

```

1 //QB-211
2 // Write an express.js script to define one JSON array of 3 objects having members name
  and age.
3 // Names must be sorted according to age. If user requests "sorted Names" URL, then all
  names should be printed according to ascending order of age.
4 const expr = require("express")
5 const app = expr();
6 student = [
7   {name:"abc",age:28},
8   {name:"def",age:40},
9   {name:"xyz",age:50}]
10
11 //home page
12 app.get ("/", (req,res)=>
13 {
14   res.set ("content-type","text/html")
15   res.send (student)
16 })
17 //sort page
18 app.get ("/sort", (req,res)=>
19 {
20   res.set ("content-type","text/html")
21   var des = student.sort((a,b)=>b.age-a.age)
22   console.log(des)
23   for (k=0;k<des.length;k++)
24   {
25     res.write ("<center><h2>"+des[k].name+ " = " +des[k].age + "</h2></center>")
26   }
27   res.send ()
28 })
29 app.listen (5200)
30
31 //QB-212
32 // Process a form using post method. Form has fields like username, password, confirm
  password, gender, submit and reset buttons.
33 // After entering all fields, If password and confirm password matches, then form should
  be processed and all relevant and selected fields'
34 // values should be printed. Otherwise, by printing warning message in red color, it
  should terminate. No need to write file having form elements.
35 const express = require('express');//approch diffrent method
36 const bodyParser = require('body-parser');
37 const app = express();
38
39 const port = 3000;
40
41 // Middleware to parse URL-encoded data
42 app.use(bodyParser.urlencoded({ extended: true }));
43
44 app.get('/', (req, res) =>
45 {
46   // Form HTML
47   res.send(`
48     <form action="/submit" method="POST">
49       <label for="username">Username:</label>
50       <input type="text" id="username" name="username" required><br><br>
51       <label for="password">Password:</label>
52       <input type="password" id="password" name="password" required><br><br>
53       <label for="confirmPassword">Confirm Password:</label>
54       <input type="password" id="confirmPassword" name="confirmPassword"
55         required><br><br>
56       <label for="gender">Gender:</label>
57       <select id="gender" name="gender" required>
58         <option value="male">Male</option>
59         <option value="female">Female</option>
60         <option value="other">Other</option>
61       </select><br><br>
62       <button type="submit">Submit</button>
63       <button type="reset">Reset</button>
64     </form>

```

```

64     `);
65 });
66
67 app.post('/submit', (req, res) =>
68     {
69         const { username, password, confirmPassword, gender } = req.body;
70
71         if (password !== confirmPassword)
72         {
73             res.send(`
74                 <p style="color: red;">Passwords do not match!</p>
75                 <a href="/">Go back</a>
76             `);
77         } else {
78             res.send(`
79                 <h2>Form submitted successfully!</h2>
80                 <p>Username: ${username}</p>
81                 <p>Gender: ${gender}</p>
82             `);
83         }
84     });app.listen(3000)
85
86 //QB-213
87 // write an express.js script to load an HTML file having username and password and
88 // submit button. On clicking submit button.
89 // It should jump on "check" page using "POST" method. If username is "admin" , then
90 // jump on next middleware to print "welcome... admin" ,
91 // if username is not "admin" , then stay on same middleware to print "warning msg" in
92 // red color.
93
94 //App.js
95 const express=require("express");
96 const app=express();
97 app.use(express.urlencoded({extended:false}));
98 app.use(express.static(__dirname));
99 app.post("/check", (req,res,next)=>
100 {
101     if(req.body.uname=="admin")
102     {
103         next();
104     }
105     else
106     {
107         res.send("<h1 style='color:red'>wrong credentials</h1>");
108     }
109 });
110 app.post("/check", (req,res)=>
111 {
112     res.write("<h1>welcome..." + req.body.uname+ "</h1>");
113     res.send();
114 }
115 ).listen(3001);
116
117 //Index.html
118 <html>
119 <body>
120 <form action ="/check" method="post">
121 Username: <input type="text" name="uname"/>
122 password: <input type="text" name="pwd"/>
123 <input type="submit"/>
124 </form>
125 </body>
126 </html>
127
128 //QB-214
129 // write a script to meet foll requirements.

