

Express JS

Miscellaneous Tasks

1. Write express js script to perform tasks as asked below. (Get Method)

1. Create one HTML file which contains two number type input fields, one dropdown which contains options like (select, addition, subtraction, multiplication, division) and one submit button.
2. The input fields must contain the value greater than 0 else it will give a message "Please enter the valid message". Also, user must select any of the formula from the dropdown else give a message "You have not selected any formula". (Message will be displayed on "/calc" page.)
3. If one formula is selected and numbers are entered then respective calculations will be performed on the page "/calc".
4. Use get method to request data.

/src/calc.js

```
var expr = require("express");
var app = expr();
app.use(expr.static("../css"));
app.use(expr.static("../html",{index:'1.html'}));

app.get("/calc",(req,res,next)=>{
  res.set("content-type","text/html");
  var n1 = parseInt(req.query.n1);
  var n2 = parseInt(req.query.n2);
  if ((n1>0) && (n2>0)) {
    if(req.query.formula == "addition"){
      a = n1+n2;
      res.write("<h1>Addition is : " + a + "</h1>");
    }
    else if(req.query.formula == "subtraction"){
      s = n1-n2;
      res.write("<h1>Subtraction is : " + s + "</h1>");
    }
    else if(req.query.formula == "multi"){
      m = n1*n2;
      res.write("<h1>Multiplication is : " + m + "</h1>");
    }
    else if(req.query.formula == "div"){
      d = n1/n2;
      res.write("<h1>Division is : " + d + "</h1>");
    }
    else{
      res.write("<h1>You have not selected any formula.</h1>");
    }
  }else{
    res.write("<h1>Please enter the valid number/s.</h1>");
  }
});
```

```
}  
  res.send()  
})  
app.listen(6012);  
/css/t1.css
```

```
.lj_form{  
  background-color: rgb(158, 158, 219);  
}
```

/html/1.html

```
<html><head>  
  <link rel="stylesheet" href="/t1.css">  
</head>  
<body>  
  <form action="/calc" method="get" class="lj_form">  
    <fieldset>  
      <legend>Calculator:</legend>  
      <label for="n1">Enter Number1:</label>  
      <input type="number" name="n1"><br><br>  
  
      <label for="n2">Enter Number2:</label>  
      <input type="number" name="n2"><br><br>  
  
      <label for="formula">Formula: </label>  
      <select name="formula" id="formula">  
        <option value="">Select formula</option>  
        <option value="addition">Addition</option>  
        <option value="subtraction">Subtraction</option>  
        <option value="multi">Multiplication</option>  
        <option value="div">Division</option>  
      </select><br><br>  
  
      <input type="submit" value="Submit">  
    </fieldset>  
  </form>  
</body></html>
```

Output:



2. Write express script to get form data using post method and display First name in Red color, Password in Green Color and Gender in Black color.

trail.html (file in public folder)

```
<form method="post" action="/process-post">
<fieldset>
  <legend style="color:blue;text-align:center" >Log In </legend>
  </br>
  <label> User Name</label>
    <input type="text" name="firstname"/>
  </br>
  <label> Password</label>
    <input type="password" name="password"/>
  </br>
  <label> Gender</label>
    <input type="radio" name="r1" value="Male">Male</input>
    <input type="radio" name="r1" value="Female">Female</input>
  </br>
  <label> Comment</label>
    <textarea rows="5" cols="15" name="comment"> </textarea>
  </br>
  <input type="Submit">
  <input type="Reset">
</fieldset>
</form>
```

trial.js (file in src folder)

```
const expr=require("express")
const app=expr();

app.use(expr.urlencoded({extended:false}))

app.use(expr.static('../public',{index:"trial.html"}))
app.post("/process-post",(req,res)=> {
  res.write(`<h1 style="color:red">Firstname=${req.body.firstname}</h1>
    <h1 style="color:green">Password=${req.body.password}</h1>
    <h1>Gender=${req.body.r1}</h1>`)
  res.end()
})
app.listen(5000)
console.log("Server is running")
```

Output:

After filling form on localhost:5000 with name:"abc", password: "abcd", gender: Male

Output on: <http://localhost:5000/process-post>

Firstname=abc

Password=abcd

Gender=Male

3. Write express js script to perform the tasks as asked below. (Post method)

1) Create one HTML file named `ljform.html` and add one form which contains username, password and submit button. Data should be submitted by HTTP post method.

