# A Project/Dissertation Review-1 Report

# on

## OurApp using Nodejs

***Submitted in partial fulfillment of the***

***requirement for the award of the degree of***

Computer Science and Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under the Supervision of**
Mr. Hradesh Kumar
Assistant Professor

**Submitted By:**
Devanshi Gupta &Kalpit kulshrestha
18SCSE1050033 & 18SCSE1050010

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING / DEPARTMENT OF COMPUTERAPPLICATION**
**GALGOTIAS UNIVERSITY, GREATER NOIDA**
**INDIA**
**DECEMBER,2021**

# SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
# GALGOTIAS UNIVERSITY, GREATER NOIDA

## CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **" OurApp using Nodejs "** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY,2021** to **DECEMBER,2021** under the supervision of **Mr. Hradesh Kumar, Assistant Professor**, **Department of Computer Science and Engineering**, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

18SCSE1050033-DEVANSHI GUPTA

18SCSE1050010-KALPIT KULSHRESTHA

**This is to certify that the above statement made by the candidates is correct to the best of my knowledge.**

Mr. Hradesh Kumar

Assistant Professor

# CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **18SCSE1050033-DEVANSHI GUPTA, 18SCSE1050010-KALPIT KULSHRESTHA** has been held on _________________ and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.**

**Signature of Examiner(s)** **Signature of Supervisor(s)**

**Signature of Project Coordinator**

**Signature of Dean**

Date: December,2021

Place: Greater Noida

# ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor, Mr. Hradesh Kumar, Assistant Professor, Computer Science and Engineering, Galgotias University, Greater Noida, for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his friendship, empathy, and great sense of humor. I am extending my heartfelt thanks to his wife, family for their acceptance and patience during the discussion I had with him on research work and thesis preparation.

I am extremely grateful to my parents, Friends for their love, prayers, caring and sacrifices for educating and preparing me for my future. Also I express my thanks to my sisters, brother, sister in law and brother in laws for their support and valuable prayers.

# ABSTRACT

OurApp is a simple real-world application where users can write short tweets, follow each other and also chat with each other. You can have a look at the final product's demo video before starting the project. This project is best for those who want to dive deeper into a full stack using Nodejs and MongoDB after learning HTML, CSS, and JS. Building a full stack application single-handedly is a tough task, but learning and building such applications will help you master your skills. Building this project will be a challenging task where you will get to learn and explore all about the MVC pattern, NoSQL Database (MongoDB), and much more.

## LIST OF CONTENT

## LIST OF FIGURES

# 1. Introduction

Objective- Build a full stack web application - OurApp (a social media app) using Node.JS, Express.Js, MongoDB, EJS (Template Engine for server-side rendering) and deploying the application.

The social media platform allows for interactions between users from different domains, leading to a stronger social structure. A prominent feature of this structure is the production of large amounts of information, which gives users an idea of a different service value. However, the decline of such information capacity is sometimes reflected in the inability of users to obtain reliable information for their use in times of need. Social media is already so entrenched in our daily lives that people rely on every need, from daily news and critical events updates to entertainment, communication with family and friends, reviews and recommendations about products / services and locations, fulfillment. of emotional needs, workplace management, and compliance with the latest hashion issues, to name a few.

OurApp is a simple real-world application where users can write short tweets, follow each other and also chat with each other. You can have a look at the final product's demo video before starting the project. This project is best for those who want to dive deeper into a full stack using Nodejs and MongoDB after learning HTML, CSS, and JS. Building a full stack application single-handedly is a tough task, but learning and building such applications will help you master your skills. Building this project will be a challenging task where you will get to learn and explore all about the MVC pattern, NoSQL Database (MongoDB), and much more.

This study investigates the most effective time to post on Instagram to improve content engagement (i.e., likes and comments). 2,958 Indonesian food product postings were analyzed using negative binomial regression to predict the number of likes and comments. This study reveals that the most effective posting dates are Mondays, Tuesdays, Wednesdays, Thursdays, and Fridays as content posted on Instagram these days results in a higher number of likes and comments. In addition, the study also encourages foods to post their content in the morning and afternoon hours as these times of the day encourage more popularity and comment. The findings of this study will help brands and social media managers incorporate planning into their social media marketing platform on Instagram.

There are two types of communication users; digital natives and digital immigrants. The Indigenous people are here
were born after 1980. they came to this world at a time when there were social media. However, digital immigrants are those born before 1980 and transformed their lives into digital media (AntonSon and Christopher, 2014). Social media forums vary from Web blogs to micro sharing platforms, to live sources to social media and much more. (AntonSon and Christopher, 2014). The new generation discussed in this research paper is school children between the ages of 12 and 19. The few forums they use are Facebook, YouTube, Google, and many more to discuss on findings from this study. A new generation of people who will lead our world in the future, should well educated so that they can influence this world and make Egypt a better country on the road to success. Therefore, research should be done on a variety of factors, which may have a negative impact or well. This study aims to assess how often students use social media, and whether they have it Any impact on their academic performance. If it affects their academic performance, how is it done do it.

## **Primary goal**

- Build this app from scratch and use it in your real life; best by adding some additional
advanced features to this base project.

- The main goal is to master your NodeJs and MongoDB skills and begin your full stack journey by developing this project.

- After building this app, your goal should be implementing private chat, image upload
in the post section, and like, comment feature to make it more practical.

## Project Stages

We can divide the project based on the stack used:

- HTML, CSS, Bootstrap: Building the UI of the application
- Serving HTML dynamically and use of EJS (template engine)
- Familiarising the NodeJS environment
- ExpressJS: Framework for creating servers.
- MongoDB: Using NoSQL Database.
- Socket.IO: Building live chatting feature
- GitHub: To publish your project.
- Heroku: Deploy the full stack application

We begin with descriptions of some important techniques related to the analysis of informal text data:

Native language processing— (NLP) is a field of computer science, artificial intelligence and languages related to the interaction between computers and human (natural) languages. In particular, a computer program that delivers comprehensive information on native language insertion and / or output production of native language.

News statistics — a measure of different quality attributes and the value of text stories (random data). Some of these qualities are: emotions, relationships and youth.

Idea mine — idea mines (emotion mines, ideas) is a research center that attempts to create automated programs to find a person's perspective on a text written in the original language.

Scraping — collecting data from online forums and other websites in the form of random text also known as site scraping, web harvesting and web data extraction.

Emotional analysis — emotional analysis refers to the use of natural language analysis, integrated language learning and textual analysis to identify and extract specific information from sources.

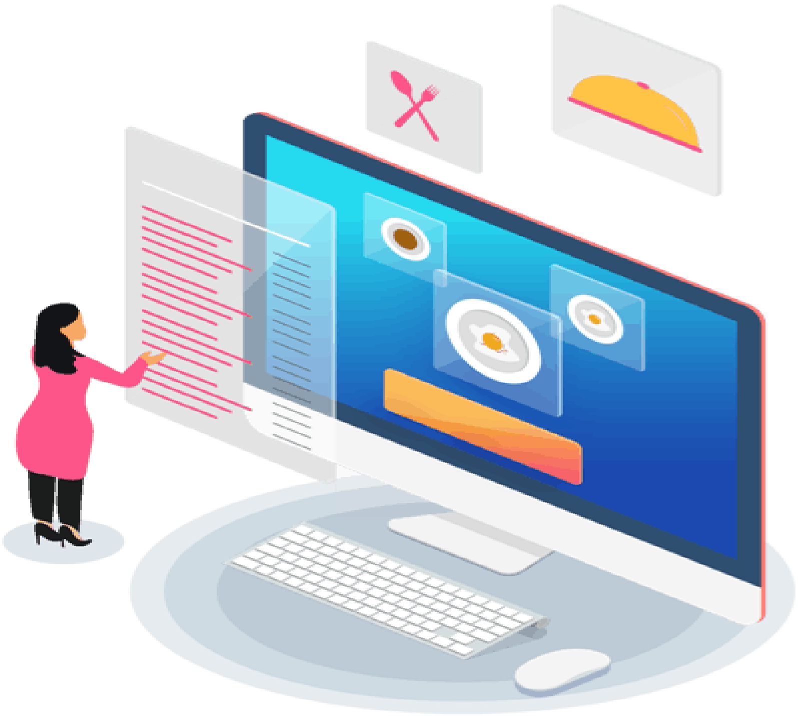
Text analysis — including retrieval (IR), lexical analysis to study word distribution, pattern recognition, marking / annotation, data extraction, data mining techniques including link analysis and correlation, visualization and predictive analysis.

## Research challenges

Social media analysis and analysis provides a rich source of educational research challenges for social scientists, computer scientists and funding organizations.

Challenges include:

Scraping — although social media data is accessible through APIs, due to the amount of data transactions, many large sources such as Facebook and Google make it very difficult for students to gain full access to their 'raw' data; There are very few social data sources that provide affordable data donations to academics and researchers. News services such as Thomson Reuters and Bloomberg often charge a premium for access to their data. In contrast, Twitter recently announced a Twitter Data Grants program, in which researchers could request access to Twitter social media tweets and historical data for information on its large data set (Twitter has over 500 million tweets per day)



Data purification — cleaning up random text data (e.g., standard text), especially real-time data streamed at high frequency, still has many problems and challenges for research.

Complete data sources — researchers are increasingly compiling and integrating novel data sources: communication platform data, real-time market and customer data and geospatial data for analysis.

Data protection — once you have created a 'big data' service, data needs protection, identity and IP issues are resolved (i.e., storing discarded data violates many of the publisher's terms of service), and users are given different levels. access; otherwise, users may try to ‘absorb’ all the important data from the website.