```

```

130 // 1) create index.html page and open it on localhost 2) after clicking submit button,
131 // it should jump to savesessionpage.store username in session.
132 // 3) After saving session, redirect to fetchsession page and read session value. put a
133 // logout link button here.
134 // 4) destroy the session on this page and redirect to index.html
135
136 //session.js
137 var express = require("express");
138 var app = express();
139 var es = require("express-session");
140 app.use(es(
141 {
142     resave:false,
143     saveUninitialized:false,
144     secret:"1jul"
145 }));
146 app.use(express.static(__dirname));
147 app.get("/savesession", (req, res) =>
148 {
149     req.session.uname = req.query.uname;
150     req.session.password = req.query.password;
151     res.redirect("fetchsession")
152 })
153 app.get("/fetchsession", (req, res) =>
154 {
155     res.write("<h1>Welcome " + req.session.uname + "</h1>")
156     res.write("<a href='/deletesession'>Logout</a>")
157     res.send();
158 })
159 app.get("/deletesession", (req, res) =>
160 {
161     req.session.destroy()
162     res.redirect('/')
163 })
164 app.listen(6177);
165
166 //QB-215
167 // write an express.js script to print "hello World" also run on localhost:4000
168 const express = require('express');
169 const app = express();
170 app.get('/', (req, res) =>
171 {
172     res.send('Hello World');
173 })
174 app.listen(4000)
175
176 //QB-216
177 //write an express.js script to use of routing method using home , about , contact ,
178 // temp page and print message.
179 // /home- welcome to my home page , /about -welcome to my about page, /contact-welcome
180 // to my contact page ,
181 // /temp - welcome to my temp page
182 const express = require('express');
183 const app = express();
184 app.get('/home', (req, res) =>
185 {
186     res.send('Welcome to my home page');
187 })
188 app.get('/about', (req, res) =>
189 {
190     res.send('Welcome to my about page');
191 })
192 app.get('/contact', (req, res) =>
193 {
194     res.send('Welcome to my contact page');
195 })
196

```

```

195 app.get('/temp', (req, res) =>
196 {
197     res.send('Welcome to my temp page');
198 });app.listen(4000)
199
200
201
202 //QB-217
203 // write an express.js script to make one index.html file in this file we write simple
    heading tag and make one css file
204 // index.css put style using appropriate selector and link with js file and show output
    on localhost:3030.
205 // your-app/
206 // |-- app.js
207 // |-- public/
208 // |   |-- index.html
209 // |   |-- index.css
210
211 //index.html
212 <html lang="en">
213 <head>
214     <meta charset="UTF-8">
215     <meta name="viewport" content="width=device-width, initial-scale=1.0">
216     <title>Index Page</title>
217     <link rel="stylesheet" href="index.css">
218 </head>
219 <body>
220     <h1 class="heading">Hello, World!</h1>
221 </body>
222 </html>
223
224 index.css
225 .heading
226 {
227     color: blue;
228     text-align: center;
229     margin-top: 50px;
230 }
231
232 //app.js
233 const express = require('express');
234 const path = require('path');
235 const app = express();
236
237 // Middleware to serve static files
238 app.use(express.static(path.join(__dirname, 'public')));
239
240 app.get('/', (req, res) =>
241 {
242     res.sendFile(path.join(__dirname, 'public', 'index.html'));
243 });app.listen(4000)
244
245
246
247
248 //QB-218
249 // write an express.js to implement a form that allows users to submit data via POST
    request.
250 // Write code to create a route that handles the POST request and logs the submitted
    data to the console.
251
252 index.html
253 <!DOCTYPE html>
254 <html lang="en">
255 <head>
256     <meta charset="UTF-8">
257     <meta name="viewport" content="width=device-width, initial-scale=1.0">
258     <title>Form Submission</title>
259 </head>

