Web Application to upload and Download Files

Submitted by

Team :- Geek Mode!

Bedanta Gautom (RA2011033010048)

Karan Keshri (RA2011033010044)

Raghav upadhay (RA2011033010016)

Shubham Kumar (RA2011033010015)

Under the Guidance of

Dr. B.Hariharan

Assistant Professor , Department of Computational Intelligence

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with Specialization in Software Engineering

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203

November 2022

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this project report titled **“Web Application" is** the bonafide work done by Karan Keshri (044), Bedanta Gautom (048), Raghav upadhay(016), Shubham Kumar (015) who carried out the Project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Dr. B.Hariharan

Computer Communication – Course Faculty

Assistant Professor ,
Department of Computational Intelligence

TABLE OF CONTENTS

1. ABSTRACT

The primary purpose of an FTP server is to allow users to upload and download files. An FTP server is a computer that has a file transfer protocol (FTP) address and is dedicated to receiving an FTP connection. FTP is a protocol used to transfer files via the internet between a server (sender) and a client (receiver). An FTP server is a computer that offers files available for download via an FTP protocol, and it is a common solution used to facilitate remote data sharing between computer.

Many people have used FTP before without even realizing it. If you have ever downloaded a file from a web page, you've used FTP. The first step is to log in, which may occur automatically or by manually inputting a username and password. FTP will also require you to access an FTP server through a specific port number. Once you access the FTP server through your FTP client, you can now transfer files. Not all public FTP servers require you to sign in because some servers enable you to access them anonymously.

Meanwhile, password-protected FTP uses a username and password to access the files. FTP secure (FTPS) offers increased security when transferring, allowing for implicit transport layer security (TLS). FTP can also employ explicit TLS, which upgrades the connection to an encrypted connection for added security.

FTP software is relatively straightforward to set up. FileZilla is a free, downloadable FTP client. Other examples of FTP clients include Transmit, WinSCP, and WS_FTP. You type in the address of the server you wish to access, the port, and the password for accessing the server. Once access has been granted, the user's files on their local system as well as the accessed server will be visible.The user can download files from the server to the local system, or upload files from the local system to the server. They can also make changes to files on the server, as long as they have the proper authorization to do so.

3. INTRODUCTION

An FTP server is an important component in FTP architecture and helps in exchanging files over the internet. The files are generally uploaded to the server from a personal computer or other removable hard drives (such as a USB flash drive) and then sent from the server to a remote client via the FTP protocol.

An FTP server needs a TCP/IP network to function and is dependent on the use of dedicated servers with one or more FTP clients. In order to ensure that connections can be established at all times from the clients, an FTP server is usually switched on; up and running 24/7.

An FTP server is also known as an FTP site or FTP host.

Although the FTP server actually sends files over the internet, it generally acts as the midpoint between the real sender of a file and its recipient. The recipient must access the server address, which can either be a URL FTP servers usually listen for client connections on port 21 since the FTP protocol generally uses this port as its principle route of communication. FTP runs on two different Transmission Control Protocol ports: 20 and 21. FTP ports 20 and 21 must both be open on the network for successful file transfers.