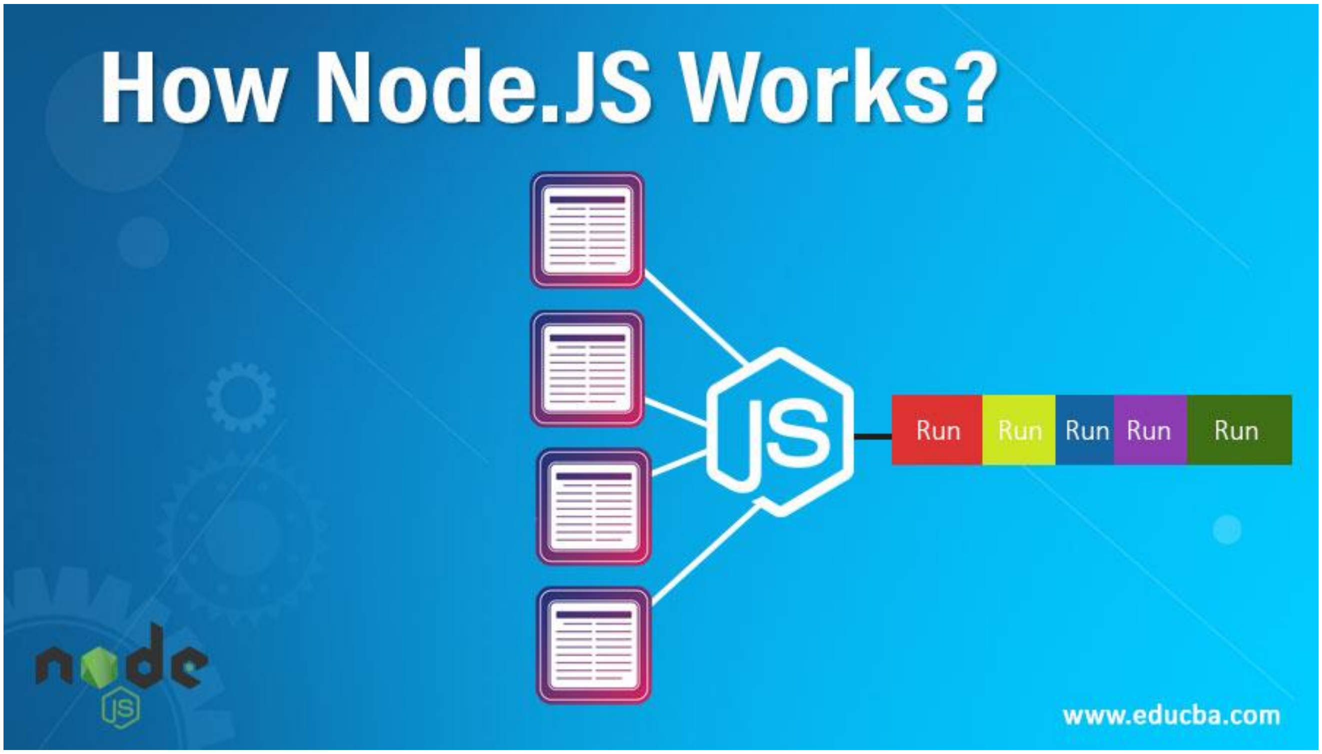
Data analysis — a complex analysis of social media data to discover ideas mines (e.g., emotional analysis) still raises a number of challenges due to foreign languages, foreign words, slang, spelling errors and natural language development.

Statistics dashboards — many social media platforms require users to write APIs in order to access feeds or statistical models for programming languages, such as Java. Although it makes sense for computer scientists, these skills are often superior to most researchers (social science). Non-programmatic communication is required to provide what might be called ‘deep access’ to ‘raw’ data, for example, to configure APIs, integrate social media feeds, integrate complete resources and develop analytical models.

Data visualization — visual representation of data in which information is summarized in a particular system for the purpose of conveying information clearly and effectively using graphic methods. Given the size of the data involved, visualization is becoming increasingly important.

- **Nodejs :**

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project. Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant. A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

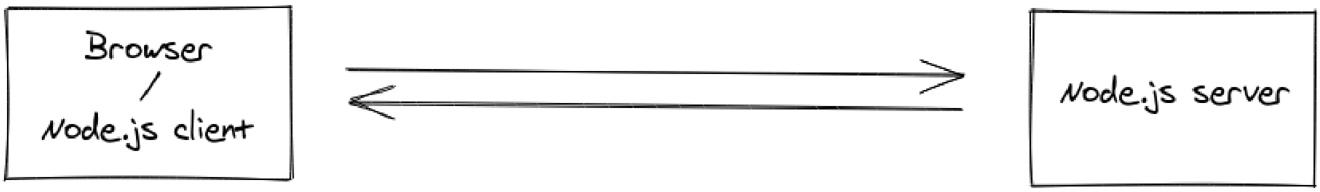


## ➢ Socket.io:

Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server. It consists of:

- a Node.js server: Source | API

- a Javascript client library for the browser (which can be also run from Node.js)



## How does that work?

The client will try to establish a WebSocket connection if possible, and will fall back on HTTP long polling if not.

WebSocket is a communication protocol which provides a full-duplex and low-latency channel between the server and the browser. More information can be found here.

So, in the best-case scenario, provided that:

- the browser supports WebSocket (97% of all browsers in 2020)
- there is no element (proxy, firewall, ...) preventing WebSocket connections between the client and the server.

**BENEFIT POINT :**

### Performant

In most cases, the connection will be established with WebSocket, providing a low-overhead communication channel between the server and the client.

### Reliable

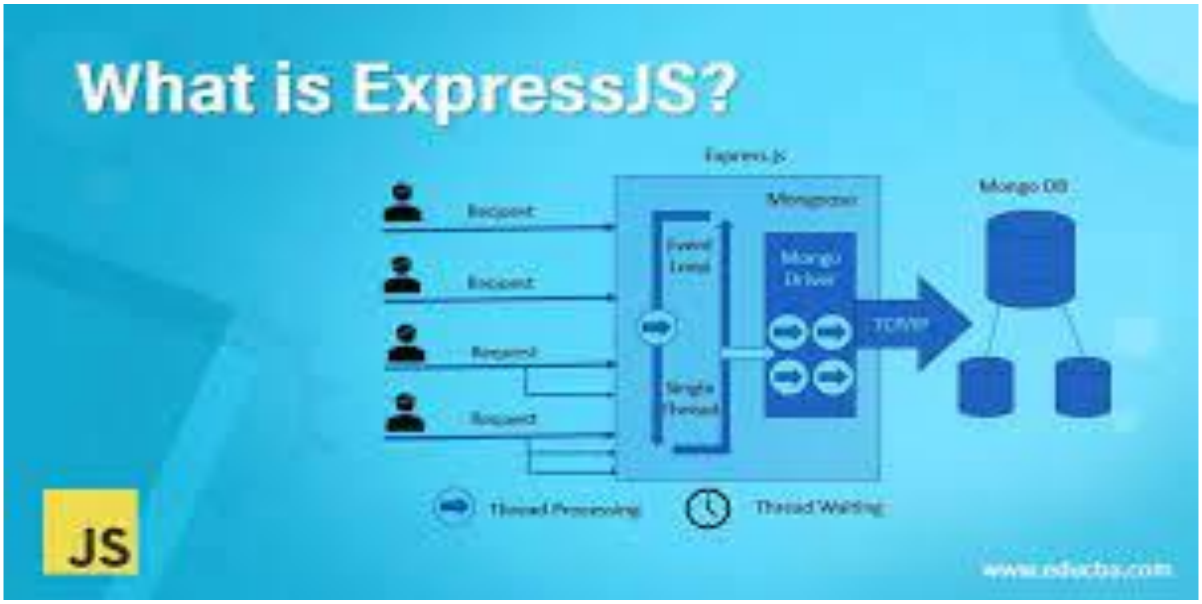
Rest assured! In case the WebSocket connection is not possible, it will fall back to HTTP long-polling. And if the connection is lost, the client will automatically try to reconnect.

### Scalable

Scale to multiple servers and send events to all connected clients with ease.

## ➢ ExpressJS

ExpressJS is a prebuilt NodeJS framework that can help you in creating server-side web applications faster and smarter. Simplicity, minimalism, flexibility, scalability are some of its characteristics and since it is made in NodeJS itself, it inherited its performance as well.

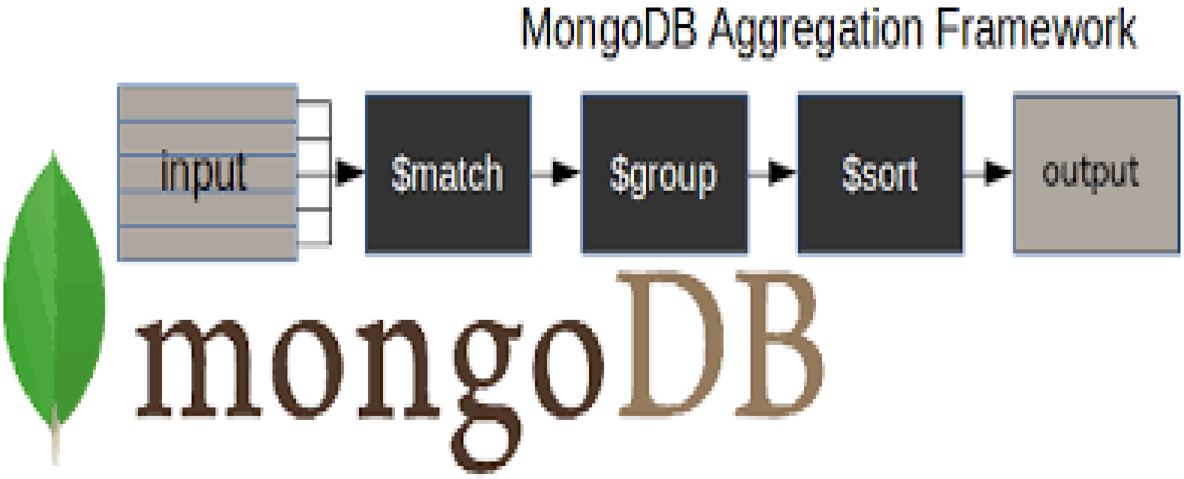


## The MEAN Stack

The combination of AngularJS, ExpressJS, and NodeJS may now seem to sound interesting. You have completely given up your dependency on LAMP languages, with the exception of MySQL database. But since we have been setting new trends in web based app development, why use an old database built? Let’s go for something better, something easier, robust and without the hassle of a complicated database schema language

## ➢ MongoDB

MongoDB is a document database designed for ease of development and scaling. The Manual introduces key concepts in MongoDB, presents the query language, and provides operational and administrative considerations and procedures as well as a comprehensive reference section.