```

```

260     <body>
261       <h1>Submit Your Data</h1>
262       <form action="/submit" method="POST">
263         <label for="name">Name:</label>
264         <input type="text" id="name" name="name" required<br><br>
265         <label for="email">Email:</label>
266         <input type="email" id="email" name="email" required<br><br>
267         <button type="submit">Submit</button>
268       </form>
269     </body>
270   </html>
271
272   //app.js
273   const express = require('express');
274   const path = require('path');
275   const bodyParser = require('body-parser');
276   const app = express();
277
278   app.use(bodyParser.urlencoded({ extended: true }));
279   app.use(express.static(path.join(__dirname, 'public')));
280
281
282   app.post('/submit', (req, res) =>
283   {
284     const formData = req.body;
285     console.log('Form Data:', formData);
286     res.send('Form submitted successfully!');
287   });app.listen(4000)
288
289
290
291
292   //QB-219 how can you create and use a middleware function in express js?
293   const express = require('express');
294   const app = express();
295   // Create a middleware function
296   function logRequest(req, res, next)
297   {
298     console.log(`${req.method} ${req.url}`);
299     next();
300   }
301
302   app.use(logRequest);
303   app.get('/', (req, res) =>
304   {
305     res.send('Welcome to the Home Page');
306   });
307
308   app.get('/about', (req, res) =>
309   {
310     res.send('Welcome to the About Page');
311   });
312
313   app.post('/submit', (req, res) =>
314   {
315     res.send('Form submitted successfully!');
316   });app.listen(3000)
317
318   // Apply middleware to specific routes
319   app.get('/about', logRequest, (req, res) =>
320   {
321     res.send('Welcome to the About Page');
322   });
323
324   app.post('/submit', logRequest, (req, res) =>
325   {
326     res.send('Form submitted successfully!');
327   });
328

```

```

329
330
331 //QB-220
332 //write an express js to link html , css and js file and show one image o.jpg. And also
show the description of image.
333 const express = require('express');
334 const app = express();
335 const path = require('path');
336 app.use(express.static(path.join(__dirname, 'public')));
337
338 app.get('/', (req, res) =>
339 {
340     res.sendFile(path.join(__dirname, 'public', 'index.html'));
341 });app.listen(3000);
342
343 //index.html
344 <!DOCTYPE html>
345 <html lang="en">
346 <head>
347     <meta charset="UTF-8">
348     <meta name="viewport" content="width=device-width, initial-scale=1.0">
349     <title>Image Display</title>
350     <link rel="stylesheet" href="style.css">
351     <script src="script.js" defer<</script>
352 </head>
353 <body>
354     <h1 class="title">Image Display Page</h1>
355     <div class="image-container">
356         
357         <p class="description">This is the description of the image.</p>
358     </div>
359 </body>
360 </html>
361
362
363
364 //QB-221
365 // using Express js make student information form only two fields like name and mobile
no and send data on console after submitting "submit" button
366 //app.js
367 const express = require('express');
368 const path = require('path');
369 const bodyParser = require('body-parser');
370
371 app.use(bodyParser.urlencoded({ extended: true }));
372 app.use(express.static(path.join(__dirname, 'public')));
373
374 app.get('/', (req, res) =>
375 {
376     res.sendFile(path.join(__dirname, 'public', 'index.html'));
377 });
378 app.post('/submit', (req, res) =>
379 {
380     const { name, mobile } = req.body;
381     console.log('Submitted Data:');
382     console.log('Name:', name);
383     console.log('Mobile:', mobile);
384     res.send('Form submitted successfully!');
385 });app.listen(4000)
386
387 //index.html
388 <form action="/submit" method="POST">
389     <label for="name">Name:</label>
390     <input type="text" id="name" name="name" required<br><br>
391     <label for="mobile">Mobile Number:</label>
392     <input type="tel" id="mobile" name="mobile" required<br><br>
393     <button type="submit">Submit</button>
394 </form>
395

```

```

396
397
398
399 //QB-222
400 //Write an ExpressJS to take a textarea & submit button. After clicking submit button
the content
401 // of textarea should be represented on next page by writing each sentence (separated by
dot) in new line.
402 const express = require('express');
403 const app = express();
404 const path = require('path');
405 const bodyParser = require('body-parser');
406
407
408 app.use(bodyParser.urlencoded({ extended: true }));
409 app.use(express.static(path.join(__dirname, 'public')));
410
411 app.get('/', (req, res) =>
412 {
413     res.sendFile(path.join(__dirname, 'public', 'index.html'));
414 });
415
416 app.post('/submit', (req, res) =>
417 {
418     const { text } = req.body;
419     const sentences = text.split('.'); // Split text into sentences based on dot (.)
420     res.render('result', { sentences });
421 });
422 app.set('view engine', 'ejs');
423 app.set('views', path.join(__dirname, 'views'));
424 app.listen(3000)
425
426
427 // //QB-223
428 // Write a script to meet following requirement using ExpressJS.
429 // 1. Create index.html which contains login page having fields like Username, Password
& Gender. Open it on localhost.
430 // 2. After clicking on submit button, it should jump to savesession page. store
username & gender in session.
431 // 3. After saving session, redirect to fetchsession page and read session values. Put a
logout link button here.
432 // 4. After clicking logout button, it jumps to deletesession page.
433 // 5. Destroy the session on this page & redirect to index.html page.
434
435 //index.html
436 <!DOCTYPE html>
437 <html lang="en">
438 <head>
439     <meta charset="UTF-8">
440     <title>Login Page</title>
441     <link rel="stylesheet" href="style.css">
442 </head>
443 <body>
444     <div class="container">
445         <h1>Login Page</h1>
446         <form action="/savesession" method="POST">
447             <label for="username">Username:</label>
448             <input type="text" id="username" name="username" required<br><br>
449             <label for="password">Password:</label>
450             <input type="password" id="password" name="password" required<br><br>
451             <label for="gender">Gender:</label>
452             <select id="gender" name="gender" required>
453                 <option value="male">Male</option>
454                 <option value="female">Female</option>
455             </select><br><br>
456             <button type="submit">Submit</button>
457         </form>
458     </div>
459 </body>

