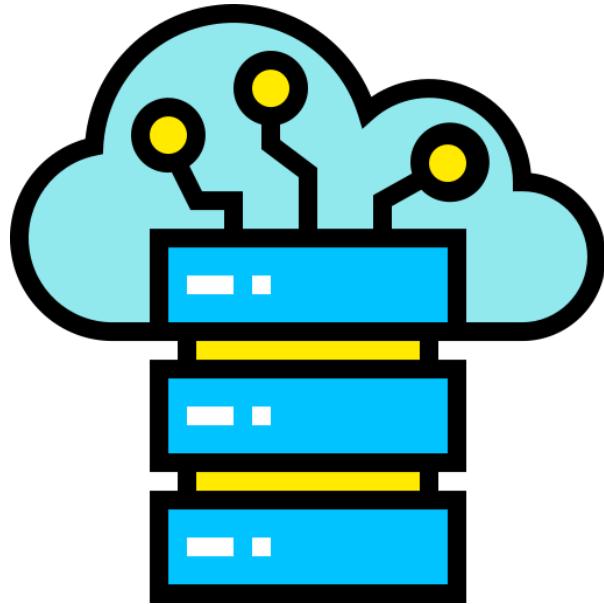


Google Cloud Storage

Database services :-

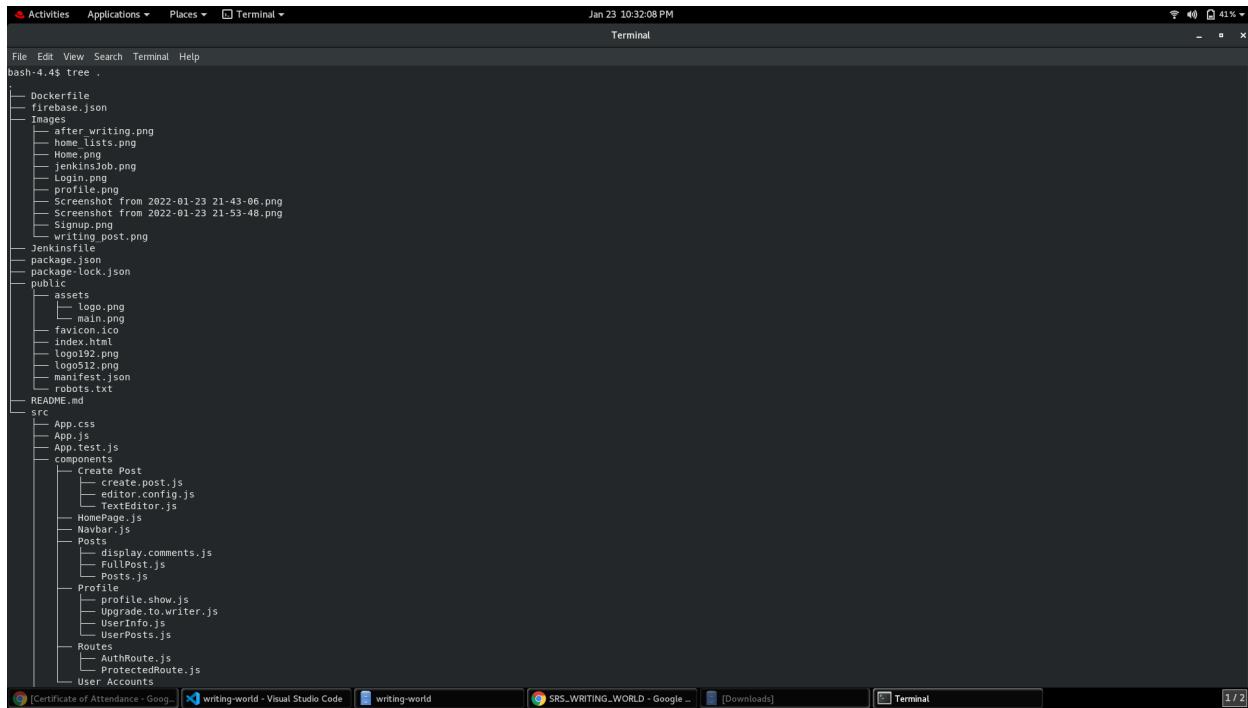
Cloud Provides variety of Database Services

- A SQL database in Cloud SQL, which provides either MySQL or PostgreSQL databases.
- A fully managed, mission-critical, relational database service in Cloud Spanner that offers transactional consistency at global scale, schemas, SQL querying, and automatic, synchronous replication for high availability.
- Two options for NoSQL data storage: Firestore, for document-like data, and Cloud Bigtable, for tabular data.