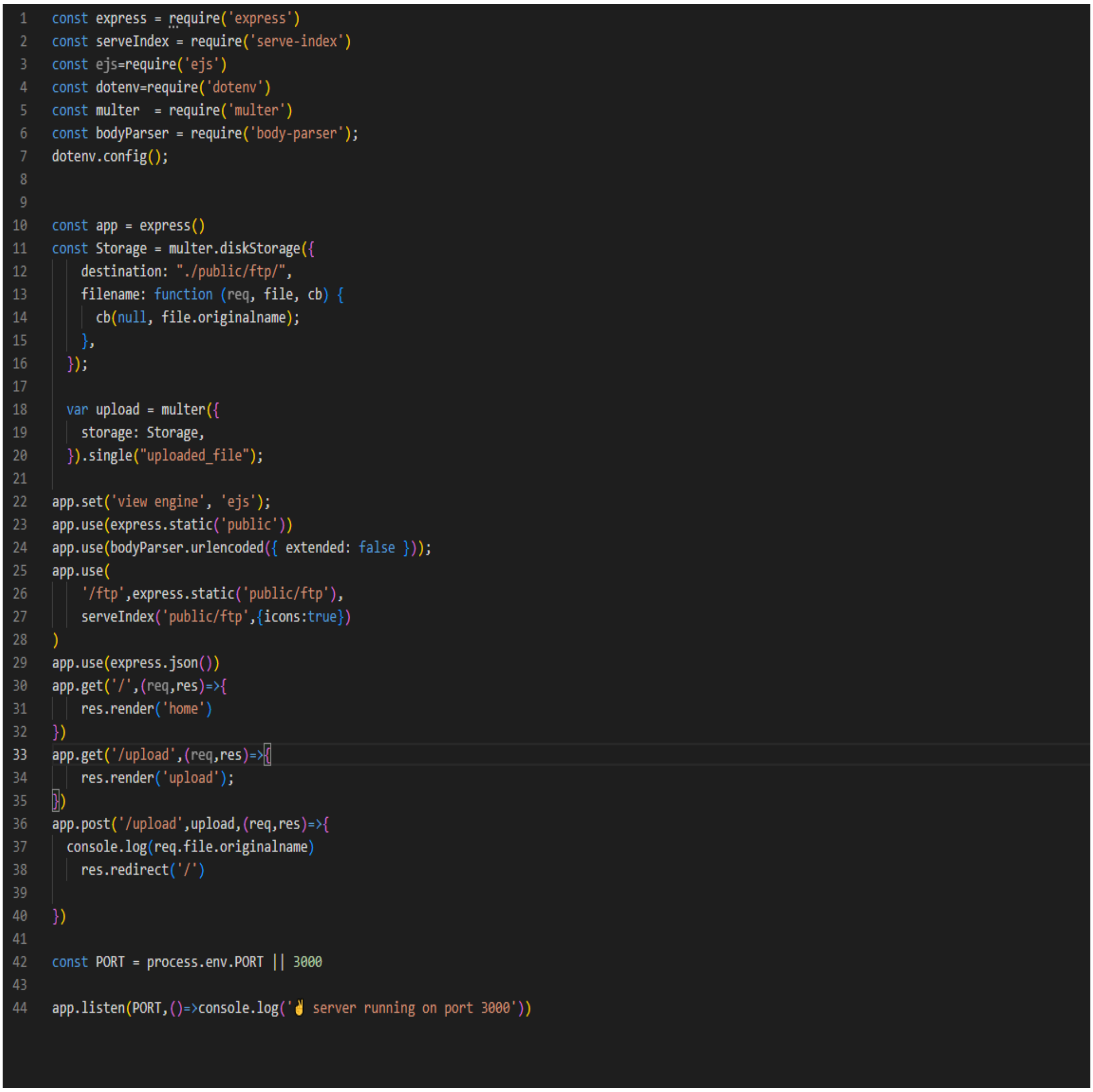
Role of the FTP Server

The FTP server allows the downloading and uploading of files. The FTP server's administrator can restrict access for downloading different files and from different folders residing in the FTP server. Files residing in FTP servers can be retrieved by common web browsers, but they may not support protocol extensions like FTPS. With an FTP connection, it is possible to resume an interrupted download that was not successfully completed; in other words, checkpoint restart support is provided.