```

```

460     </html>
461
462     //app.js
463     const express = require('express');
464     const session = require('express-session');
465     const path = require('path');
466     const bodyParser = require('body-parser');
467
468     app.use(bodyParser.urlencoded({ extended: true }));
469     app.use(express.static(path.join(__dirname, 'public')));
470     app.use(session(
471         {
472             secret: 'your-secret-key',
473             resave: false,
474             saveUninitialized: true
475         }
476     ));
477     app.get('/', (req, res) =>
478     {
479         res.sendFile(path.join(__dirname, 'public', 'index.html'));
480     });
481
482     app.post('/save-session', (req, res) =>
483     {
484         const { username, gender } = req.body;
485         req.session.username = username;
486         req.session.gender = gender;
487         res.redirect('/fetch-session');
488     });
489
490     app.get('/fetch-session', (req, res) =>
491     {
492         const username = req.session.username;
493         const gender = req.session.gender;
494         res.render('fetch-session', { username, gender });
495     });
496
497     app.get('/delete-session', (req, res) =>
498     {
499         req.session.destroy(err =>
500         {
501             if (err)
502             {
503                 console.error('Error destroying session:', err);
504                 res.status(500).send('Error destroying session');
505             }
506             else
507             {
508                 res.redirect('/');
509             }
510         });
511     }); app.listen(5000)
512
513     //QB-224
514     // You have been assigned to develop a user feedback form for a website using Express.js
515     // and cookies.
516     // Implement the following requirements: Process a form with the following fields:
517     // Name, Email , Message ,
518     // Rating (radio buttons: Bad, Average, Good, Very Good, Excellent) When the user
519     // submits the form, store their feedback
520     // information (name, email, message, and rating) in a cookie named "feedback" that
521     // expires in 10 seconds. Display
522     // a confirmation message to the user after successfully submitting the form & Create a
523     // link to display the feedback details
524     // stored in the "feedback" cookie. When the user click to the link, retrieve the
525     // feedback information from the cookie and
526     // display it on the page also include a link on the feedback details page to Logout.
527     // When the user clicks the link, user
528     // redirected to home page. Make app.js file use get method in express js. No need to

```



```

522 write html file having form elements.
523 const express = require('express');
524 const app = express();
525 const cookieParser = require('cookie-parser');
526 const path = require('path');
527 const bodyParser = require('body-parser');
528
529 app.use(bodyParser.urlencoded({ extended: true }));
530 app.use(express.static(path.join(__dirname, 'public')));
531 app.use(cookieParser());
532
533
534 app.get('/submit', (req, res) =>
535 {
536     const { name, email, message, rating } = req.query;
537     res.cookie('feedback', JSON.stringify({ name, email, message, rating }), { maxAge:
10000 }); // Expires in 10 seconds
538     res.send('<h2>Thank you for your feedback!</h2><p><a href="/feedback">Click here</a>
to view your feedback details.</p>');
539 });
540 app.get('/feedback', (req, res) =>
541 {
542     const feedback = req.cookies.feedback ? JSON.parse(req.cookies.feedback) : null;
543
544     if (feedback)
545     {
546
547         // Display feedback details
548         res.send(`
549             <h2>Your Feedback Details:</h2>
550             <p><strong>Name:</strong> ${feedback.name}</p>
551             <p><strong>Email:</strong> ${feedback.email}</p>
552             <p><strong>Message:</strong> ${feedback.message}</p>
553             <p><strong>Rating:</strong> ${feedback.rating}</p>
554             <p><a href="/logout">Logout</a></p>`);
555     }
556     else
557     {
558         res.send('<p>No feedback submitted yet.</p>');
559     }
560 });
561 app.get('/logout', (req, res) =>
562 {
563     res.clearCookie('feedback');
564     res.redirect('/');
565 });app.listen(3000)
566
567
568
569 //QB-225
570 // write express script to maintain session and print how many times user visit the
page. For ex., if user visit first time
571 // ,"you have visited page First time" message will print. if user visit second time
,"you have visited page second time" message will print.
572 // and so on
573 const express = require('express');
574 const session = require('express-session');
575 const path = require('path');
576
577 app.use(session(
578 {
579     secret: 'your-secret-key',
580     resave: false,
581     saveUninitialized: true
582 }));
583
584 app.get('/', (req, res) =>
585 {