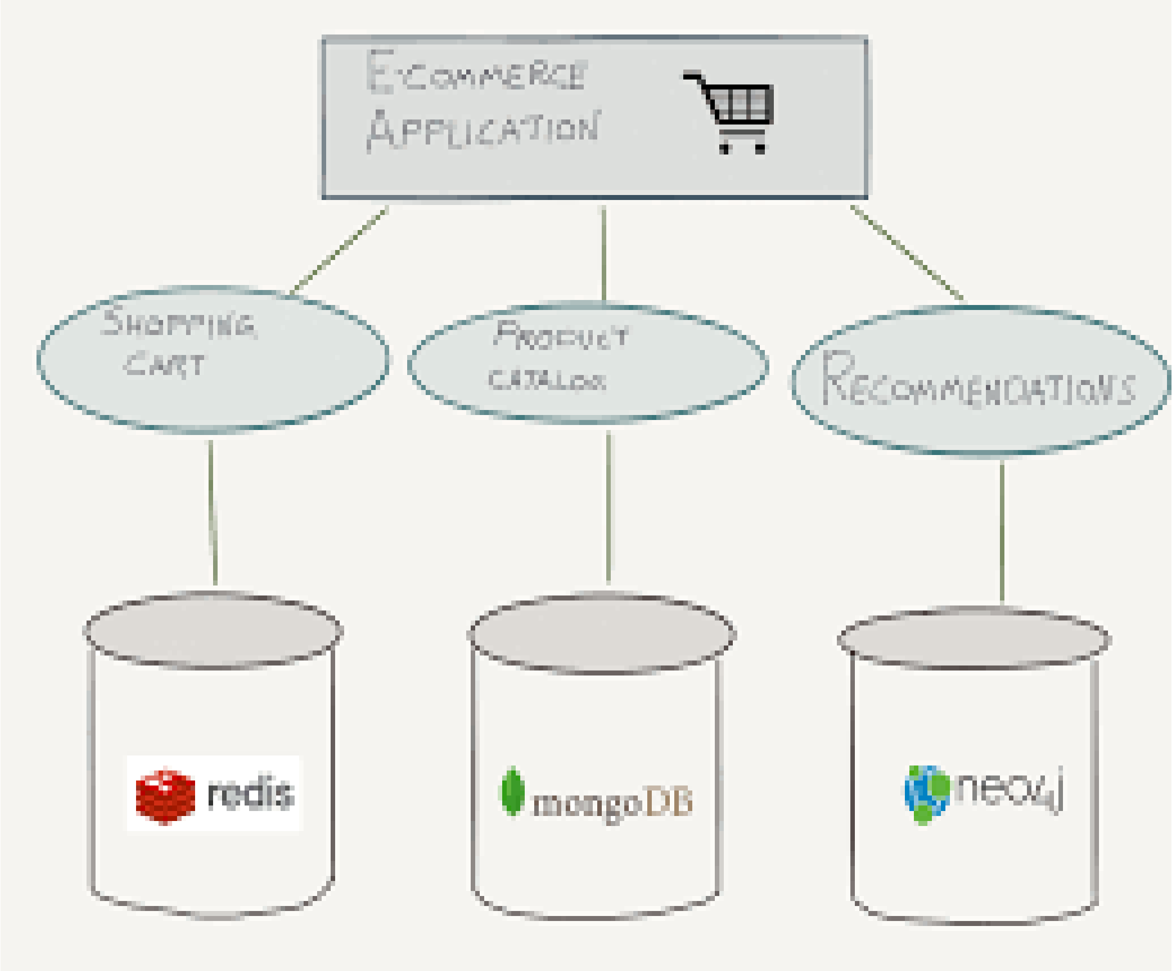


MongoDB offers both local and cloud-hosted deployment options:

- For locally hosted deployments, MongoDB offers both a *Community* and an *Enterprise* version of the database:

- MongoDB Community is the source available and free to use edition of MongoDB.
- MongoDB Enterprise is available as part of the MongoDB Enterprise Advanced subscription and includes comprehensive support for your MongoDB deployment. MongoDB Enterprise also adds enterprise-focused features such as LDAP and Kerberos support, on-disk encryption, and auditing.

- MongoDB Atlas is a hosted MongoDB Enterprise service option in the cloud which requires no installation overhead and offers a free tier to get started.

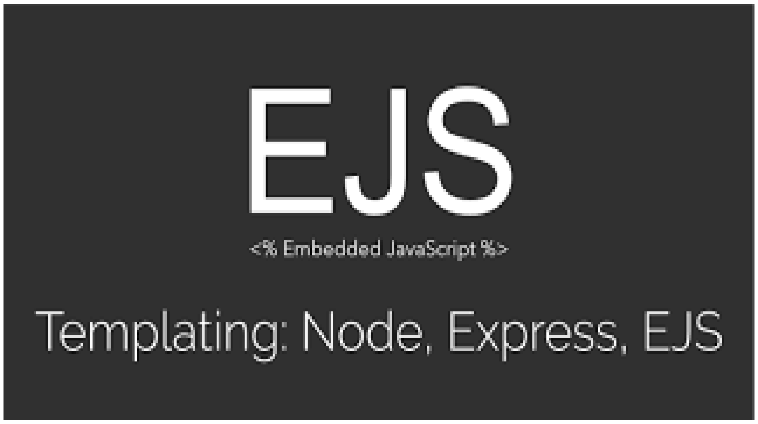


## HTML/CSS

HTML is the standard markup language for website development. It describes the structure of the web page which is to be designed.it consist of series of element, that tells the browser how to display the content. CSS controls the layout of the web pages all at once. It can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

## EJS

An EJS file contains code written in the Embedded JavaScript (EJS) templating language, which is utilized to generate HTML markup using JavaScript. It is typically used as part of a web application and includes tags that the EJS engine replaces with information from a database to produce an .HTML webpage at runtime.



### ➢ GITHUB

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

This tutorial teaches you GitHub essentials like repositories, branches, commits, and pull requests. You'll create your own Hello World repository and learn GitHub's pull request workflow, a popular way to create and review code.

In this quickstart guide, you will:

- Create and use a repository
- Start and manage a new branch
- Make changes to a file and push them to GitHub as commits
- Open and merge a pull request

we suggest three image content effects:

“Consequence of existence” (in common forums where only text and photo and image posts are combined): The image content of social media posts helps posts stand out in the media cluster where most of the online content is text-based. Additionally, some social networks allow users to share a link to an image posted on other social media platforms. We specifically classify these linked images into images that appear quickly in our analysis in order to capture the benefit of displaying the image content directly in the post.

"Image Icon Effect": The image content of social media posts may provide information, beauty, or self-improvement without the content of the text and as a result may increase the overall appeal of the post. However, an image with unpleasant content or low quality may cause and lead to low user engagement. In this study, we identify four aspects of imagery — color, facial expression and mood, image source, and image quality — and assess their effects on social media user engagement.

"Equal image-text effect": Both images and text content of social media posts are used by the author to express a particular point of view. Research has shown that images that are not relevant to printed ads create more difficulties for readers to understand the main message and thus lead to worse ad recall and unpleasant attitudes toward the ad (Heckler and Childers 1992; Lee and Mason 1999). We think the same image effect – text may be present in the context of a social media platform.

# 2. Literature Survey

Given the inconsistencies in the use of keywords in social media research, manual search, instead of keywords, is considered appropriate to identify existing publications on social media. In addition, since keywords in social media books often overlap topics and / or theories in other related research areas, keyword searches may reveal insignificant articles. For example, keyword search "Network Network" returns articles related to social network theories, which are not part of the social media platform.

It is web-based App that can be used to create blog, a small real-world project where you will learn about authentication, socket, server-side rendering and mongodB.

High-Level Approach

• Building the UI of the app: the guest page, the home dashboard, profile page, create a
post, single post screen, and more.

• Set up your server and dynamically render HTML.

• Implementing authentication (login and signup) for users.

• Implementing routing and MVC patterns.

• Building models for database and connecting database to server.

• Building live chat using socket.IO.

• Adding validators and security features to the project.

• Publish to Github and finally deploy to heroku.

## Task 1: Prototyping the Application and environment setup

To develop application, we must setup our environment for the development of the application. This application consists of full stack development by master NodeJs and MongoDB.

To get a better understanding of this project, we have to clarify this application by reading other research paper and watching the some demo video. To setup the environment we are going to download some application environment and setting their configuration as per the project guide.

Requirement :

- Explore the OurApp’s demo.

- We are using visual studio code for javascript.
- Installing NodeJs on Device. Here we are using NPM.
- A database for the storage of the data, we are using mongodB atlas account to manage your data in the cloud.
- This is a nodeJs app built using Express.Js framework to build the web server and project following the MVC pattern.
- Install necessary dependency packages.

• Install necessary dependency packages (you may refer to the ones mentioned below) using npm.

## Task 2: Building the pages layout using HTML, CSS.

->Layout of the pages

- Home-guest page
- Home-dashboard page
- Create-post page
- Profile page
- Single post screen page
- Chat-box page

Requirements :

- Create the basic UI for Guest page and dashboard.
- Flexbox plays the most important role while bulding the layouts.
- Build all pages using own CSS.
- Implement media queries for mobile view.

At the end of this task, the html pages must be made. Building one 404 error code page would be a plus for the app

## Task 3: Building the Express server and rendering the HTML dynamically

we will start sing Node.js and Express.js to build the backend of the application. This task takes some steps to procced and that is.

1. Building the webserver using Express.
2. Knowing about the file structure and setting up the project.
3. Using EJS (template engine) to build dynamic contents into the HTML templates.
4. Building the HTML templates in a reusable fashion.

**Requirement:**

- Start building a server using express. Express.js, which is a lightweight framework for creating web servers, makes it easier to organize your application's functionality with middlewares and routing.
- After building the server, your project folder must be consisting of server.js file along with package.json, package-lock.json, and node_modules.
- Now create a folder named views inside your main folder and place all HTML files inside it and create a folder public and place the CSS files inside it.

At the end of this task, your server should be listening for requests at a port. And, your views folder should contain the partials and pages using EJS. The terminal should look like this after using nodemon when starting the server.

## Task 4: using MongoDB database with express server

In this project, we will be using MongoDB as the database.

1. Creating a MongoDB database atlas account to manage our data.
2. Connecting the database to the express server.
3. Knowing about MVC architecture in Node,js application.

Requirement:

- Create an Atlas account. Connect to your cluster, insert and view data in your cluster.
- Create a db.js file in your folder and create a connection to a MongoDB instance that returns the reference to the database.
- Make the connection in such a way that the server starts listening to requests only after the connection to the database is made.
- Explore the MVC architecture before starting to implement the functionalities in the next task.

At this point of the project, the integration of the database must be completed. You should have got an idea about the MVC architecture which we will be building from the next task. Get hands-on with MongoDB.

## Task 5: Implementing user registration and login

In this task, we will be building the authentication for the app. Let's divide this task into

smaller steps:

1. Building models for users.
2. Hashing the password before storing it in the database.

3. Implementing session-based authentication.
4. Using Gravatar for User profile photos.

Requirements:

- Implement routing of all routes in a separate file and call controller functions, respectively.
- Separate the controllers and models w.r.t their functionalities. For example, make user related functions in userController.js and models in userModel.js. Validate attributes before storing it to the database. Like this, divide others based on functionalities.
- Implement hashing before storing the passwords to the database. For hashing bcryptjs and md5 packages can be used.
- You can use Mongodb without Mongoose also. But using mongoose makes it easier to implement the functionalities.