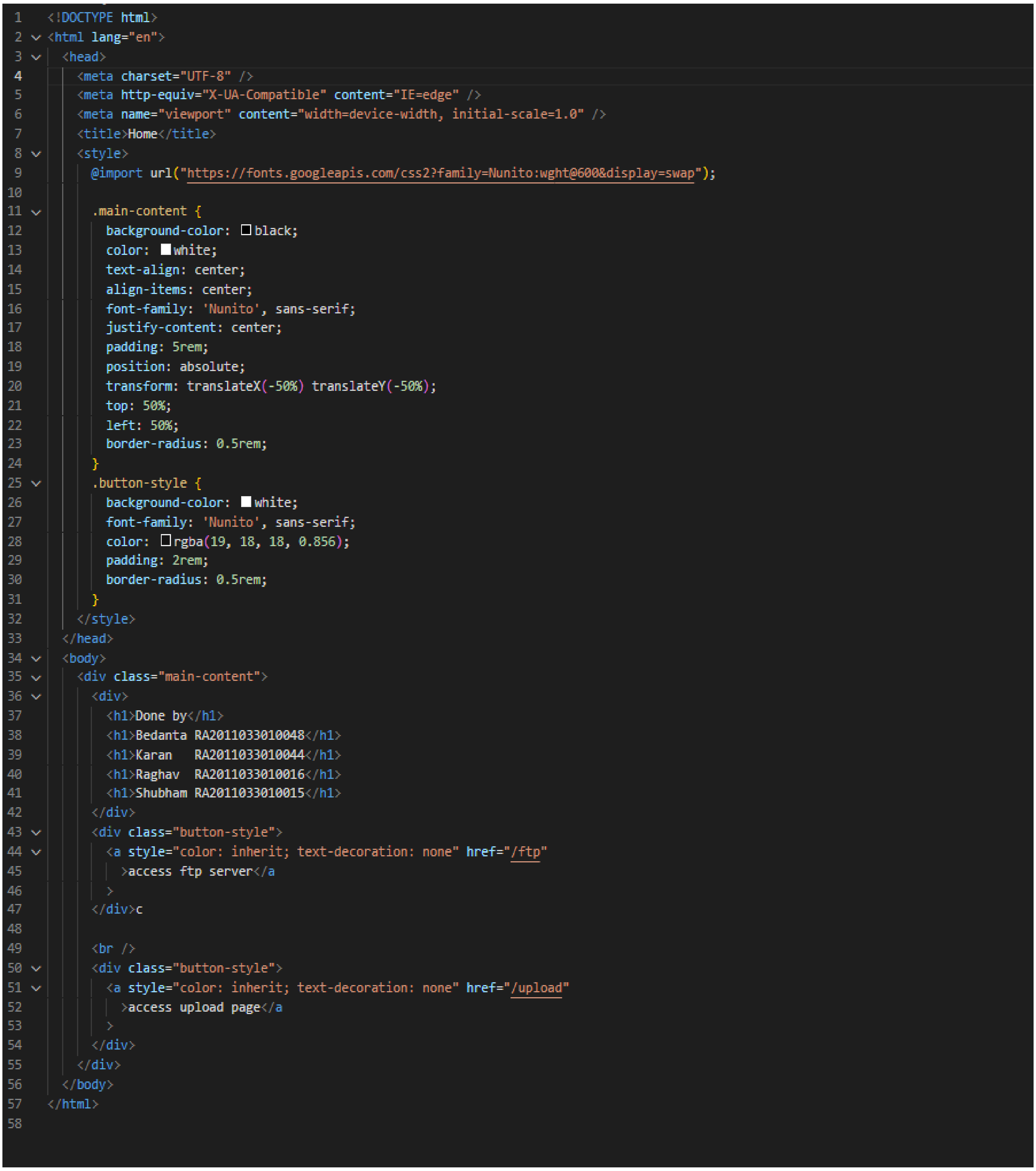
For the client to establish a connection to the FTP server, the username and password are sent using USER and PASS commands. Once accepted by the FTP server, an acknowledgment is sent to the client and the session can start. Failure to open both ports 20 & 21 prevents the full back-and-forth transfer from being made.