```

```

586     let visitCount = req.session.visitCount || 0;
587     visitCount++;
588     req.session.visitCount = visitCount;
589
590     let visitText = '';
591     switch (visitCount)
592     {
593         case 1:
594             visitText = 'First';
595             break;
596         case 2:
597             visitText = 'Second';
598             break;
599         case 3:
600             visitText = 'Third';
601             break;
602         default:
603             visitText = `${visitCount}th`;
604             break;
605     }
606     res.send(`<h1>You have visited the page ${visitText} time.</h1>`);
607 });app.listen(3000)
608
609
610 //QB-226
611 // Write an express.js script to define one JSON array of 3 objects having members name
612 // and age.
613 // If user requests "maxName" URL, then only the details of oldest person should be
614 // displayed.
615 const express = require('express');
616 const app = express();
617 const people = [
618     { name: 'Alice', age: 30 },
619     { name: 'Bob', age: 25 },
620     { name: 'Charlie', age: 35 }
621 ];
622 app.get('/maxName', (req, res) =>
623 {
624     let oldestPerson = findOldestPerson(people);
625     res.json(oldestPerson);
626 });
627 function findOldestPerson(arr)
628 {
629     if (arr.length === 0) return null;
630
631     let oldest = arr[0];
632     for (let i = 1; i < arr.length; i++)
633     {
634         if (arr[i].age > oldest.age)
635         {
636             oldest = arr[i];
637         }
638     }
639     return oldest;
640 }
641 app.get('/', (req, res) =>
642 {
643     res.json(people);
644 });
645 app.listen(4000)
646
647 //QB-227
648 // Write an express.js script to define 2 pages. 1st page has username and password.
649 // Store this username to cookie on 2nd page. Cookie must stay live for 1 day.
650 const express = require('express');
651 const app = express();
652 const cookieParser = require('cookie-parser');
653 const path = require('path');
```

```

652 const bodyParser = require('body-parser');
653
654 app.use(bodyParser.urlencoded({ extended: true }));
655 app.use(express.static(path.join(__dirname, 'public')));
656 app.use(cookieParser());
657
658 app.get('/', (req, res) =>
659 {
660     res.sendFile(path.join(__dirname, 'public', 'index.html'));
661 });
662
663 app.post('/submit', (req, res) =>
664 {
665     const { username } = req.body;
666
667     res.cookie('username', username,
668     {
669         maxAge: 24 * 60 * 60 * 1000, // 1 day in milliseconds
670         httpOnly: true // cookie is only accessible via HTTP(S)
671     });
672     res.redirect('/welcome');
673 });
674 app.get('/welcome', (req, res) =>
675 {
676     const username = req.cookies.username;
677     if (username)
678     {
679         res.send(`<h1>Welcome, ${username}!</h1>`);
680     }
681     else
682     {
683         res.redirect('/');
684     }
685 });
686 app.listen(4000)
687
688
689 // //QB-228
690 // Write an expressJS code which loads login.html file upon browsing localhost:3010.
691 // The Login.html file contains input for username,password and two checkboxes named
692 // remember and subscribe,on
693 // submitting the file it should go to /data page where username,password and selected
694 // checkboxes are pintoed,in
695 // addition to that a logout button should be there,onclicking this button it should go
696 // back to home page
697 // "localhost:3010/". (GET/POST any method can be used) [Write all necessary files code]
698
699 //index.html
700 <!DOCTYPE html>
701 <html lang="en">
702 <head>
703     <meta charset="UTF-8">
704     <title>Login Page</title>
705     <link rel="stylesheet" href="/style.css">
706 </head>
707 <body>
708     <div class="container">
709         <h1>Login Page</h1>
710         <form action="/login" method="POST">
711             <label for="username">Username:</label>
712             <input type="text" id="username" name="username" required><br><br>
713
714             <label for="password">Password:</label>
715             <input type="password" id="password" name="password" required><br><br>
716
717             <input type="checkbox" id="remember" name="remember">
718             <label for="remember">Remember Me</label><br><br>
719
720             <input type="checkbox" id="subscribe" name="subscribe">