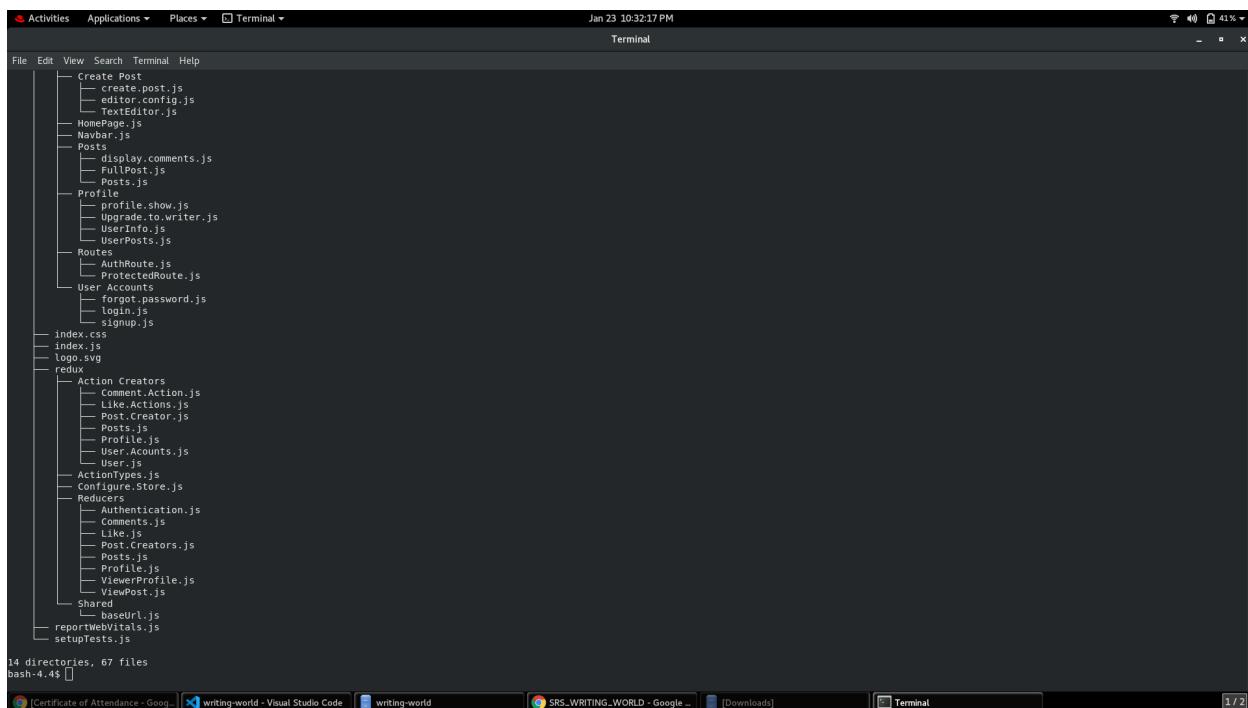


Project Details

Files Workflow



```
File Edit View Search Terminal Help
bash-4.4$ tree .
.
├── Dockerfile
├── firebase.json
└── Images
    ├── after_writing.png
    ├── home_lists.png
    ├── Home.png
    ├── logo_128px.png
    ├── Login.png
    ├── profile.png
    └── Screenshot from 2022-01-23 21-43-06.png
        └── Screenshot from 2022-01-23 21-53-48.png
            └── Signup.png
            └── write post.png
└── Jenkinsfile
├── package.json
└── package-lock.json
└── public
    ├── assets
    │   ├── logo.png
    │   ├── main.png
    │   ├── favicon.ico
    │   ├── index.html
    │   ├── logo192.png
    │   ├── logo512.png
    │   └── manifest.json
    └── robots.txt
└── README.md
src
├── App.css
├── App.js
└── App.test.js
└── components
    ├── Create Post
    │   ├── create.post.js
    │   ├── editor.config.js
    │   └── TextEditor.js
    ├── HomePage.js
    └── Navbar.js
└── Posts
    ├── display.comments.js
    ├── FullPost.js
    └── Posts.js
└── Profile
    ├── profile.show.js
    ├── Upgrade.to.writer.js
    └── UserInfo.js
    └── UserPosts.js
└── Routes
    ├── AuthRoute.js
    └── ProtectedRoute.js
└── User Accounts
    ├── forgot.password.js
    └── login.js
    └── signup.js
└── index.css
└── index.js
└── logo.svg
└── redux
    ├── Action Creators
    │   ├── Comment.Action.js
    │   ├── Like.Actions.js
    │   ├── Post.creator.js
    │   └── Posts.js
    ├── Profile.js
    └── User Accounts.js
    └── User.js
    ├── ActionTypes.js
    └── Configure.Store.js
    └── Reducers
        ├── Authentication.js
        ├── Comment.js
        ├── Like.js
        ├── Post.Creators.js
        ├── Posts.js
        ├── Profile.js
        └── ViewerProfile.js
        └── ViewPost.js
    └── Shared
        └── baseUrl.js
        └── reportWebVitals.js
        └── setupTests.js
14 directories, 67 files
bash-4.4$
```



```
File Edit View Search Terminal Help
bash-4.4$ tree .
.
├── Create Post
│   ├── create.post.js
│   ├── editor.config.js
│   └── TextEditor.js
├── HomePage.js
└── Navbar.js
└── Posts
    ├── display.comments.js
    ├── FullPost.js
    └── Posts.js
└── Profile
    ├── profile.show.js
    ├── Upgrade.to.writer.js
    └── UserInfo.js
    └── UserPosts.js
└── Routes
    ├── AuthRoute.js
    └── ProtectedRoute.js
└── User Accounts
    ├── forgot.password.js
    └── login.js
    └── signup.js
└── index.css
└── index.js
└── logo.svg
└── redux
    ├── Action Creators
    │   ├── Comment.Action.js
    │   ├── Like.Actions.js
    │   ├── Post.creator.js
    │   └── Posts.js
    ├── Profile.js
    └── User Accounts.js
    └── User.js
    ├── ActionTypes.js
    └── Configure.Store.js
    └── Reducers
        ├── Authentication.js
        ├── Comment.js
        ├── Like.js
        ├── Post.Creators.js
        ├── Posts.js
        ├── Profile.js
        └── ViewerProfile.js
        └── ViewPost.js
    └── Shared
        └── baseUrl.js
        └── reportWebVitals.js
        └── setupTests.js
14 directories, 67 files
bash-4.4$
```