The FTP server can provide connection to users without login credentials. however, the FTP server can authorize these to have only limited access. FTP servers can also provide anonymous access. This access allows users to download files from the servers anonymously but prohibits uploading files to FTP servers.Beyond routine file transfer operations, FTP servers are also used for offsite backup of critical data. FTP servers are quite inexpensive solutions for both data transfer and backup operations, especially if security is not a concern. However, when simple login and authentication features are not sufficient to guarantee an adequate degree of security (such as when transferring sensitive or confidential information), two secure file transfer protocol alternatives, SFTP and FTP/S, are also available. These secure FTP server options offer additional levels of security such as data encryption.

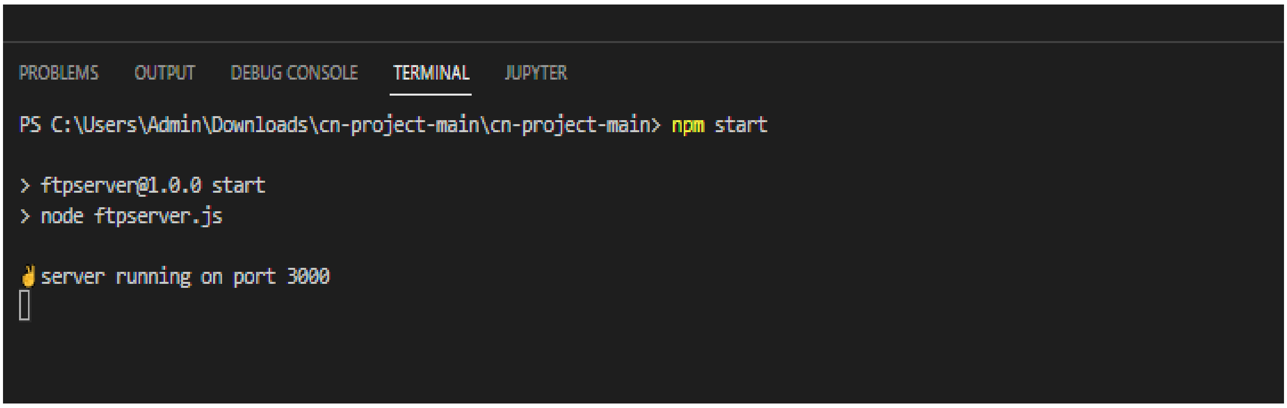
Code



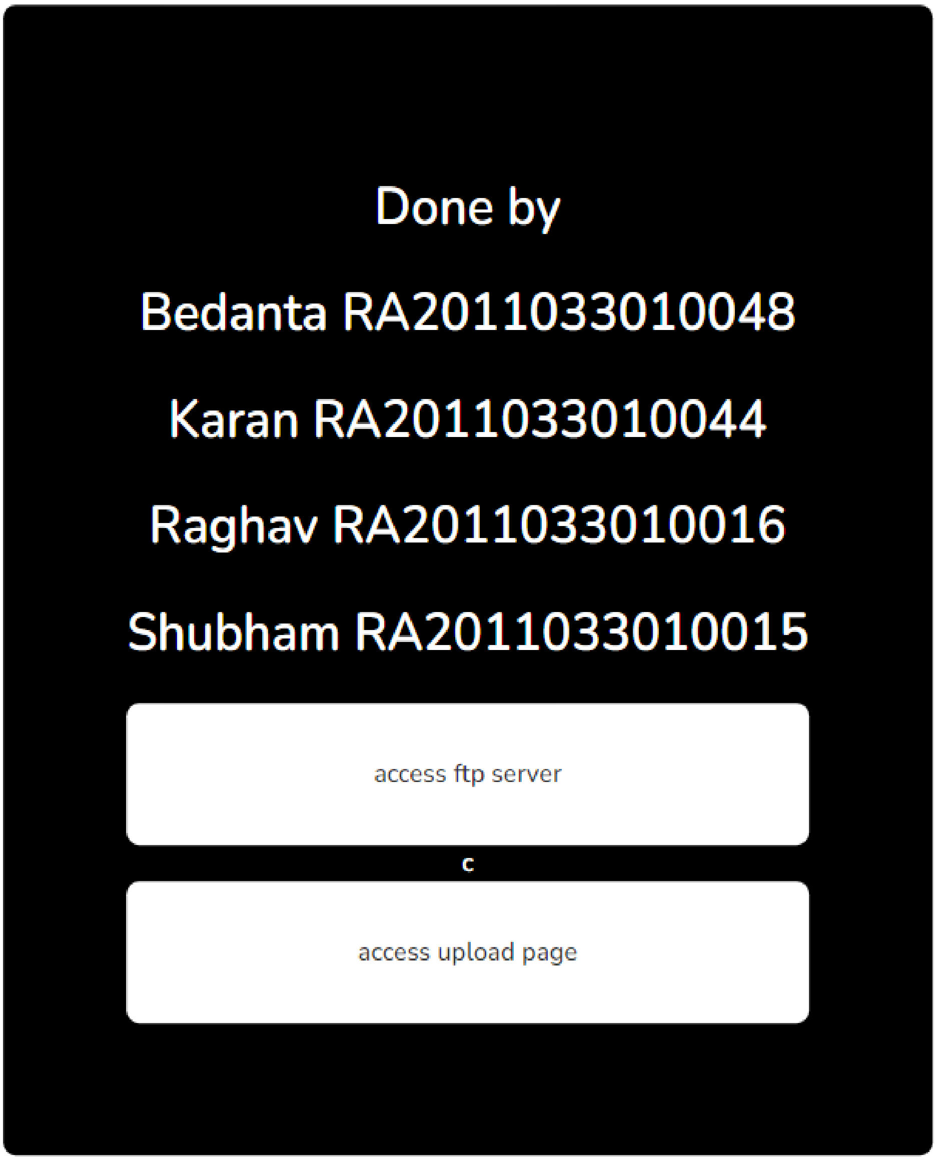
```
const express = require('express')
const serveIndex = require('serve-index')
const ejs=require('ejs')
const dotenv=require('dotenv')
const multer  = require('multer')
const bodyParser = require('body-parser');
dotenv.config();


const app = express()
const Storage = multer.diskStorage({
    destination: "./public/ftp/",
    filename: function (req, file, cb) {
      cb(null, file.originalname);
    },
  });

  var upload = multer({
    storage: Storage,
  }).single("uploaded_file");

app.set('view engine', 'ejs');
app.use(express.static('public'))
app.use(bodyParser.urlencoded({ extended: false }));
app.use(