```

```

718         <label for="subscribe">Subscribe to Newsletter</label><br><br>
719
720         <button type="submit">Login</button>
721     </form>
722 </div>
723 </body>
724 </html>
725
726 const express = require('express');
727 const app = express();
728 const cookieParser = require('cookie-parser');
729 const path = require('path');
730
731 app.use(express.static(path.join(__dirname, 'public')));
732 app.use(express.urlencoded({ extended: true }));
733 app.use(cookieParser());
734
735 app.get('/', (req, res) =>
736 {
737     res.sendFile(path.join(__dirname, 'public', 'login.html'));
738 });
739 app.post('/login', (req, res) =>
740 {
741     const { username, password, remember, subscribe } = req.body;
742
743     if (remember === 'on')
744     {
745         res.cookie('remember', 'true', { maxAge: 24 * 60 * 60 * 1000 }); // 1 day in
746         milliseconds
747     }
748     else
749     {
750         res.clearCookie('remember');
751     }
752
753     if (subscribe === 'on')
754     {
755         res.cookie('subscribe', 'true');
756     }
757     else
758     {
759         res.clearCookie('subscribe');
760     }
761     res.redirect('/data');
762 });
763 app.get('/data', (req, res) =>
764 {
765     const username = req.cookies.remember ? req.cookies.username : 'Guest';
766     const rememberMsg = req.cookies.remember ? 'Yes' : 'No';
767     const subscribeMsg = req.cookies.subscribe ? 'Yes' : 'No';
768
769     res.send(`
770         <h1>Welcome, ${username}!</h1>
771         <p><strong>Remember Me:</strong> ${rememberMsg}</p>
772         <p><strong>Subscribe to Newsletter:</strong> ${subscribeMsg}</p>
773         <form action="/logout" method="POST">
774             <button type="submit">Logout</button>
775         </form>
776     `);
777 });
778 app.post('/logout', (req, res) =>
779 {
780     res.clearCookie('username');
781     res.clearCookie('remember');
782     res.clearCookie('subscribe');
783     res.redirect('/');
784 });
785 app.listen(4000)

```

```

786
787
788 //QB-229
789 // Write an express.js script to define one JSON array of 3 objects having members
790 // 'name' and 'height'. height must be sorted descending order according to name.
791 const express = require('express');
792 const app = express();
793
794 const people1 =
795 [
796   { name: 'Alice', height: 165 },
797   { name: 'Bob', height: 180 },
798   { name: 'Charlie', height: 175 }
799 ];
800
801 people1.sort((a, b) => {
802   if (a.name > b.name) return -1;
803   if (a.name < b.name) return 1;
804   return 0;
805 });
806 app.get('/sortedData', (req, res) =>
807 {
808   res.json(people1);
809 });
810 app.listen(5000)
811
812
813 //QB-230
814 // Write an express.js script to define one JSON array of 3 objects having members
815 // 'name' and 'score'. score must be sorted descending order according to name.
816 const express = require('express');
817 const app = express();
818 const students = [
819   { name: 'Alice', score: 85 },
820   { name: 'Bob', score: 92 },
821   { name: 'Charlie', score: 78 }
822 ];
823
824 students.sort((a, b) => {
825   if (a.name > b.name) return -1;
826   if (a.name < b.name) return 1;
827   return 0;
828 });
829
830 app.get('/sortedData', (req, res) =>
831 {
832   res.json(students);
833 });
834 app.listen(5000)
835
836
837 //QB-231
838 // Write an express js script to link html , css and js file to show message "LJ
839 // University with a difference" with font size 50px and color "blue".
840
841 //index.html
842 <!DOCTYPE html>
843 <html lang="en">
844 <head>
845   <meta charset="UTF-8">
846   <title>LJ University</title>
847   <link rel="stylesheet" href="/style.css">
848   <script src="/script.js" defer</script>
849 </head>
850 <body>
851   <div class="container">
852     <h1 id="message">LJ University with a difference</h1>
853   </div>