2) Submit button is of black color with white text. (External CSS)

3) On home page form should be displayed and while submitting the form, on next page named `/login` if username is admin then it will display "Welcome admin" else display "Please login with Admin name".

In 3.js file in src folder

```
var expr = require("express");
var app = expr();
app.use(expr.urlencoded());
app.use(expr.static('../public', {index: 'ljform.html'}));
app.post("/login", (req, res) => {
  if (req.body.uname == 'admin') {
    res.write(`<h1 style="color:green">Hey ${req.body.uname}, Welcome!</h1><br>`);
  }
  else {
    res.write(`<h1 style="color:red">Please login with Admin name</h1>`);
  }
  res.send();
})
app.listen(5222);
```

In 1.css file in public folder

```
.click{
  background-color: black;
  color: white
}
```

In login.html file in public folder

```

<html>
  <head>
    <link rel="stylesheet" href="1.css">
  </head>
  <body>
    <form method="post" action="/login">
      <label>User name:</label>
      <input type="text" name="uname" ></input>
      <label>Password:</label>
      <input type="text" name="password" ></input>
      <button name="Submit" value="Submit" class="click">Submit</button>
    </form>
  </body>
</html>

```

4. Write express JS script to load in html file having username, password and submit button. On clicking of submit button, it should jump on check page using post method. If username is “admin” then jump on next middleware to print “welcome admin”. For any other username it should stay on same middleware to print warning message in red color.

two.html (file in public folder)

```

<form action="/check" method="post">
  <fieldset>
    <legend>Sign-up</legend>
    Username: <input type="text" name="uname"/>
  </br></br>
    password: <input type="text" name="pwd"/>
  <br></br>
  <input type="submit"/>
</fieldset>
</form>

```

task.js (file in src folder)

```

const expr=require("express");
const app=expr();

```

```

app.use(expr.urlencoded({extended:false}))
app.use(expr.static('../public',{index:"two.html"}));

app.post("/check",(req,res,next)=>{
  const u1=req.body.uname
  if(u1=="admin")  {
    next();
  }
  else  {
    res.write(`<h1 style="color:red"> You are not admin </h1>`);
    res.send();
  } } );
app.post("/check",(req,res)=>{
  res.write("<h1>welcome admin</h1>");
  res.send()  });
app.listen(7002)

```

5. Write express js script to perform following tasks. (Middleware)

Create one html file which contains one text field for name, email field and checkbox for subscription. Html file will be loaded on home page. Email and name fields are required fields.

On login page welcome user and email id data should be printed.

- If user checked the subscription then “Thank you for the subscription” message will be printed and “logout” link will be displayed under the message. If user clicks logout link then he/she will be redirected to the home page.
- If user has not opted for the subscription then “You can subscribe to get daily updates” message will be printed and “subscribe” link will be displayed under the message.
- If user clicks subscribe link then he/she will be redirected to the subscription page. In this page “Thank you for the subscription” message will be printed and “logout” link will be displayed under the message. If user clicks logout link then he/she will be redirected to the home page.

Use concept of the middleware and you can use any of http methods(get/post).

/src/app1.js

```

var expr = require("express");
var app = expr();
app.use(expr.static('../public',{index:'trial1.html'}));
app.get("/login",(req,res,next)=>{
  console.log(req.query);
  res.set("content-type","text/html");
  res.write("<center><h1>Welcome " + req.query.name + "</h1>");
  res.write("<center><h2>Your email id is " + req.query.email + "</h2>");

```

```

    next();
  })
  app.get("/login",(req,res,next)=>{
    if(req.query.newsletter && req.query.newsletter == "on"){
      res.write("<h3>Thank you for your subcsription</h3><a href='/>Logout</a>");
    }else{
      res.write("<h3>You can subscribe to get daily updates</h3><a
href='/subscribe'>Subscribe</a></center>");
    }
    next();
  });
  app.get("/subscribe",(req,res)=>{
    res.set("content-type","text/html");
    res.write("<h3>Thank you for your subcsription</h3></center><a href='/>Logout</a>");
    res.send();
  });
  app.listen(5001);

```

/public/trial1.html

```

<form action="/login" method="get">
  <label for="name">Name:</label>
  <input type="text" name="name" required><br><br>

  <label for="email">Email:</label>
  <input type="email" name="email" required><br><br>

  <label for="newsletter">Subscribe?</label>
  <input type="checkbox" name="newsletter" id="newsletter"><br><br>

  <input type="submit" value="Submit">
</form>