    '/ftp',express.static('public/ftp'),
    serveIndex('public/ftp',{icons:true})
)
app.use(express.json())
app.get('/',(req,res)=>{
    res.render('home')
})
app.get('/upload',(req,res)=>{
    res.render('upload');
})
app.post('/upload',upload,(req,res)=>{
  console.log(req.file.originalname)
    res.redirect('/')

})

const PORT = process.env.PORT || 3000

app.listen(PORT,()=>console.log('✌ server running on port 3000'))
```



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Home</title>
    <style>
      @import url("https://fonts.googleapis.com/css2?family=Nunito:wght@600&display=swap");

      .main-content {
        background-color: black;
        color: white;
        text-align: center;
        align-items: center;
        font-family: 'Nunito', sans-serif;
        justify-content: center;
        padding: 5rem;
        position: absolute;
        transform: translateX(-50%) translateY(-50%);
        top: 50%;
        left: 50%;
        border-radius: 0.5rem;
      }
      .button-style {
        background-color: white;
        font-family: 'Nunito', sans-serif;
        color: rgba(19, 18, 18, 0.856);
        padding: 2rem;
        border-radius: 0.5rem;
      }
    </style>
  </head>
  <body>
    <div class="main-content">
      <div>
        <h1>Done by</h1>
        <h1>Bedanta RA2011033010048</h1>
        <h1>Karan   RA2011033010044</h1>
        <h1>Raghav  RA2011033010016</h1>
        <h1>Shubham RA2011033010015</h1>
      </div>
      <div class="button-style">
        <a style="color: inherit; text-decoration: none" href="/ftp"
          >access ftp server</a
        >
      </div>c