```

```

853 </body>
854 </html>
855
856 //style.Css
857 body
858 {
859     font-family: Arial, sans-serif;
860     background-color: #f0f0f0;
861     text-align: center;
862 }
863
864 .container
865 {
866     margin-top: 100px;
867 }
868
869 #message
870 {
871     font-size: 50px;
872     color: blue;
873 }
874
875 //app.js
876 const express = require('express');
877 const path = require('path');
878 app.use(express.static(path.join(__dirname, 'public')));
879 app.listen(5200)
880
881
882 //QB-232
883 // Write express js script to perform tasks as asked below.
884 // 1. Create one HTML file which contains two number type input fields, one dropdown
885 // which contains options to select like
886 // (addition, subtraction, multiplication, division) and one submit button.
887 // 2. The input fields must contain the value greater than 0 else it will give a message
888 // "Please enter the valid number".
889 // Also, user must select any type of formula from the dropdown else give a message "You
890 // have not selected any formula".
891 // (Message will be displayed on "/calc" page.)
892 // 3. If one formula is selected and numbers are entered then Both numbers should be
893 // stored in cookies which expires in 50 seconds.
894 // Respective calculations will be performed on the page "/calc".
895
896 //index.html
897 <!DOCTYPE html>
898 <html lang="en">
899 <head>
900     <meta charset="UTF-8">
901     <title>Calculator</title>
902 </head>
903 <body>
904     <div class="container">
905         <h2>Calculator</h2>
906         <form action="/calc" method="POST">
907             <label for="num1">Number 1:</label>
908             <input type="number" id="num1" name="num1" min="1" required><br><br>
909
910             <label for="num2">Number 2:</label>
911             <input type="number" id="num2" name="num2" min="1" required><br><br>
912
913             <label for="operation">Operation:</label>
914             <select id="operation" name="operation" required>
915                 <option value="">Select Operation</option>
916                 <option value="addition">Addition</option>
917                 <option value="subtraction">Subtraction</option>
918                 <option value="multiplication">Multiplication</option>
919                 <option value="division">Division</option>
920             </select><br><br>

```

```

918         <button type="submit">Calculate</button>
919     </form>
920 </div>
921 </body>
922 </html>
923
924 //app.js
925 const express = require('express');
926 const app = express();
927 const path = require('path');
928
929 app.use(express.static(path.join(__dirname, 'public')));
930 app.use(express.urlencoded({ extended: true }));
931
932 app.post('/calc', (req, res) =>
933 {
934     const { num1, num2, operation } = req.body;
935
936     if (num1 <= 0 || num2 <= 0)
937     {
938         return res.send('Please enter valid numbers (greater than 0).');
939     }
940
941     if (!operation)
942     {
943         return res.send('Please select an operation.');

```

```

984
985 const express = require('express');
986 const app = express();
987 const path = require('path');
988
989 app.use(express.urlencoded({ extended: true }));
990 app.use(express.static(path.join(__dirname, 'views')));
991
992 app.get('/', (req, res) =>
993 {
994     res.sendFile(path.join(__dirname, 'views', 'login.html'));
995 });
996
997 app.get('/login', (req, res, next) =>
998 {
999     const { username, password, message } = req.query;
1000     res.locals.username = username;
1001     res.locals.password = password;
1002     res.locals.message = message;
1003     next();
1004 }, (req, res) =>
1005 {
1006     res.render('login.html',
1007     {
1008         username: res.locals.username,
1009         password: res.locals.password,
1010         message: res.locals.message
1011     });
1012 });
1013 app.get('/message', (req, res, next) =>
1014 {
1015     const { message } = res.locals;
1016     const vowelCount = countVowels(message);
1017     res.locals.vowelCount = vowelCount;
1018     next();
1019 },
1020 (req, res) =>
1021 {
1022     res.render('message.html',
1023     {
1024         vowelCount: res.locals.vowelCount
1025     });
1026 });
1027 function countVowels(str)
1028 {
1029     const vowels = 'aeiouAEIOU';
1030     let count = 0;
1031     for (let char of str)
1032     {
1033         if (vowels.includes(char))
1034         {
1035             count++;
1036         }
1037     }
1038     return count;
1039 }
1040 app.listen(5000)
1041
1042
1043 //QB-234
1044 //Write an express.js script to print "The Pacific Ocean is the largest and deepest of
the world ocean" also run on appropriate localhost.
1045 const express = require('express');
1046 app.get('/', (req, res) =>
1047 {
1048     res.send('The Pacific Ocean is the largest and deepest of the world ocean');
1049 });app.listen(3000)
1050
1051