At the end of this task, a user must be able to create an account and log in to the app and visit their homepage, and should have the liberty to access different sections of his profiles.

## Task 6: Implementing core functionalities of users

In this task, we will be building the core functionalities of the app. Let's divide this task into
smaller steps:

1. Building the model for posts.
2. Letting users create and edit their posts.
3. Letting users follow each other.

4. User dashboard will consist of posts of the other users that they follow.

**Requirements**

- Build a database model for the posts by including all the attributes which are required
- to be saved for the post.
- Write your controller function to save and edit the posts by the user.
- Build the model in such a fashion that the followers and followings can be retrieved by the user.
- Fetch the posts of the users whom you follow in their respective dashboard.

At the end of this task, users should be able to create, edit and delete posts. Users can follow each other and view each others' posts.

## Task 7: Implementing the live chat and live search feature

In this task, we will be building a live chat and live search feature of the app. Let's divide this task into smaller steps:

1. Use front-end javascript to build a search panel.
2. Create one separate folder for the front-end JS files and bind it using webpack.
3. Similarly, build a chat panel and bind it.
4. Use Socket.IO for chat.

**Requirements**

- Create one separate folder frontend-js and install webpack, webpack-cli, @babel/core, and @babel-loader.
- Configure the webpack.config.js file and change the watch option according to it in the package.json file.
- Now run the npm watch and check the public folder, there will be a main-bundled.js file in it.
- Start building the frontend files for search and chat panels.
- Create a controller function for searching the results by posts.
- Use socket.io for building chat functionalities.

At the end of this task, users can search posts and chat with other users who are currently active.

## Task 8: Implementing live validation in the registration form and adding CSRF protection

In this task, we will be building live validation for the registration form and implementing csrf security to the app.

Requirements:

- Implement the live validation as separate frontend-js done before for live search.
- Prevent Cross-Site Request Forgery (CSRF) in your application using csurf package.

At the end of this task, the signup form should have live validation as shown in feed.

## Task 9: Pushing the project to GitHub

In this task, you will be pushing your project into GitHub. Make a good README for your
project so that your project is well-documented.

Requirements:

- Initialize the project as a github repository using git init.
- You will get to see a .git file inside your repository and the unstaged files.
- Ignore the node_modules and .env file (if any) before pushing it to GitHub using
- .gitignore file.
- Commit your changes and push it to github.

At the end of this task, your project will be published onto GitHub.

## Task 10: Making the app live

In this task, we will be deploying it to Heroku to make it live.

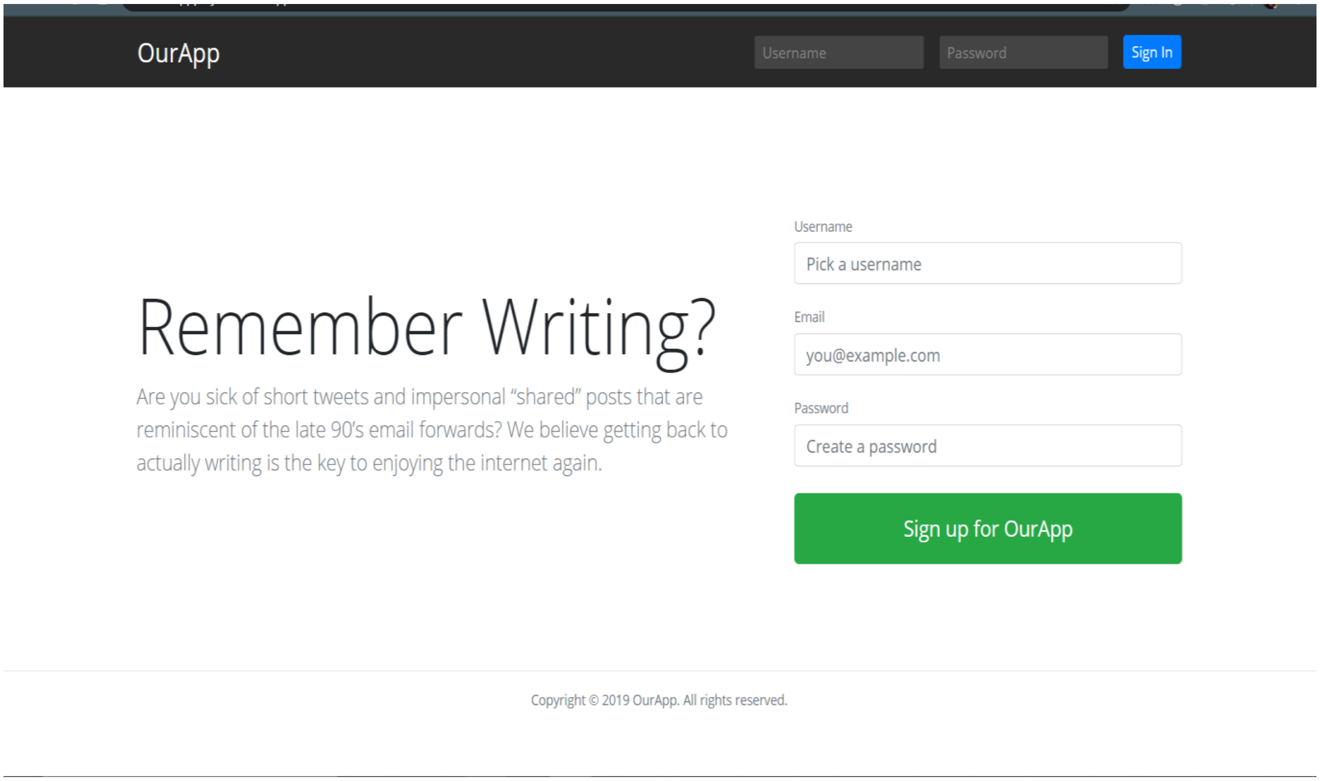
Requirements

- Go ahead and go to Heroku and create your free account. The dashboard is incredibly simple and user-friendly so do explore it.
- If you have secret key-value pairs in your .env file, use config vars in the setting menu of heroku to store those values before deploying.
- Now use the command line to deploy your node app.

At the end of this task, your project should become live and you can share the link among your peers, add it in your resume and do add it in your GitHub repository

## 3. **Project Design**

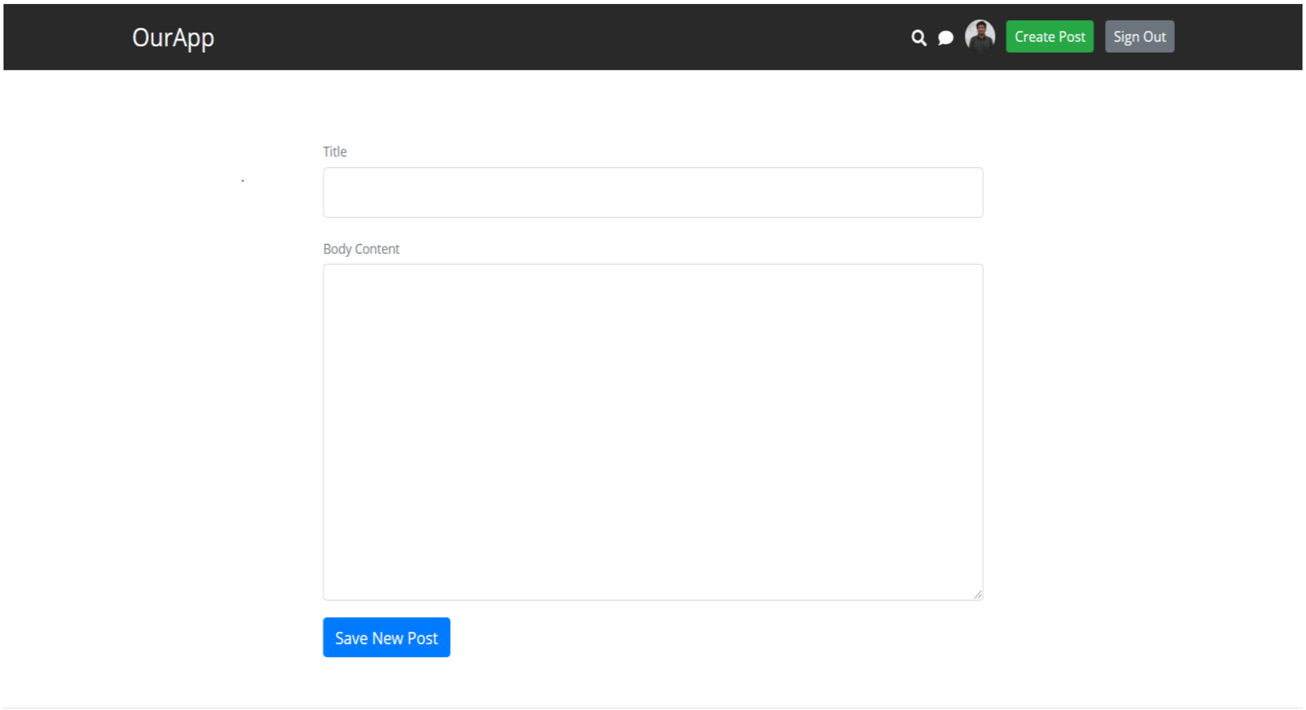
The design of OurApp will be look like this.

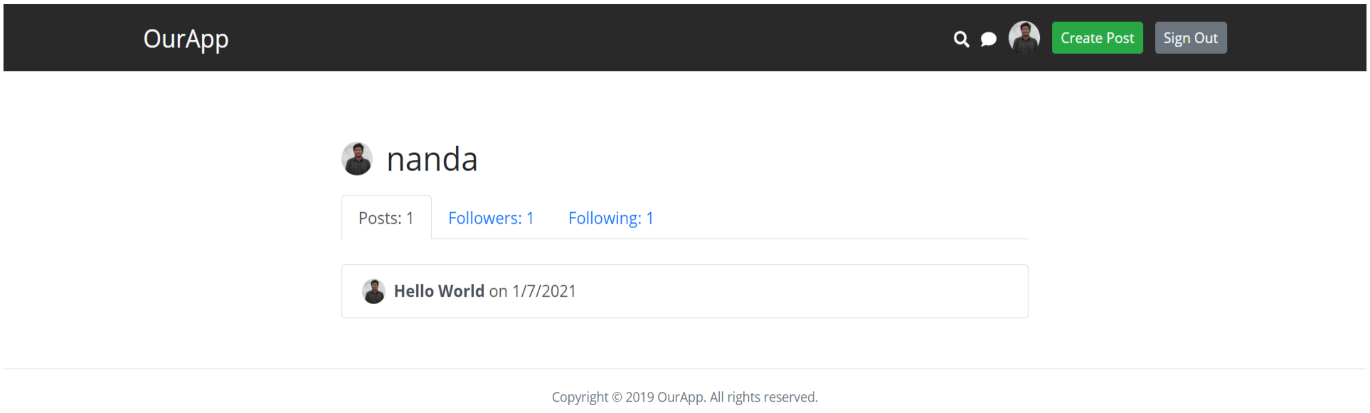


Remember Writing?