Dockerfile

```
FROM node:14-alpine

RUN mkdir /app

WORKDIR /app

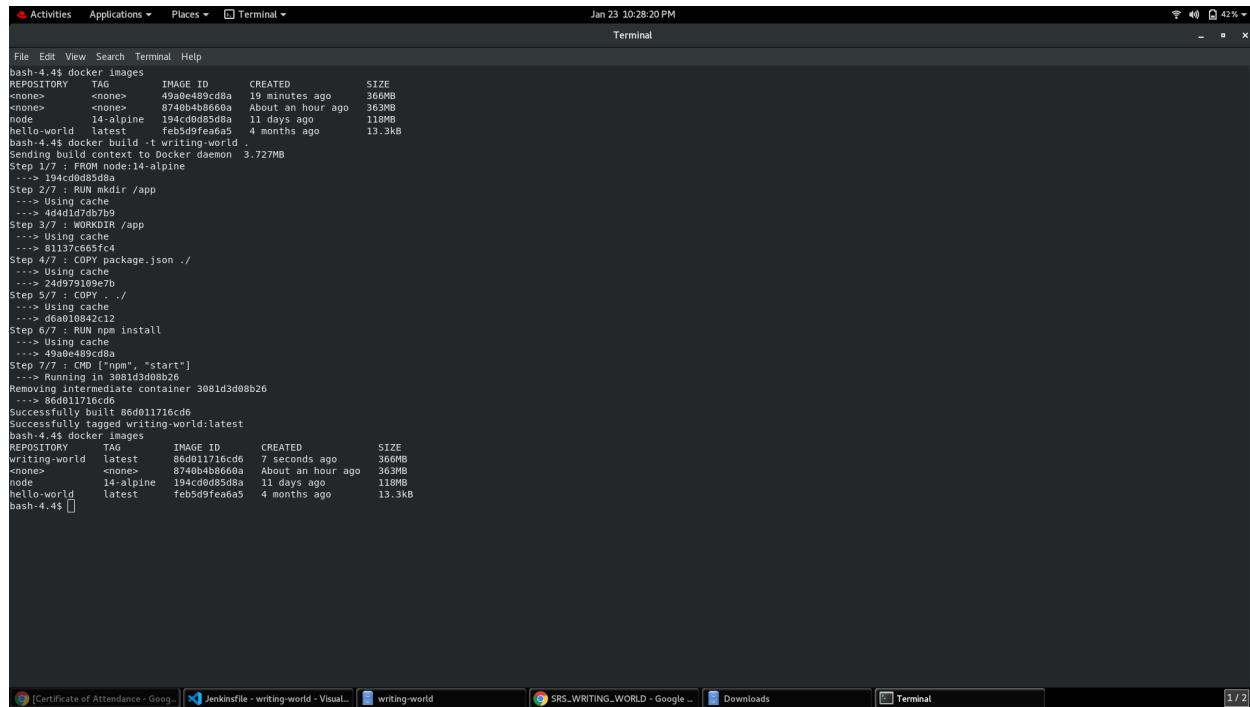
COPY package.json ./

COPY . ./

RUN npm install

CMD ["npm", "start"]
```