```

← → ↻ ⓘ localhost:5001

Welcome lju

Your email id is lju@gmail.com

Thank you for your subscription

☐ **Subscribe?**

[Logout](#)

Welcome lju

Your email id is lju@gmail.com

You can subscribe to get daily updates

[Subscribe](#)

6. You have been assigned to develop a user feedback form for a website using Express.js and cookies.

Implement the following requirements:

- Create a form with the following fields:
 - Name (input field)
 - Email (input field)
 - Message (textarea field)
 - Rating (radio buttons: Bad, Average, Good, Very Good, Excellent)
- When the user submits the form, store their feedback information (name, email, message, and rating) in a cookie named "feedback" that expires in 10 seconds.
- Display a confirmation message to the user after successfully submitting the form & Create a link to display the feedback details stored in the "feedback" cookie.
- When the user click to the link, retrieve the feedback information from the cookie and display it on the page also include a link on the feedback details page to Logout.
- When the user clicks the link, user redirected to home page.

In cookietask2.html file:

```
<html>
<head>
  <title>User Feedback Form</title>
</head>
<body>
  <h1>User Feedback Form</h1>
  <form action="/submit-feedback" method="POST">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>

    <label for="message">Message:</label>
    <textarea id="message" name="message" required></textarea><br><br>

    <label for="rating">Rating:</label>
    <input type="radio" id="rating1" name="rating" value="Bad" required>
    <label for="rating1">Bad</label>
    <input type="radio" id="rating2" name="rating" value="Average" required>
    <label for="rating2">Average</label>
    <input type="radio" id="rating3" name="rating" value="Good" required>
    <label for="rating3">Good</label>
    <input type="radio" id="rating4" name="rating" value="Very Good" required>
    <label for="rating4">Very Good</label>
    <input type="radio" id="rating5" name="rating" value="Excellent" required>
```



```

<label for="rating5">Excellent</label><br><br>

<input type="submit" value="Submit">
</form>
</body>
</html>

```

In task_cookie2.js file:

```

const express = require('express');
const cp = require('cookie-parser');
const app = express();
app.use(cp());
app.use(express.urlencoded({ extended: true }));

app.get('/', (req, res) => {
  res.sendFile(__dirname + '/cookietask2.html');
});

app.post('/submit-feedback', (req, res) => {
  const { name, email, message, rating } = req.body;

  // Create the feedback object
  const feedback = {
    name,
    email,
    message,
    rating
  };

  // Set the feedback cookie with a 10-second expiration
  res.cookie('feedback', feedback, { maxAge: 10000 });

  res.send("Thank you for your feedback! <br> <a href=\"/feedback-details\"> Show Feedback</a>");
});

app.get('/feedback-details', (req, res) => {
  const feedback = req.cookies.feedback;

  if (feedback) {
    res.send(`
      <h1>Feedback Details</h1>
      <p><strong>Name:</strong> ${feedback.name}</p>
      <p><strong>Email:</strong> ${feedback.email}</p>
      <p><strong>Message:</strong> ${feedback.message}</p>
    `);
  }
});

```

```

    <p><strong>Rating:</strong> ${feedback.rating}</p>
    <a href="/" > logout </a>`);
  }
else {
  res.send('No feedback available.');
```

```

  }
});

app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

7. write a script to meet following requirements (Authentication)

- 1)create index.html file page and open it on localhost.
- 2)after clicking submit button, it should jump on “savesession” page. Store username in session.
- 3)After saving session, redirect to “fetchsession” page and read value. Put a LOGOUT link button here.
- 4)Jump on delete session page after clicking LOGOUT button.
- 5)Destroy the session on this page and redirect to index.html page.

index.html

```

<html>
  <body>  <center>
    <form method="get" action="/savesession">
      <div><input type = "text" name="uname" placeholder="Username" ></input></div>
      <div><input type = "password" name="pass" placeholder="Password"></input></div>
      <input class="lj_in" type="submit" name="submit" value="Submit">
    </form>
  </center>  </body> </html>
```

Session.js

```

var expr = require("express");
var app = expr();
var es = require("express-session");
app.use(es({
  resave:false,
  saveUninitialized:false,
  secret:"ljul1"
}));
app.use(expr.static("../public"))
app.get("/savesession",(req,res)=>{
  var uname=req.query.uname;
  var pass=req.query.pass;
```

```

req.session.uname = uname;
req.session.password = pass;
res.redirect("/fetchsession")