Are you sick of short tweets and impersonal “shared” posts that are reminiscent of the late 90’s email forwards? We believe getting back to actually writing is the key to enjoying the internet again.

Copyright © 2019 OurApp. All rights reserved.

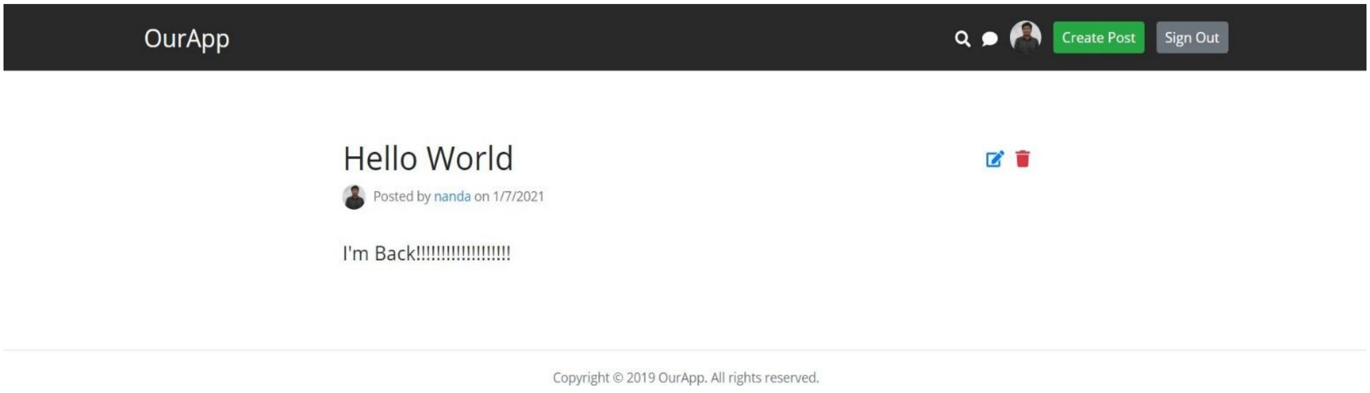

OurApp
Create Post
Sign Out
Title
Body Content
Save New Post



nanda

**Hello World** on 1/7/2021

Copyright © 2019 OurApp. All rights reserved.

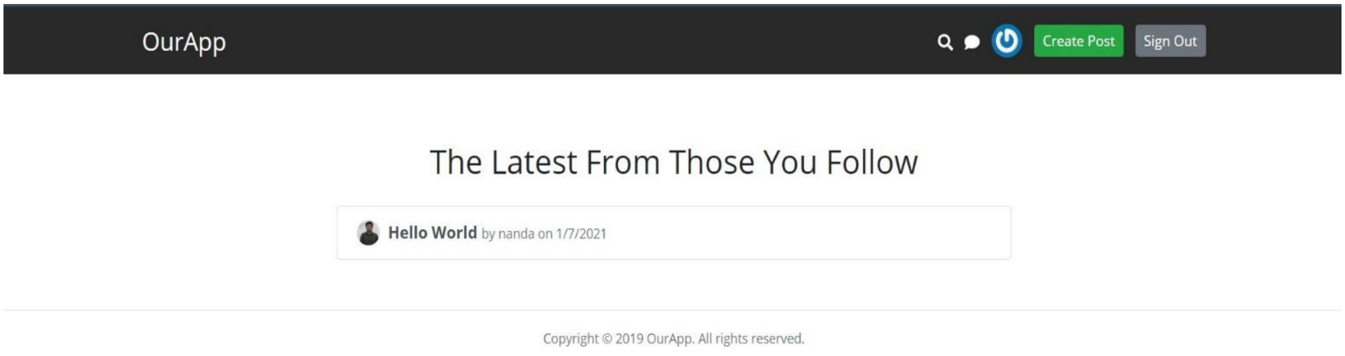


Hello World

Posted by nanda on 1/7/2021

I'm Back!!!!!!!!!!!!!!!!!!

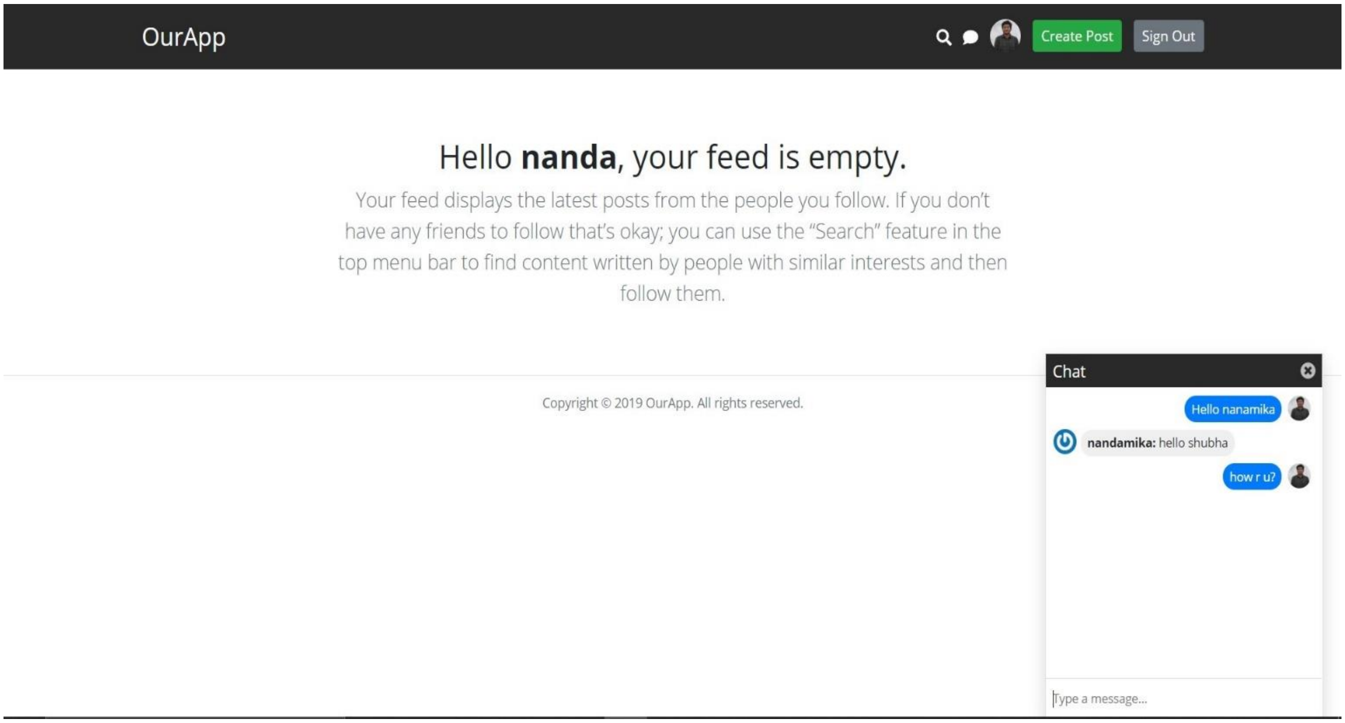
Copyright © 2019 OurApp. All rights reserved.



The Latest From Those You Follow

**Hello World** by nanda on 1/7/2021

Copyright © 2019 OurApp. All rights reserved.



Hello **nanda**, your feed is empty.

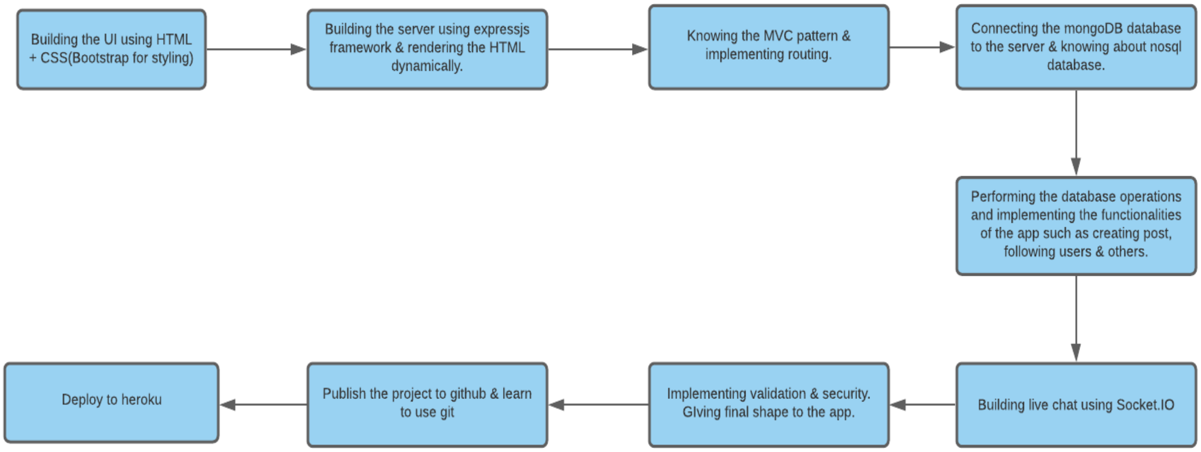
Your feed displays the latest posts from the people you follow. If you don't have any friends to follow that's okay; you can use the "Search" feature in the top menu bar to find content written by people with similar interests and then follow them.

Copyright © 2019 OurApp. All rights reserved.

Hello nanamika

**nandamika:** hello shubha

how r u?