Docker Codes



The screenshot shows a terminal window titled 'Terminal' running on a Linux desktop. The window displays the command-line history of a Docker build process. The build steps are numbered from 1/7 to 7/7, showing the creation of an image named 'writing-world'. The terminal also shows the removal of a temporary container and the successful tagging of the image.

```
Activities Applications ▾ Places ▾ Terminal ▾
File Edit View Search Terminal Help
Jan 23 10:28:20 PM
bash-4.4$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
<none> <none> 49a0e4dcd6a 19 minutes ago 360MB
<none> none 8740dbab669a About an hour ago 363MB
node 14-alpine 194cd0d8d5d8a 11 days ago 11.0MB
hello-world latest febd59fea6a5 4 months ago 13.3kB
bash-4.4$ docker build -t writing-world .
Step 1/7 : FROM node:14-alpine
--> 4d4d1d7db7b9
Step 2/7 : RUN mkdir /app
--> Using cache
Step 3/7 : WORKDIR /app
--> Using cache
Step 4/7 : COPY package.json .
--> Using cache
--> 24d979109e7b
Step 5/7 : COPY . .
--> Using cache
--> 49a0e4dcd6a2
Step 6/7 : RUN npm install
--> Using cache
--> 49a0e4d89cd8a
Step 7/7 : CMD ["npm", "start"]
--> Running in 3081d3d00b26
Removing intermediate container 3081d3d00b26
--> 86d011716cd6
Successfully built 86d011716cd6
Successfully tagged writing-world:latest
bash-4.4$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
writing-world latest 86d011716cd6 7 seconds ago 360MB
<none> <none> 8740dbab669a About an hour ago 363MB
node 14-alpine 194cd0d8d5d8a 11 days ago 11.0MB
hello-world latest febd59fea6a5 4 months ago 13.3kB
bash-4.4$
```

A screenshot of a Linux desktop environment. At the top, there is a horizontal menu bar with "Activities", "Applications", "Places", and "Terminal". The date and time "Jan 23 10:29:33 PM" are displayed in the top right corner. Below the menu bar is a terminal window titled "Terminal" with the command "bash-4.4\$" at the prompt. Inside the terminal, the output of the "docker ps" command is shown:

```
bash-4.4$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
b9f159409277eab9ab9e6b132ce8a9ae83ca49d23c56115ecfa738800d12d
bash-4.4$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
b9f159409277eab9ab9e6b132ce8a9ae83ca49d23c56115ecfa738800d12d
bash-4.4$
```

The terminal window has a dark background and light-colored text. Below the terminal is a docked application bar with several icons: "[Certificate of Attendance - Google]", "[Jenkinsfile - writing-world - Visual...]", "[writing-world]", "[SRS_WRITING_WORLD - Google ...]", "[Downloads]", and "[Terminal]". On the far right of the dock, there is a small "1 / 2" indicator.

Jenkinsfile

```
//https://writingworld-9balf.web.app/
pipeline {
    agent any
    stages{
        stage('Build an Image') {
            steps{
                sh 'docker build -t writing-world .'
            }
        }
        stage('Run container') {
            steps{
                sh 'docker run -d --name writing-world writing-world'
            }
        }
        // stage('Clean docker image'){
        //     steps{
        //         sh 'docker stop writing-world && docker rm writing-world && docker rmi writing-world'
        //     }
        // }
    }
}
```

Activities Applications Places Google Chrome

writing-world [Jenkins] 172.17.0.2 127.0.0.1:8080/job/writing-world/ Apps Tasks

Jan 23 9:41:15 PM

Jenkins

Dashboard > writing-world >

Back to Dashboard Status Changes Build Now Configure Delete Pipeline Full Stage View Rename Pipeline Syntax

Build History trend ▾

Filter builds... #10 Jan 23, 2022 9:39 PM #9 Jan 23, 2022 9:35 PM #8 Jan 23, 2022 9:31 PM #7 Jan 23, 2022 9:30 PM #6 Jan 23, 2022 9:30 PM #5 Jan 23, 2022 9:29 PM #4 Jan 23, 2022 9:21 PM #3 Jan 23, 2022 9:11 PM #2 Jan 23, 2022 9:10 PM

Pipeline writing-world

Recent Changes

Add description Disable Project

Stage View

Average stage times: (Average full run time: ~1min 15s)

| | Declarative: Checkout SCM | Build an Image | Run container | Clean docker image |
|-----------------------------|---------------------------|----------------|---------------|--------------------|
| #10 Jan 23 21:59 No changes | 1s | 25s | 272ms | 470ms |
| #9 Jan 23 21:55 1 commit | 700ms | 1min 4s | 543ms | |
| #8 Jan 23 21:31 No changes | 1s | 57s | 535ms | |
| #7 Jan 23 21:31 No changes | 825ms | 53s | 565ms | 2s |
| #6 Jan 23 21:30 No changes | | | | |
| #5 Jan 23 21:30 No changes | 915ms | 324ms | 23ms | 24ms |

1 / 2

The screenshot shows the Jenkins Pipeline interface for the 'writing-world' project. On the left, there's a sidebar with various management links like Status, Changes, and Build History. The main area is titled 'Pipeline writing-world' and displays a 'Stage View' chart. The chart tracks the execution time for four stages across six builds. The stages are: Declarative: Checkout SCM, Build an Image, Run container, and Clean docker image. The chart shows significant variability in build times, with some builds taking over a minute due to the 'Build an Image' stage.