})
app.get("/fetchsession",(req,res)=>{
  if(req.session.uname == "admin" && req.session.password=="admin@123"){
    res.write("<h1 style='color:green;'>Welcome Admin</h1>")
    res.write("<a href='/deletesession'>Delete Session</a>")
  }else{
    res.write("<h1 style='color:red;'>Please enter valid username or password</h1><a href='/>Login</a>")
  }
  res.send();
});
app.get("/deletesession",(req,res)=>{
  req.session.destroy();
  res.write("<h1>Session Destroyed</h1><a href='/>Login</a>")
  res.send();
});
app.listen(6177);

```

8. Write a script to meet following requirements:

- Create session.html file page which contains form (username,password,login button). and open it on localhost.
- After clicking submit button, it should jump on “**save**” page. Store username and password in session.
- After saving session, redirect to “fetchdata” page and read value. Put a LOGOUT link button here. On this page check authentication of user. User name and password must be “admin” and “admin@123” respectively.
 - If this condition is true then display welcome admin and display logout link on this page(fetchdata).
- By clicking on logout link user should jump to “destroy” page and destroy the session there and display the message “Session destroyed”. And give the link of “login” under that message. By clicking that link user will be redirected to the home page.
 - Else display “Please enter valid username and password” and login link on this page(fetchdata).

session.html

```

<html> <body>
  <center>
    <form method="get" action="/savesession">

      <div><input type = "text" name="uname" placeholder="Username"
></input></div>
      <div><input type = "password" name="pass"
placeholder="Password"></input></div>
      <input class="lj_in" type="submit" name="submit" value="Login">
    </form>
  </center>

```

```
</body> </html>
```

Session.js

```
var expr = require("express");
var app = expr();
var es = require("express-session");
app.use(es({
  resave:false,
  saveUninitialized:false,
  secret:"lju1 "
}));
app.get('/',(req,res)=>{
  res.sendFile(__dirname + "/session.html");
});
app.get("/save",(req,res)=>{
  var uname=req.query.uname;
  var pass=req.query.pass;

  req.session.uname = uname;
  req.session.password = pass;
  res.redirect("fetchsession")
})
app.get("/fetchdata",(req,res)=>{
  if(req.session.uname == "admin" && req.session.password=="admin@123"){
    res.write("<h1 style='color:green;'>Welcome Admin</h1>")
    res.write("<a href='/destory'>Logout</a>")
  }else{
    res.write("<h1 style='color:red;'>Please enter valid username or password</h1><a
href='/>Login</a>")
  }
  res.send();
});
app.get("/destroy",(req,res)=>{
  req.session.destroy(err => {
  if (err) {
    res.send(err)
  }
  else {
    res.clearCookie('connect.sid')
    res.redirect('/')
  }
  }) });
app.listen(6177);
```

9. Write an ExpressJS to take a UserName, Password, Textarea for “message” & submit button using get method.

1) After clicking submit button the content of submitted details should be represented on “/login” page along with one “show vowel” link.

2) By clicking “show vowel” link count of vowel used in submitted “message” will display on “/message” page. (Use next() to route page)

index.html

```
<html>
  <body>
    <form method="get" action="/login">
      <fieldset>
        <legend style="color:blue;text-align:center" >Log In </legend>
        <br>
        <label> User Name</label>
        <input type="text" name="name"/>
      </br>
      <label> Password</label>
      <input type="password" name="password"/>
      </br>
      <label> Comment</label>
      <textarea rows="5" cols="15" name="message"> </textarea>
      </br>
      <input type="submit">
      <input type="Reset">
    </fieldset>
  </form>
</body>
</html>
```

Vowel.js

```
var expr = require("express");
var app = expr();

app.use(expr.static("../public"));

app.get("/login",(req,res,next)=>{
  console.log(req.query);
  res.set("content-type","text/html");
  res.write("<center><h1>Welcome " + req.query.name + "</h1>");
  res.write("<center><h1> Your Password is " + req.query.password + "</h1>");
  res.write("<center><h2>Your message id is " + req.query.message+ "</h2>");
  res.write("<h3>You can Find vowel in message by clicking</h3><a
href='/message'>Message</a></center>");
  a=req.query.message
  next();
```