      <br />
      <div class="button-style">
        <a style="color: inherit; text-decoration: none" href="/upload"
          >access upload page</a
        >
      </div>
    </div>
  </body>
</html>

```



```
PS C:\Users\Admin\Downloads\cn-project-main\cn-project-main> npm start

> ftpserver@1.0.0 start
> node ftpserver.js

✌server running on port 3000
```



Done by

Bedanta RA2011033010048

Karan RA2011033010044

Raghav RA2011033010016

Shubham RA2011033010015

PROJECT STATEMENT:

The aim of the project is to create a web application which enables us to upload and download files. It is created with the help of JavaScript, Node JS, Express framework and Multer. It is mainly based on FTP(File Transfer Protocol).

FTP made handling data across the Internet much easier and intuitive. Without FTP and its later iterations, we would not be able to easily stream video content, use video calls, play online games, share files, or enjoy cloud storage.Today, FTP operates behind the scenes as a backbone for data transfer from servers around the world to millions of clients every second of every day.

IMPLEMENTATION

Implementing specific features depends on the pre defined features that are essential to incorporate in the web application. For example, intelligent forms or real-time financial transactions require a wholesome code, back-end deployments, and testing. With agile practices and the update of the conventional waterfall approach with 'sprint' environments, implementation professionals focus on Test-Driven Development and Implementation.

The web app implementation strategy benefits visionaries and business owners who would like to reimagine the web presence and increase outreach to the next level. It helps streamline online operations and reassures online presence. Development experts can ensure a successful implementation – offering organizations the potential to scale up with greater efficiency and accelerate long-term growth.

Agile methodologies help incur latest changes in the code and scale with business demandsInclusive approach to iterate, deploy and monitor for KPIs such as speed and responsivenessA well-defined layout that can go beyond legendary web apps to mobile screen resolutionsCost-effectiveness in terms of implementation strategies towards crucial monetary advantageAscertain app component behavior with UI implementation of control buttons and toolbar

_4. LITERATURE REVIEW

- **FTP SERVER**

The File Transfer Protocol is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture using separate control and data connections between the client and the server.

- **MULTER**

Multer is a node.js middleware for handling multipart/form-data which is primarily used for uploading files. It is written on top of a busboy for maximum efficiency.Multer adds a body object and a file or files object to the request object. The body object contains the values of the text fields of the form, the file or files object contains the files uploaded via the form

- **CONSTANT EXPRESS**

Constants are block-scoped, much like variables declared using the let keyword. The value of a constant can't be changed through reassignment (i.e. by using the assignment operator), and it can't be redeclared

- **DOTNEV**

Dotenv is a zero-dependency module that loads environment variables from a .env file into Process.nv Storing configuration in the environment separate from code is based on the twelve factor methodology

- **APP.SET**

The app.set() function is used to assigns the setting name to value. You may store any value that you want, but certain names can be used to configure the behavior of the server.

- **APP.USE**

The app. use() method mounts or puts the specified middleware functions at the specified path. This middleware function will be executed only when the base of the requested path matches the defined path.

- **APP.GET**

The **app.get()** function routes the HTTP GET Requests to the path which is being specified with the specified callback functions. Basically it is intended for binding the middleware to your application.

Syntax:

```
app.get( path, callback )
```

- **APP.POST**

The app.post() method routes all the HTTP POST requests to the specified path with the specified callback functions.

Syntax

```
app.path(path, callback, [callback]
```

➢ WEB Server

One of the widely used servers in today’s market is a web server. A web server is a special kind of application server that hosts programs and data requested by users across the Internet or an intranet. Web servers respond to requests from browsers running on client computers for web pages, or other web-based services.

9. REFERENCES

Books:-

- Data and Computer Network by William Stallings • Computer Network and technologies by Bill Hancock

Websites:-

- www.geeksforgeeks.org
- www.ibm.com
- www.javatpoint.com

-------------------------------------The End-----------------------------------