Activities Applications Places Google Chrome Jan 23 10:52:08 PM

writing-world #10 Console | Writing World | WhatsApp | GitHub | +

127.0.0.1:8080/job/writing-world/10/console

Apps Tasks G Reading list

Jenkins AARAVETI THIRUMALESH log out

Dashboard > writing-world > #10

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#) View as plain text

[Edit Build Information](#)

[Delete build #10'](#)

[Git Build Data](#)

[Restart from Stage](#)

[Replay](#)

[Pipeline Steps](#)

[Workspaces](#)

[Previous Build](#)

Console Output

Started by user AARAVETI THIRUMALESH
 Obtained Jenkinsfile from git <https://github.com/thirumaleshaaraveti/writing-world.git>
 [Pipeline] Start of Pipeline
 [Pipeline] node
 Running on Jenkins in /var/lib/jenkins/workspace/writing-world
 [Pipeline] stage
 [Pipeline] { (Declarative: Checkout SCM)
 [Pipeline] checkout
 Selected Git installation does not exist. Using Default
 The recommended git tool is: NONE
 No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/writing-world/.git # timeout=10
 Fetching changes from the remote Git repository
 > git config remote.origin.url <https://github.com/thirumaleshaaraveti/writing-world.git> # timeout=10
 Fetching upstream changes from <https://github.com/thirumaleshaaraveti/writing-world.git>
 > git fetch --tags --progress
 > git -version # 'git version 2.27.0'
 > git fetch --tags --progress - <https://github.com/thirumaleshaaraveti/writing-world.git> +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 Checking out Revision 4a2491f98c71eb1bb79881eb75c8acabd56344ee (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git rev-list --no-walk 4a2491f98c71eb1bb79881eb75c8acabd56344ee # timeout=10
 [Pipeline] Commit message: "Remove image clean in Jenkinsfile"
 > git config rev-list --no-walk 4a2491f98c71eb1bb79881eb75c8acabd56344ee # timeout=10
 [Pipeline] // stage
 [Pipeline] withEnv
 [Pipeline] {
 [Pipeline] // stage
 [Pipeline] { (Build an Image)
 [Pipeline] sh
 + docker build -t writing-world .
 Sending build context to Docker daemon 1.01MB
 Step 1/7 : FROM node:14-alpine

1/7 : FROM node:14-alpine

writing-world #10 Console [Jenk... README.md - writing-world - Visu... Images SRS_WRITING_WORLD - Google ... [Downloads] Terminal GitHub: Where the world builds s... 1 / 2

Activities Applications Places Google Chrome Jan 23 10:52:10 PM

writing-world #10 Console | Writing World | WhatsApp | GitHub | +

127.0.0.1:8080/job/writing-world/10/console

Apps Tasks G Reading list

Dashboard > writing-world > #10

```

  + docker build -t writing-world .
  Sending build context to Docker daemon 1.01MB

Step 1/7 : FROM node:14-alpine
--> 194c0d8b58a
Step 2/7 : RUN mkdir /app
--> Using cache
--> 44d41d7dbb9
Step 3/7 : WORKDIR /app
--> Using cache
--> 81137c665f4
Step 4/7 : COPY package.json .
--> Using cache
--> 24d979109e7b
Step 5/7 : COPY /
--> 4bfc7869497e
Step 6/7 : RUN npm install
--> Running in 68fb07cd998c

> core-js@2.6.12 postinstall /app/node_modules/babel-runtime/node_modules/core-js
> node -e "try{require('../postinstall')}catch(e){}"
[196mThank you for using core-js ([194m https://github.com/zloirock/core-js ([196m) for polyfilling JavaScript standard library![194m

[196mThe project needs your help! Please consider supporting of core-js on Open Collective or Patreon: [194m
[196m[194m https://opencollective.com/core-js [196m
[196m[194m https://www.patreon.com/zloirock [196m

[196mAlso, the author of core-js ([194m https://github.com/zloirock ([196m) is looking for a good job -)[196m

> core-js@3.15.2 postinstall /app/node_modules/core-js
> node -e "try{require('../postinstall')}catch(e){}"
[196mThank you for using core-js ([194m https://github.com/zloirock/core-js ([196m) for polyfilling JavaScript standard library![194m

[196mThe project needs your help! Please consider supporting of core-js on Open Collective or Patreon: [194m
[196m[194m https://opencollective.com/core-js [196m
[196m[194m https://www.patreon.com/zloirock [196m

[196mAlso, the author of core-js ([194m https://github.com/zloirock ([196m) is looking for a good job -)[196m