```
  })
  app.get("/login",(req,res)=>{
    res.send()
  }
)
app.get("/message",(req,res)=>{
  res.set("content-type","text/html");
  res.write("<center><h2>Vowel in Your message id : " + a + "</h2>");
  const string =a
  const vowels = ["a", "e", "i", "o", "u"]
  function countVowel(str) {
    let count = 0;
    for (let letter of str.toLowerCase()) {
      if (vowels.includes(letter)) {
        count++;
      }
    }
    return count
  }
  const result = countVowel(string);
  res.write(` ${result} `);
  res.send()
})
app.listen(5121,()=>{
  console.log("server start");
});
```

Task:

Write an express js script to configure the multer middleware. Perform following tasks.

- 1) Create a html form file named "form.html" in public folder. This file contain centrally oriented heading(h3) "Upload your File" in red color with background-color aqua. Along with choose file option(to browse and select file) and submit button(to upload the file). (Must use external css having name "effect.css" in public folder)
- 2) Create a js file to show result after uploading any type of file, message should be displayed on "/upload" page "(file original name) has been uploaded".(Css effect must include while running js code)
- 3) Save uploaded files to specific directory named "File". And in this folder file must be stored in format of "data-file.pdf" where "data" is the field name.

Solution**App.js (In src folder)**

```
const express = require('express');
const multer = require('multer');
const path = require('path');

// Initialize the app
const app = express();

// Serve static files from the public folder
app.use(express.static("../public",{index:"form.html"}))

// Set up storage using multer
const storage = multer.diskStorage({
  destination: "File",
  filename: function (req, file, cb) {
    cb(null, file.fieldname+"-file.pdf");
  }
});

// Initialize upload middleware
const upload = multer({ storage: storage }).single('data')

// Handle file upload
app.post('/upload', upload, (req, res) => {
  if (!req.file) {
    return res.status(400).send('No file uploaded.');
```

```

    res.send(`<p>${req.file.originalname} has been uploaded</p><a href="/">Go back</a>`);
  });

// Start the server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

```

form.html (In public folder)

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>File Upload</title>
  <link rel="stylesheet" href="/effect.css">
</head>
<body>
  <div class="container">
    <h3>Upload your File</h3>
    <form action="/upload" method="post" enctype="multipart/form-data">
      <input type="file" name="data" required>
      <button type="submit">Submit</button>
    </form>
  </div>
</body>
</html>

```

effect.css(In public folder)

```

body { background-color: aqua;}

h3 { color: red;}

//optional
.container { text-align: center;}

```

Task: Write a code to set up nodemailer in Express.js. Sender email id: “lju@gmail.com”. Receiver email ids: “student@gmail.com and faculty@gmail.com”. Mail subject should be “LJ University” Mail body contains “Welcome Student” in h3 tag and in table display data Date 28/06/23, Exam name FSD-2.

Solution:

```
const nodemailer = require('nodemailer');
const transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'lju@gmail.com',
    pass: 'yourpassword'
  }
});

const mailOptions = {
  from: 'lju@gmail.com',
  to: 'student@gmail.com, faculty@gmail.com',
  subject: 'LJ University',
  html: '<h3>welcome student</h3><table border="1px"><tr><th>date</th><th>Exam  
name</th></tr><tr><td> 28/06/23</td><td>FSD-2</td></tr>'
};

transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    console.log(err)
  }
  console.log("mail send")
});
```

Task: Write an express js script to configure the multer middleware. Perform following tasks.

1) Create a pug file named "file.pug". This file contains heading(h3) "Upload your CV" in red color. And, a form with input type file(to browse and select file) and submit(to upload the file).

- 2) Create a js file named "file.js" and link this js and pug file to browse pug file on "/home" page.
- 3) After uploading a file display message on "/upload" page "(file original name) has been uploaded".
- 4) Save uploaded files to specific directory named "upload". And in this folder file must be stored in format of "lju-file.pdf" where "lju" is the field name.

Solution:**xyz.pug**

```
html
body
  form(action="/upload" method="post" enctype="multipart/form-data")
    input(type="file" name="lju")
    input(type="submit" value="upload")
```

xyz.js

```
const express = require('express');
const multer = require('multer');
const app = express();
app.set('view engine', 'pug')
app.set('views', __dirname)
const storage = multer.diskStorage({
  destination: 'file',
  filename: function (req, file, cb) {
    cb(null, file.fieldname + '-file.pdf');
  }
});
const upload = multer({ storage: storage });
app.get("/", (req, res) => {
  res.render('xyz')
})

app.post('/upload', upload.single('lju'), (req, res) => {

  res.send(`<h3 style="text-align: center;">${req.file.originalname} has been uploaded</h3>`);

}).listen(3000);
```

Task: Write an expressJS code in which RESTapi is created for json object named Places I love which contains name, country, state, city and rating out of 10 (no decimal points) is given. upon passing ratings on the browser it should display the places having that rating. i.e. on localhost:30001/a/10 should display all the places having 10 ratings.