Building the UI using HTML + CSS(Bootstrap for styling)
Building the server using expressjs framework & rendering the HTML dynamically.
Knowing the MVC pattern & implementing routing.
Connecting the mongoDB database to the server & knowing about nosql database.
Performing the database operations and implementing the functionalities of the app such as creating post, following users & others.
Building live chat using Socket.IO
Implementing validation & security. GIving final shape to the app.
Publish the project to github & learn to use git
Deploy to heroku

# 4. PRON AND CONS

1.A lack of private involvement: AI instruments will operate with minor errors which will not have an effect on the diagnosis method or operation significantly. not like computers, physicians will violate some rules to try to their utmost to avoid wasting a person’s life.

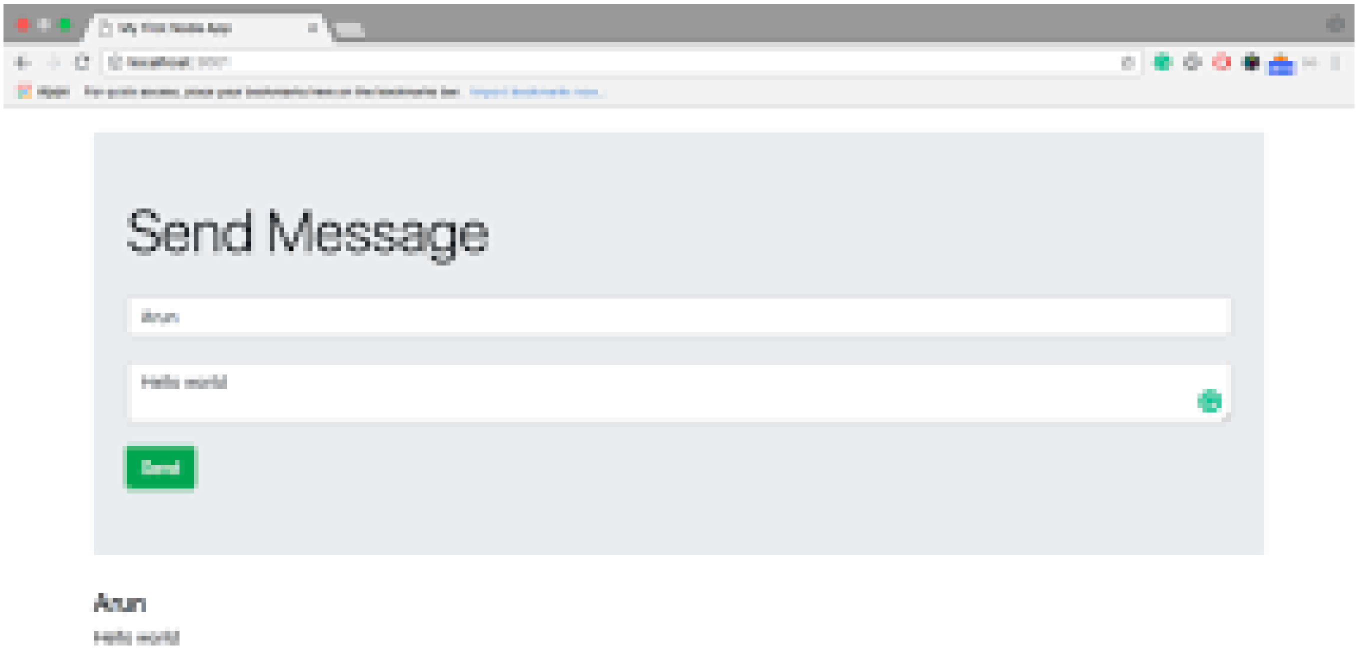
2. An increase in state rates among aid workers: Since AI has been enforced throughout the full system of aid on a grander scale, several activities that were historically performed by humans will be done by machines today. Chatbots and robots will give psychological state facilitate, analyse the condition of patient’s health, and foresee some issues like seizures, sepsis, pathology, etc.

3. Chance of a Defective Diagnosis: The correct diagnosing for a definite unwellness depends on numerous knowledge collected from several people that have intimate similar symptoms and conditions. to urge the suitable comparison, the AI info ought to contain decent info regarding the patients of the actual cluster. Therefore, if there's a scarcity of data a few persons from a definite background, AI will give AN inaccurate diagnosing.

4. Social Prejudices: AI-based machines cannot totally perceive attribute and also the background that creates them biased against the patients being diagnosed.

