

Example-1

You have been assigned to develop a user feedback form for a website using Express.js and cookies. Implement the following requirements: Process a form with the following fields: Name, Email , Message , 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 clicks 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 Logout link, user redirected to home page. Make app.js file use get method in express js.

app.js

```
const express = require('express');
const cp = require('cookie-parser');
const app = express();
app.use(cp());
app.use(express.urlencoded({ extended: true }));
app.use(express.static(__dirname, {index: 'feedback.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');  
});
```

feedback.html

```
<!DOCTYPE html>  
<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>
```

Example-2

Write express js script to perform tasks as asked below.

1. Create one HTML file which contains two number type input fields, one dropdown which contains options to select like (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 number". Also, user must select any type of 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 Both numbers should be stored in cookies which expires in 50 seconds. Respective calculations will be performed on the page "/calc".

cookie.js

```
const express = require('express');
const cookieParser = require('cookie-parser');

const app = express();

app.use(express.urlencoded({ extended: true }));
app.use(cookieParser());
app.use(express.static(__dirname, {index: 'cookie.html'}));

// Handle form submission and perform calculations
app.post('/calc', (req, res) => {
  const { number1, number2, operation } = req.body;

  const num1 = parseFloat(number1);
  const num2 = parseFloat(number2);

  // Store numbers in cookies that expire in 50 seconds
  res.cookie('number1', num1, { maxAge: 50000 });
  res.cookie('number2', num2, { maxAge: 50000 });
  // Validate input fields and operation
  if (num1 <= 0 || num2 <= 0) {
    return res.send('Please enter valid numbers greater than 0.');
```

```
switch (operation) {
  case 'addition':
    result = num1 + num2;
    break;
  case 'subtraction':
    result = num1 - num2;
    break;
  case 'multiplication':
    result = num1 * num2;
    break;
  case 'division':
    if (num2 === 0) {
      return res.send('Division by zero is not allowed.');
```

cookie.html

```
<body>
<h1>Calculation Form</h1>
<form action="/calc" method="POST">
  <input type="number" id="number1" name="number1" required><br>
  <br>
  <input type="number" id="number2" name="number2" required><br>
  <br>
  <select id="operation" name="operation" required>
    <option value="">Select an operation</option>
    <option value="addition">Addition</option>
    <option value="subtraction">Subtraction</option>
    <option value="multiplication">Multiplication</option>
    <option value="division">Division</option>
  </select>
  <br>
  <button type="submit">Calculate</button>
</form>
</body>
```

Example-3

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.

file.pug

```
html
  head
    title Pug Form
  body
    h3(style="color:red") Upload your CV
    form(action="/upload",method="post",enctype="multipart/form-data")
      input(type="file", name="lju")
      input(type="submit",value="Submit")
```

file.js

```
// require the installed packages
const express = require('express')
const multer = require('multer');
const app = express();

//CREATE EXPRESS APP
app.set("view engine", "pug");
var p = require("path");
const staticp = p.join(__dirname, "../frontend");
app.get("/home",(req,res)=>{
  res.render(staticp+"/file");
});

// SET STORAGE
var storage = multer.diskStorage({
  destination:"upload",
  filename: function (req, file, cb) {
    cb(null, file.fieldname+"-file.pdf")
  }
})
```

```

var upload = multer({ storage: storage })
app.post('/upload', upload.single('lju'), (req, res) => {
  const file = req.file
  console.log(file)
  if (file) {
    res.send("<h1>File <span style='color:red>'"+ file.originalname + "'</span> has been
uploaded in <span style='color:red>' + file.destination + " </span>folder")
  }
})
app.listen(6788);

```

Example-4

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.

form.js

```

const express = require('express');
const multer = require('multer');
const app = express();

app.use(express.static(__dirname, {index:'co1.html'}));

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

```

```
    }  
  });  
  
const upload = multer({ storage: storage });  
  
// Route to handle file upload  
app.post('/upload', upload.single('data'), (req, res) => {  
  const file = req.file;  
  res.send(`  
    <h2>${file.originalname} has been uploaded</h2>  
    <a href="/">Go back</a>  
  `);  
});  
app.listen(3000)
```

form.html

```
<head>  
  <link href="effect.css" rel="stylesheet">  
</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><br><br>  
      <button type="submit">Upload</button>  
    </form>  
  </div>  
</body>
```

effect.css

```
body {  
  justify-content: center;  
  align-items: center;  
}  
  
.container {  
  text-align: center;  
  background-color: aqua;  
}  
  
h3 {  
  color: red;
```

}

Example-5

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)

Example-6

Write expressJS script to perform task as asked:

- (A) Create one HTML file which contains text-field named username, one dropdown which contains options of country like India, USA, Canada, Australia. & one submit button.
- (B) Once user clicked on submit button it will jump to next page than username & You are from “country name” which ever selected from drop box should be printed.
- (C) Use get method to request data.

Example-7

Process a form using post method. Form has fields like username, password, confirm password, gender, submit and reset buttons. After entering all fields, If password and confirm password matches, then form should be processed and all relevant and selected fields’ values should be printed. Otherwise, by printing warning message in red color, it should terminate. **No need to write file having form elements.**

Example-8

write a script to meet following requirements.

- 1) create index.html page and open it on localhost
- 2) after clicking submit button, it should jump to savesessionpage.store username in session.
- 3) After saving session, redirect to fetchsession page and read session value. put a logout link button here.
- 4) destroy the session on this page and redirect to index.html

Example-9

Write an expressJS code which loads login.html file upon browsing localhost:3010. The Login.html file contains input for username,password and two checkboxes named remember and subscribe,on submitting the file it should go to /data page where username,password and selected checkboxes are pinte,in addition to that a logout button should be there. By clicking this button it should go back to home page “localhost:3010/”.(GET/POST any method can be used)[Write all necessary files code]

Example-10

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.

Example-11

Write ExpressJS script to load following table using pug file on localhost.

Sr.no	Subject	Faculty Name
1	FSD-2	ABC
2	Python-2	DEF
3	COA	XYZ