```

writing-world #10 Console [Jenk... README.md - writing-world - Visu... Images SRS_WRITING_WORLD - Google ... [Downloads] Terminal GitHub: Where the world builds s... 1 / 2

Activities Applications Places Google Chrome Jan 23 10:52:15 PM

writing-world #10 Console | Writing World | WhatsApp | GitHub | +

Dashboard > writing-world > #10

```
Thank you for installing [[35mEJS][0m: built with the [[32mJake][0m JavaScript build tool ([[32mhttps://jakejs.com/][0m)

[[01m]WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@0.1.2.13 (node_modules/webpack-dev-server/node_modules/fsevents):
[[0m][[1m]WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@0.1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
[[0m][[1m]WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@01.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@01.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
[[0m][[1m]WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@0.3.2 (node_modules/fsevents):
[[0m][[1m]WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@0.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
[[0m][[1m]WARN added 1899 packages from 840 contributors and audited 1903 packages in 51.626s

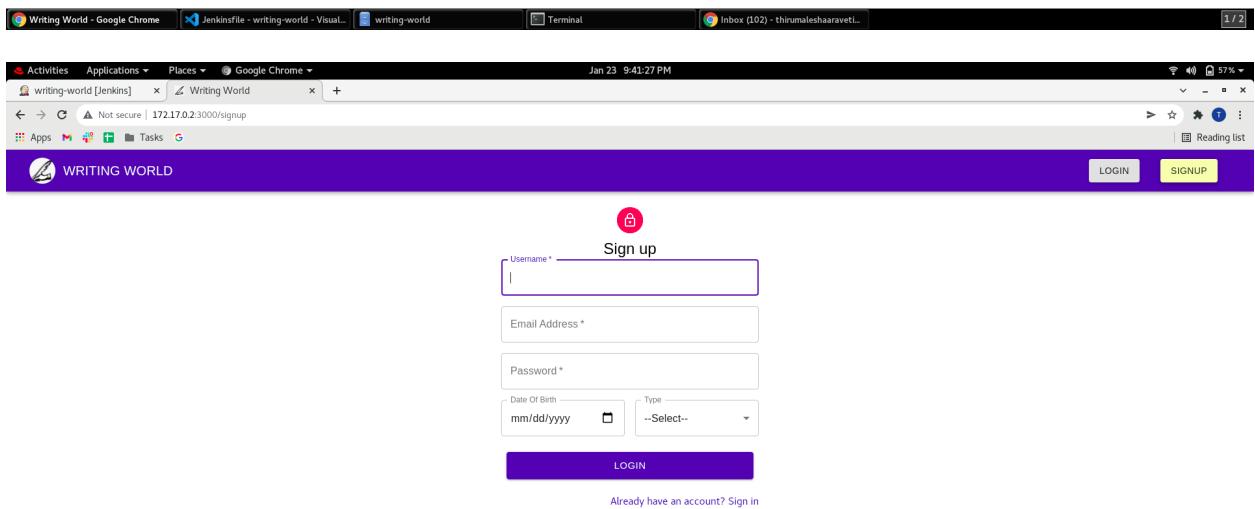
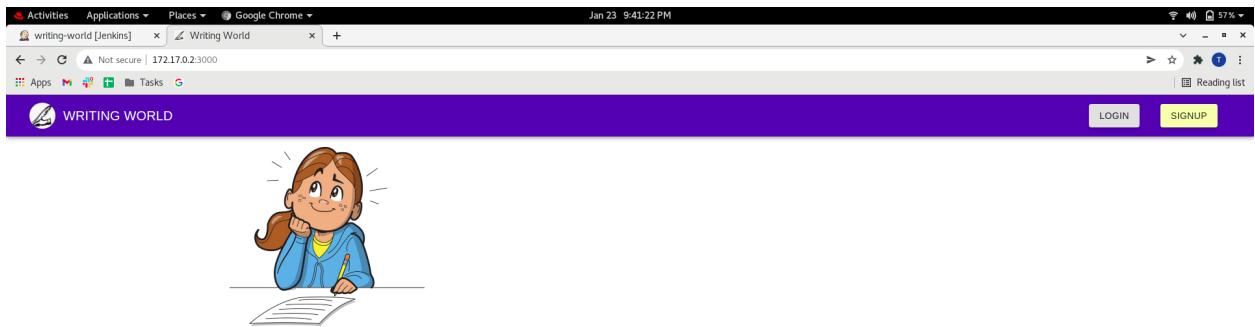
150 packages are looking for funding
  run `npm fund` for details

found 52 vulnerabilities (2 low, 37 moderate, 12 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
  Removing intermediate container 68fbcd7c8d98c
    --> c74d2a7be96b
Step 7/7 : CMD [ "npm", "start" ]
--> 7e63e7681414395
  Removing intermediate container c03e76814395
    --> a4df21941178
Successfully built a4df21941178
Successfully tagged writing-world:latest
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run container)
[Pipeline] sh
+ docker run -d --name writing-world writing-world
d92e1db6d29c70666590eb2d542bf6c7a03bc7a92a17d00890ddebc3c6d2f544
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.319.2

writing-world #10 Console [Jenk... README.md - writing-world - Visu... Images SRS_WRITING_WORLD - Google ... [Downloads] Terminal GitHub: Where the world builds s... 1 / 2

Web Pages



Activities Applications Places Google Chrome

writing-world [Jenkins] Writing World Jan 23 9:41:24 PM