## 5. FIGURES.

amazon
slack
shopify
asana
certify
stripe
mongoDB.
zendesk
Magento
salesforce



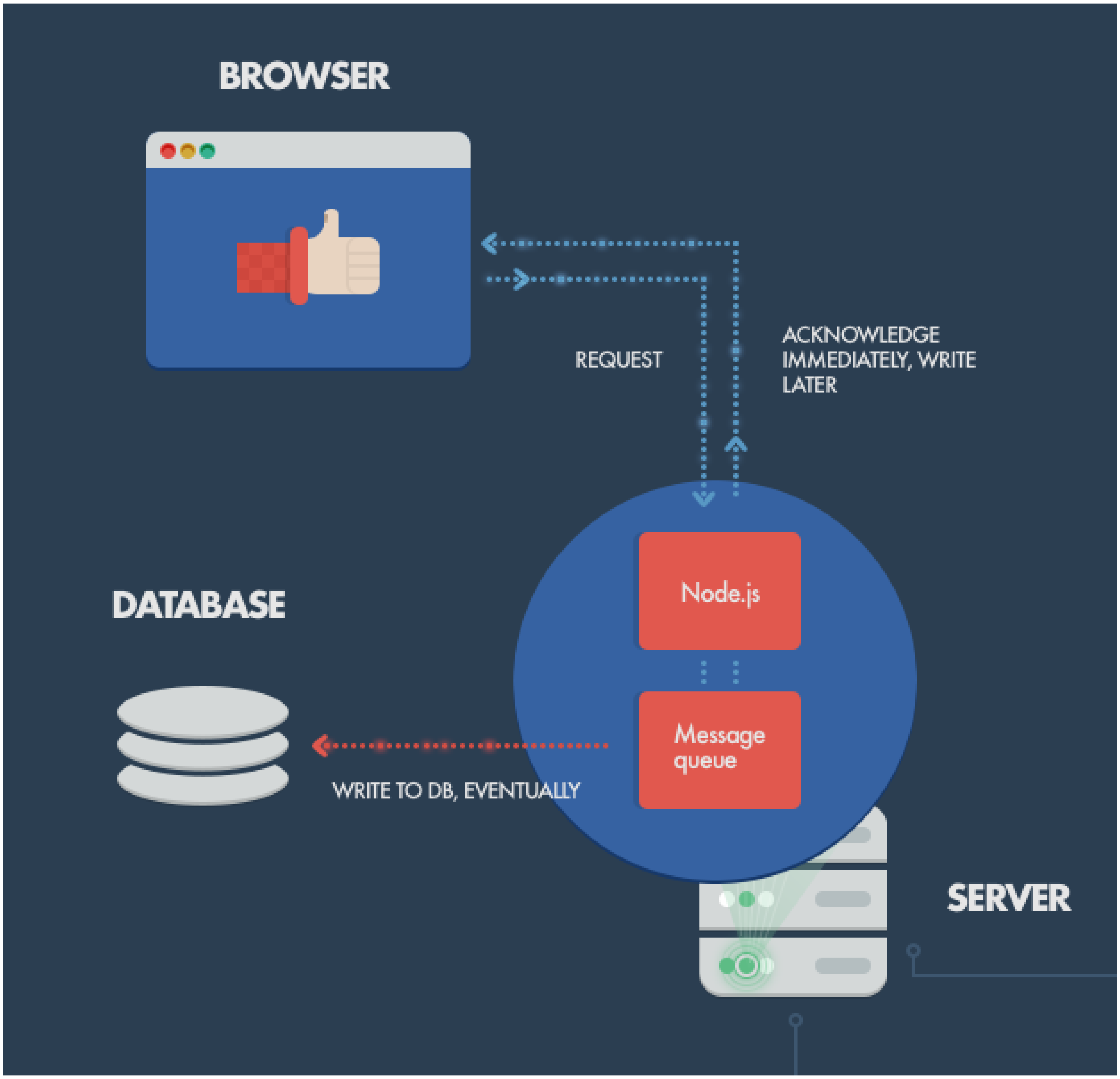
Send Message

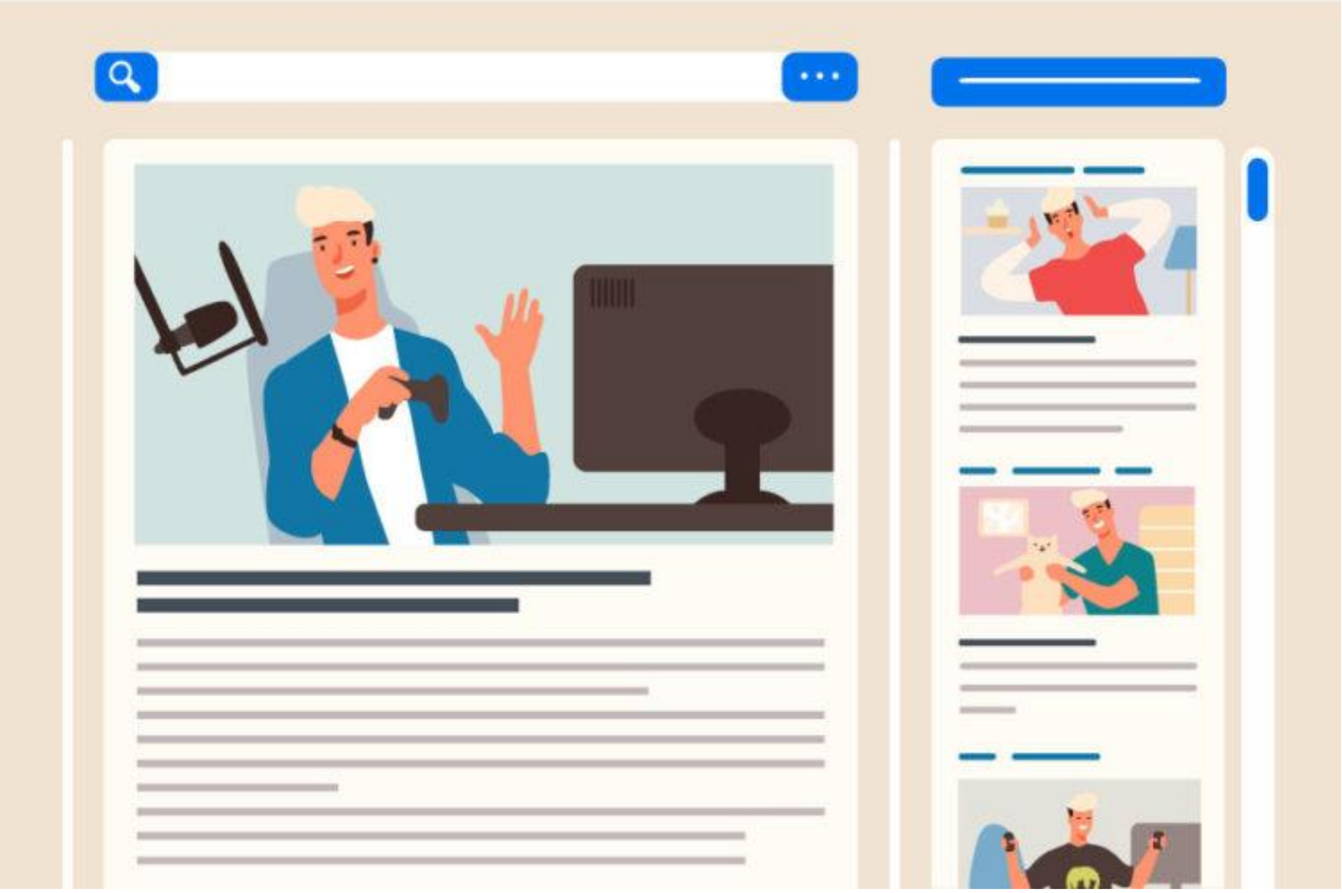
Arun

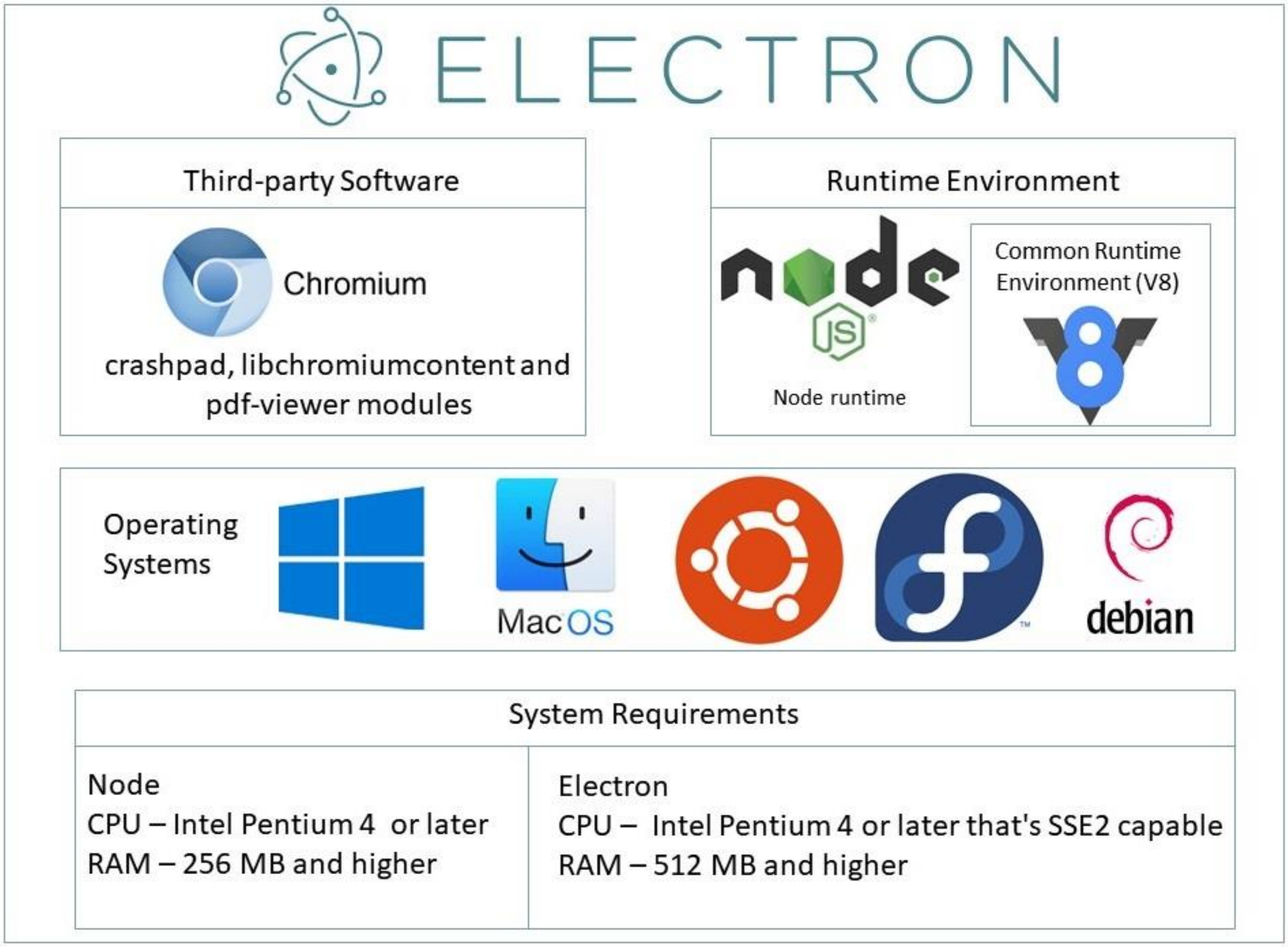
Hello world

Arun

Hello world


BROWSER
REQUEST
ACKNOWLEDGE IMMEDIATELY, WRITE LATER
DATABASE
Node.js
Message queue
WRITE TO DB, EVENTUALLY
SERVER





ELECTRON

Third-party Software

Chromium

crashpad, libchromiumcontent and pdf-viewer modules

Runtime Environment

node JS

Node runtime

Common Runtime Environment (V8)

Operating Systems

MacOS

debian

System Requirements

| Node | Electron |
|---|---|
| CPU – Intel Pentium 4 or later | CPU – Intel Pentium 4 or later that's SSE2 capable |
| RAM – 256 MB and higher | RAM – 512 MB and higher |

photo & video

Follow These 5 Instagram Accounts for a Daily Dose of Holiday Food Inspiration

by [illegible]

[illegible]

8 Thanksgiving Photography Tips For Capturing Authentic Holiday Moments

by [illegible]

[illegible]

The Best Wines to Pair with Thanksgiving Dinner

by [illegible]

[illegible]

Connect with us

Search

What are you looking for?

Popular Classes

The Complete Wedding Photography Experience

[illegible]

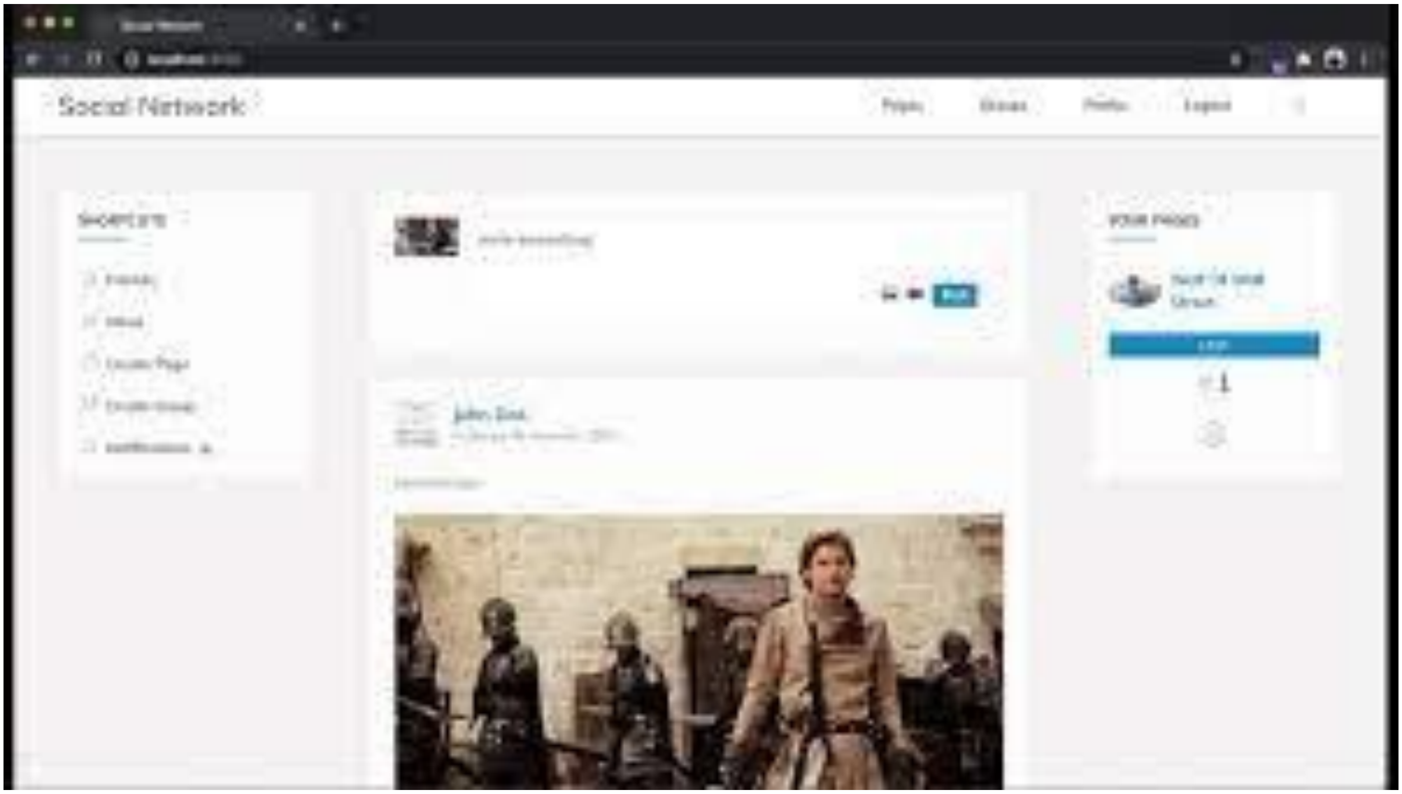
Portrait Photography Bootcamp

[illegible]