Solution:**api.js**

```
const expr = require("express");
const router = expr.Router();
const PlacesIlove=[
  {name:'manali',country:'india',state:'jammu',city:'manali',rating:8},
  {name:'kullu',country:'india',state:'jammu',city:'manali',rating:9},
  {name:'kashmir',country:'india',state:'jammu',city:'manali',rating:10}]
module.exports = router;
router.get("/:rating([0-9]){1,2})",(req,res)=>{
  var current_data = data.filter((i1)=>{
    if(i1.rating == req.params.rating){
      return true;
    }
  })
  if(current_data.length >0){
    res.send(current_data);
  }else{
    res.send("Not Found")
  }
})
```

main.js

```
const expr = require("express");
const app = expr();
const api = require("./api")
app.use("/a",api);
app.listen(30001);
```

Task: Prepare Simple For loop, if...else condition in PUG file.

//-Simple for loop:

```
//- html
//-   head
//-     title hello world
```

```
//-   body
//-   h1 for/each loop
//-   each n in [10,20,30,40,50]
//-   p= n
```

//-for loop with modification:

```
//- html
//-   head
//-     title hello world
//-   body
//-     h1 for/each loop
//-     each n,i in [10,20,30,40,50]
//-     p= i + "-" + n
```

//-if else condition:

```
html
  head
    title condition in pug
  body
    h1 My web applications
    - let user = { name: "Admin" , loggedIn:true, lastLogin:6}
    if user.loggedIn
      p
        | Welcome back
        strong #{user.name}
    else
      p
        |Please login with admin name
```

Task: Identify Output of Pug file

```
p(style="font-size:25px;")
  i Good morning
//
  p b LJU Students
```

```

p b hello students
p
  b
    i test
-var a=5
h4 My favorite color is
  i(style="text-decoration:underline")= 2*a
|Display content
this

```

Output:*Good morning***hello students***test***My favorite color is *red***

Display content

HTML file of above pug file to understand the structure

```

<p style="font-size:25px;"><i>Good morning</i></p>
<!--p b LJU Students-->
<p>b hello students</p>
<p> <b> <i>test </i></b></p>
<h4>My favourit color is <i style="color:red">red</i></h4>
Display content
<this> </this>

```

Task: (File Upload for ref only*)

Write an express js script that allows **only pdf** type file to be uploaded using the multer middleware and saves the file to the specific directory called **“specific”**. If file other than pdf has been uploaded then it will give an error message that “Only PDF format allowed”. Also, final output to be displayed in browser through pug file.

[All files in same folder]

In mult.html

```
<html>
  <form action="/uploadfile" method="post" enctype="multipart/form-data">
    <input type="file" name="myfile" accept=".pdf"/>
    <input type="submit" value="Upload"/>
  </form>
</html>
```

In one.pug

```
html
  head
    title FORM Data
  body
    h1 File #{file} have been uploaded
```

In mult.js

// require the installed packages

```
const express = require('express')
const multer = require('multer');
const app = express();
app.set("view engine", "pug");
```

// SET STORAGE

```
var store = multer.diskStorage({
  destination: "specific",
  filename: function (req, file, cb) {
    cb(null, file.originalname)
  }
})
```

// Allow only pdf to be uploaded

```
const filter = (req, file, cb) => {
  if (file.mimetype == "application/pdf" )
  {
    cb(null, true);
  }
  else
```

```
{
  cb(null, false);
  return cb('Only pdf format allowed!');
}
}
const maxSize=1024*1024*1024;
var upload = multer({ storage:store, limits:{ fileSize:maxSize},fileFilter:
filter}).single('myfile')

app.get('/',(req,res)=>{
  res.sendFile(__dirname + "/mult.html");
})
app.post('/uploadfile', upload, (req, res) => {
  if (req.file) {
    res.render(__dirname+"/one",{ file:req.file.originalname}) //one.pug
  }
})
app.listen(6782);
```

---To compare mimetype as well as extname (For ref*)---

```
const filter=(req,file,cb)=> {
  var filetypes=/jpg|jpeg|png/;
  var mimetype=filetypes.test(file.mimetype);
  var extname=filetypes.test(path.extname
    (file.originalname).toLowerCase());
  if(mimetype && extname)
  {
    return cb(null,true);
  }
  cb("error:fileuploaded is problematic");
}
}
```