Not secure | 172.17.0.2:3000/login

Apps Tasks

WRITING WORLD

Sign In

Email

password

LOGIN

[Forgot Password](#)

[Don't have an Account click here](#)

LOGIN **SIGNUP**

172.17.0.2:3000/login

Writing World - Google Chrome Jenkinsfile - writing-world - Visual... writing-world Terminal Inbox (102) - thirumaleshaaraveti...

Activities Applications Places Google Chrome

writing-world [Jenkins] Writing World Jan 23 9:53:21 PM

Not secure | 172.17.0.2:3000

Apps Tasks

WRITING WORLD

CREATEPOST **Profile**

hello just now



Kotha prapancham

ekkadi nundi vachamo telya

guest 2 months ago



CREATEPOST **Profile**

Writing World - Google Chrome Jenkinsfile - writing-world - Visual... writing-world Terminal Inbox (102) - thirumaleshaaraveti... Pictures Screenshot from 2022-01-23 21...

Activities Applications Places Google Chrome

writing-world [Jenkins] Writing World +

Not secure | 172.17.0.2:3000/createpost

Apps Tasks G

Reading list

WRITING WORLD

Title *

Post Image * Choose File No file chosen

Paragraph B I “ “ ⚡ ━ ━ ━ ━ ━ ━ ━ ━

PUBLISH

172.17.0.2:3000/crepost

Writing World - Google Chrome Jenkinsfile - writing-world - Visual... writing-world Terminal Inbox (102) - thirumaleshaaravet... Pictures Screenshot from 2022-01-23 21:23:33

Activities Applications ▾ Places ▾ Google Chrome ▾

writing-world [Jenkins] x Writing World x G nature image - Google Se... x (2) WhatsApp x +

Jan 23 9:53:12 PM

Not secure | 172.17.0.2:3000/crepost

Apps Tasks

CREATEPOST

WRITING WORLD

Title *

Kotha prapancham

Post Image *

Choose File istockphoto-1186330948-612x612.jpg

Paragraph

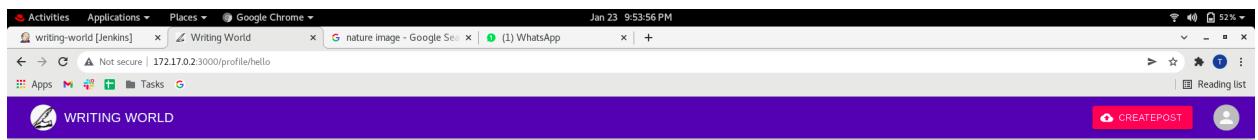
ekkadi nundi vachamo telyadu
ikkade perigam.
ikkade nercham.
ikkake adam.padam, paddam.
kulam ledu
matam ledu.
sneham ane bandam tappa
vikasinchey vayasulu
badi ane chettu kinda
upadyayulu ane kiranalatho
oka chinna mokka nundi
oka vruksham laga perigamu.
evariki upayoga padani oka mokka nundi
andarki needu niche okka vrukshyam la edigam.
Badi(school) oka marachi poleni gnakapam.



Kotha prapancham

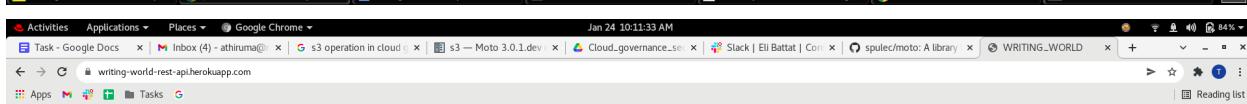
ekkadi nundi vachamo telyadu
ikkade perigam.
ikkade nercham.
ikkake adam.padam, paddam.
kulam ledu
matam ledu.
sneham ane bandam tappa

Show all



Backend