Photography for Beginners: Nikon Kit

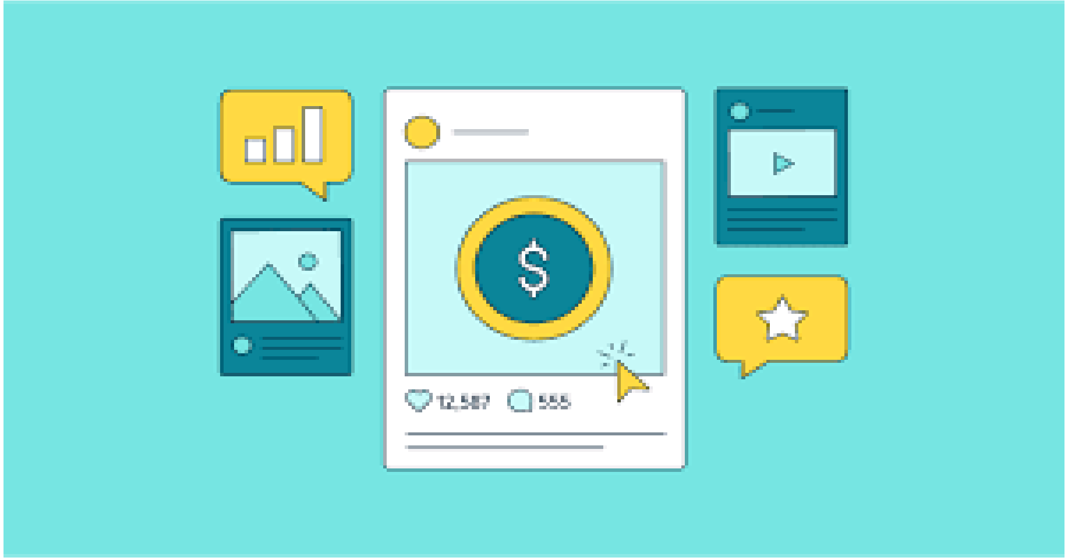
[illegible]

Popular Posts


localhost
Social Network
Pages
Groups
Profile
Logout
SHORTCUTS
Friends
Inbox
Create Page
Create Group
Notifications
YOUR PAGES
John Doe

# 6. SCOPE

The study got the finding about the implementation of Node.js. Below is discussed the implementations positive findings as a result of literature review and the survey. ⎫ The Node.js have made Full Stack Developers' job a dream come true. In absence of Node.js it was hard for a developer to learn several different languages and environments to manage the complete system at server side and client side. ⎫ Organizations and developers can now with the invent of Node.js build highly load bearable and faster applications and by using Single Page Applications (SPA) now the server calls are reduced and the applications are more user friendly and faster. ⎫ Node.js made it easy to achieve high load operations like graphic processing and Internet GIS very quicker, and it can be reliably used in every field where the files sizes are high or the network bandwidth is highly consumed. Node.js will make such operations faster and with less need of bandwidth. ⎫ Community like its feature that the same language is also being used at server side while JavaScript is always been used at client side for ages. Node.js have some challenges in context of its use in the community as well as its adoption by the developers and organizations over the existing programming technologies. No doubt that Node.js have great benefits, it have also some challenges to the community.



One such challenge is the ability of misuse of the widely used language by developers.

One enthusiast have made a backdoor software using Node.js on Raspberry Zero. It can create backdoors in the target computer and their network even if the computer is password locked. Although there are solutions from such backdoors but some seems impractical like totally blocking the USB ports, and closing the web browser every time the user leaves the computer. And other options are not implemented by majority and mostly might not be aware of it like using secure layer on ones websites (https), and enabling secure flags on the cookies which common users might not know about it [45]. ⎫ There is a plus point but as understood from the survey conducted that the community feel it hard to learn JavaScript for Node.js ⎫ Also the developers having knowledge of other programming languages have complications in adopting Node.js. Even the setting up of server for their programming work is not an obstacle. This is as concluded from the survey results. ⎫ Another plus points were event-Driven Programming, Non-Blocking I/O, and asynchronous feature. But according the survey results is that the features like event-driven programming, NonBlocking I/O, and Asynchronous processing is a hindranceAs a result of the survey, a challenge comes to front is that most organizations are not ready to adopt the new technologies like Node.js over their existing ones like PHP, .Net, etc. Also there is a lack of market awareness which is causing a barrier to adopt Node.js for implementation. At a developer level, there is a challenge which is seen from the survey results that they are not feeling it easy to learn the database working and using of the JavaScript environment. And there also seems a lack of enough knowledge among the developers as from the survey results a reasonable respondents to the questions on the important features like event-driven, non-blocking I/O and asynchronous processing is making the decision about this. The community can be taken to the Node.js by little efforts of trainings, and conducting workshops to

introduce Node.js to new and existing programmers and explain to them the benefits of Node.js features like event-driven programming, non-blocking I/O, and asynchronous processing. The community need to be updated about the features of how Node.js is faster in performing network related tasks specially when it come to the challenge of handling large files over the network and handling multiple callbacks in the other languages © 2017 Global Journals Inc. (US) Global Journal of Computer Science and Technology Volume XVII Issue II Version I 81Year 2017 ( ) E Node.js Challenges in Implementation like PHP, .NET, etc, causing the server to overload or require more memory. Also organization should be briefed about the benefits of hiring Full Stack Developers based on Node.js and how they will cut costs related to server bandwidths and developers hiring and speed of the applications built over Node.js References Références Referencias 1. "The benefits of web-based applications," [Online]. Available: http://www.magicwebsolutions.co.uk/blog /the-benefits-of-web-basedapplications.htm. [Accessed 25 November 2016]. 2. Web Application Basics, Pearson Higher Education. 3. F. Bridge, "What Types of Developers Are There?," tree house, 24 June 2016.

# 7. REFERENCE

1. "Benefits of web-based applications," [Online]. Available at: http://www.magicwebsolutions.co.uk/blog /the-benefits-of-web-basedapplications.htm. [Accessed 25 November 2016]. 2. Web Application Basics, Pearson Higher Education. 3. F. Bridge, "What Types of Engineers Are There ?," tree house, 24 June 2016. [Online]. Available: http://blog.teamtreehouse.com/what-types-of developer-are-there. [Accessed 25 November 2016]. 4. M. Wales, "Front-End vs Back-End vs Full Stack Developers," Udacity, 08 December 2014. [Online]. Available: http: //blog.udacity.com/2014/12/front-end vs-back-end-vs-full-stack-web-developers.html. [Accessed 25 November 2016]. 5. J. Long, "I Don't Speak Your Language: Frontend vs. Backend," tree house, 25 September 2012. [Online]. Available: http://blog.teamtreehouse.com/ dont-speak-your-language-frontend-vs-backend. [Accessed 25 November 2016]. 6. A. Mardan, "PHP vs. Node.js," Programming Weblog, [Online]. Available: http://webapplog.com/ php-vs-node-js /. [Accessed 28 January 2016]. 7. J. Kaplan-Moss, "Quora," [Online]. Available: https://www.quora.com/What-are-the-benefits-of developed-in-Node-js-versus-Python. [Accessed 29 June 2016]. 8. "Node.js Tutorial," learning point, [Online].

Available at: https://www.tutorialspoint.com/nodejs/index.htm. [Accessed 25 November 2016]. 9. N. Chhetri, "Comparative Analysis of Node.js (Server-Side JavaScript)," Climax Projects in Computer Science and Information Technology., P. 5, 2016. 10. RR McCune, "Node.js Paradigms and Bench marks," 2011. 11. Node.js, "Node.js Home Page," Joyent, 2016. [Online]. Available: https : // nodejs.org/zu/. [Accessed May 01, 2016]. 12. G. Developers, "Chrome V8 | Google Developers," Google, [Online]. Available: https: // developers. google.com/v8/. [Accessed 27 May 2016]. 13. event machine, "GitHub, Inc.," [Online]. Available: https://github.com/eventmachine/eventmachine. atAc -cessed 30 June 2016]. 14. Twisted Matrix Labs, "Twisted Matrix Labs," [Online]. Available: http://twistedmatrix.com/trac/. [Accessed 30 June 2016]. 15. Apache Software Foundation, "Apache MINA," [Online]. Available: http://mina.apache.org/. [Accessed June 30, 2016]. 16. Apache Software Foundation, "Apache MINA," [Online]. Available: http: // mina.apache.org/ async web-project / index.html. [Accessed 30 June 2016]. 17. Ryan Dahl: Original Node.js presentation. [Film]. Youtube, 2012. 18. "How Uploading Time Affects Your Bottom Line," Kissmetrics Blog, [Online]. Available: https: // blog. kissmetrics.com/loading-time/. [Accessed 25 November 2016]. 19. C. Buckler, "Site Point Smack Down: PHP vs Node.js," Site Point, [Online]. Available: http: // www. sitepoint.com/sitepoint-smackdown-php-vs-node js /. [Accessed 28 January 2016]. 20. Firehose, "Firehose," Firehose Project, [Online]. Available: http: //blog.thefirehoseproject.com/posts/n odejs-vs-rails /. [Accessed 30 June 2016]. 21. "Quora," [Online]. Available: https: //www.quora. com / What-evil-use-Node-js. [Accessed 30 June 2016]. 22. "Hashnode," Hashnode, [Online]. Available: https://hashnode.com/post/what-are-the-actual-dis benefits-of-using-nodejs-ciibz8fd3017yj3xtxqz 1r9 hs. [Accessed 30 June 2016] 23. Flickr, "Flickr, Yahoo Company," Yahoo, [Online]. Available: https://www.flickr.com. [Accessed 18 November 2016]. 24. A Harris, "The Birth of a Node: Where Did It Come From? Creator Ryan Dahl Share History," Silicon ANGLE, 01 April 2016. [Online]. Available: http://siliconangle.com/blog/2013/04/01/the-birth-of node-where-did-it-come-from-creator-ryan-dahl shar es-the-history /. [Accessed 18 November 2016]. 25. L. ORSINI, "What You Should Know About Node.js," read the article, 07 November 2013. [Online]. Available: http://readwrite.com/2013/11/07/what you-need-to-know-about-nodejs /. [Accessed 2016 November 2016]. 26. G. Nemeth, "Rising Stack Engineering," [Online]. Available: https://blog.risingstack.com/history-of node-js /. [Accessed 27 October 2016]. 27. R. Dahl, "Porting Node to Windows With Microsoft's Help," Node.js, 23 June 2011. [Online]. Available: https://nodejs.org/en/blog/uncategorized/porting node-to-windows-with-microsofts-help /. [Accessed 18 November 2016]. 28. P. Krill, "Q&A: Why io.js decided to launch Node.js," JAVAWORLD, 04 December 2014. [Online]. Available at: http: //www.javaworld.com/article/2855639/open- source-tools / egg-why-io-js-decided-to-fork node-js.html. [Accessed 18 November 2016]. 29. "Node.js Foundation Advances Community Collaboration, Announcing New Members and Certified Technology Management," Linux Foundation, 15