```
Activities Applications ▾ Places ▾ Terminal ▾
Jan 24 10:26:06 AM
Terminal
File Edit View Search Terminal Help
bash-4.4$ tree .
.
├── app.js
├── controllers
│   ├── Comments
│   │   └── DeleteComment.js
│   ├── Likes
│   │   ├── like.js
│   │   └── unlike.js
│   ├── posts
│   │   ├── createPost.js
│   │   ├── deletePost.js
│   │   ├── GetAllPosts.js
│   │   ├── MyPosts.js
│   │   └── PostData.js
│   └── profile
│       ├── MyProfile.js
│       └── UpdateProfile.js
├── UserCredentials
│   ├── ForgotPassword.js
│   ├── login.js
│   └── sendVerificationMail.js
└── signup.js
├── package.json
└── package-lock.json
Profile
public
├── index.html
└── README.md
serviceAccountKey.json
utils
├── admin.js
└── config.js
└── firebase.js
8 directories, 26 files
bash-4.4$
```



REST API

| Methods | POST | GET | DELETE | PUT |
|-----------------|--|------------------|----------------------------------|------------------------------------|
| /login | • email • password | /posts | /deletepost/:postId | /UpdateProfile • photo |
| /signup | • username • email • password • type • dob | /post/:postId | /post/:postId/comment/:commentId | /post/:postId/comment • comment |
| /createpost | • title • body • image | /myposts | /like/:postId | |
| /like/:postId | | /profile/:userId | | |
| /forgotpassword | | | | |



Firebase Pages

The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation sidebar includes sections for Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning, Release & Monitor, Analytics, Engage, and Extensions. The main area displays a collection named 'comments' under a document ID 'BpE1Smf4lLQop1T5y4iW'. The collection contains three documents: 'likes', 'posts', and 'users'. The 'users' document has the following fields:

- comment: "Love"
- createdAt: "2021-10-18T16:13:10.214Z"
- imageURL: "https://firebasestorage.googleapis.com/v0/b/writingworld-9ba1f.appspot.com/o/profileImages%2Fdhanunjay.jpg?alt=media"
- postId: "RR30NbVs1Ku1m0zcPGE"
- username: "dhanunjay"

The screenshot shows the Firebase Authentication interface. The main area displays a table of users. One user, 'Deleted macha@gmail.com', is highlighted with a green checkmark and a tooltip. The table columns are Identifier, Providers, Created, Signed In, and User UID. The data includes:

| Identifier | Providers | Created | Signed In | User UID |
|-----------------------------|-----------|--------------|--------------|---------------------------------|
| satyanarayanan@gmail.c... | ✉️ | Dec 30, 2021 | Dec 30, 2021 | 3FNbNncktsVJBuDvRQQwv9y82 |
| thiru1083@gmail.com | ✉️ | Dec 30, 2021 | Dec 30, 2021 | g2kiH4DfncM1YRB3yMVXhsfxq53 |
| guest@mailer.com | ✉️ | Nov 17, 2021 | Nov 17, 2021 | kEakgNA2D0bJUNEuht0tqTFqdP... |
| prathyub123@gmail.com | ✉️ | Nov 3, 2021 | Nov 3, 2021 | kCaTVkx9fbFKV2IRw4QmdRTJTB3 |
| indvseng2018@gmail.com | ✉️ | Oct 18, 2021 | Oct 18, 2021 | XK1RufREyjTBUnmxYa6UloQLL2 |
| supersuper1075@gmail.co... | ✉️ | Jul 17, 2021 | Jul 18, 2021 | p8KGZV3qmkrhnnKE5fbIrd3s4K3 |
| r161083@ruktrkv.ac.in | ✉️ | Jul 14, 2021 | Jul 18, 2021 | I1bhSS2crVxENk4m4q4EHDS812 |
| thirumaleshaaraveti@gmai... | ✉️ | Jul 14, 2021 | Dec 8, 2021 | bJegHUCacRahHMZBic7zNUzjhIZX... |

Github repositories

Frontend : <https://github.com/thirumaleshaaraveti/writing-world>

Backend : <https://github.com/thirumaleshaaraveti/writing-world-rest-api>

Application Urls

Frontend : <https://writingworld-9ba1f.web.app/>

Backend : <https://writing-world-rest-api.herokuapp.com>

References

[https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

<https://en.wikipedia.org/wiki/Node.js>

<https://cloud.google.com/firestore/docs#:~:text=Cloud%20Firestore%20is%20a%20NoSQL,web%20apps%20%2D%20at%20global%20scale.>

<https://www.bmc.com/blogs/devops-basics-introduction/>

<https://aws.amazon.com/what-is-cloud-computing/>

<https://www.docker.com/resources/what-container>

<https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology>

<https://console.firebaseio.google.com/u/0/project/writingworld-9ba1f.firebaseio/data/~2Fcomments~2FBpE1Smf4iLQop1T5y4iW>