```



```

1052 //QB-235
1053 // Write expressJS script to perform task as asked:
1054 // (A) Create one HTML file which contains text-field named username, one dropdown which
contains options of country like India, USA, Canada, Australia.
1055 // & one submit button.
1056 // (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.
1057 // (C) Use get method to request data.
1058
1059 //index.html
1060 <!DOCTYPE html>
1061 <html lang="en">
1062 <head>
1063     <meta charset="UTF-8">
1064     <title>User Information</title>
1065 </head>
1066 <body>
1067     <div class="container">
1068         <h2>User Information</h2>
1069         <form action="/result" method="GET">
1070             <label for="username">Username:</label>
1071             <input type="text" id="username" name="username" required><br><br>
1072
1073             <label for="country">Country:</label>
1074             <select id="country" name="country" required>
1075                 <option value="">Select Country</option>
1076                 <option value="India">India</option>
1077                 <option value="USA">USA</option>
1078                 <option value="Canada">Canada</option>
1079                 <option value="Australia">Australia</option>
1080             </select><br><br>
1081
1082             <button type="submit">Submit</button>
1083         </form>
1084     </div>
1085 </body>
1086 </html>
1087
1088 //app.js
1089 const express = require('express');
1090 const app = express();
1091 const path = require('path');
1092
1093 app.use(express.static(path.join(__dirname, 'public')));
1094
1095 app.get('/result', (req, res) =>
1096 {
1097     const { username, country } = req.query;
1098     res.render('result.html',
1099     {
1100         username: username,
1101         country: country
1102     });
1103 });app.listen(5000)
1104
1105
1106
1107 //QB-236 MIddleWare Defination and Example
1108
1109
1110 //QB-237 how to link html,css and js file in express js give any short example form it
1111 // as shown above
1112
1113
1114 //QB-238 write a use of bodyparse.urlencoded method in express js
1115 // as shown above
1116
1117
1118 //QB-239 app.use is overloded method or not jystify that

```

```

1119 // as shown above
1120
1121
1122 //QB-240 give example of cookies. create cookies and delete cookies.
1123 // as shown above
1124
1125
1126 //QB-241 how to send HTML and JSON data as a Response using Express js.
1127 // as shown above
1128
1129
1130
1131
1132
1133 //
=====
===== //
1134
1135 //QB-271
1136 // Write an expressJS code in which RESTapi is created for json object named Places I
love which contains name,country,state,city
1137 // 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.
1138 // on localhost:30001/a/10 should display all the places having 10 ratings.
1139 const express = require('express');
1140 const app = express();
1141 const places = [
1142   {
1143     "name": "Golden Gate Bridge",
1144     "country": "USA",
1145     "state": "California",
1146     "city": "San Francisco",
1147     "rating": 9
1148   },
1149   {
1150     "name": "Eiffel Tower",
1151     "country": "France",
1152     "state": "Île-de-France",
1153     "city": "Paris",
1154     "rating": 10
1155   },
1156   {
1157     "name": "Great Barrier Reef",
1158     "country": "Australia",
1159     "state": "Queensland",
1160     "city": "Cairns",
1161     "rating": 10
1162   }
1163 ]
1164
1165 app.get('/places', (req, res) =>
1166 {
1167   res.json(places);
1168 });
1169
1170 app.get('/a/:rating', (req, res) =>
1171 {
1172   const rating = parseInt(req.params.rating);
1173   const filteredPlaces = places.filter(place => place.rating === rating);
1174   res.json(filteredPlaces);
1175 });app.listen(5000)
1176
1177
1178
1179 //QB-272
1180 // write a code to create a link named "Contact Us" using a Pug template engine inside
Express code.
1181 // When you click on "Contact Us" it will redirect to the next page "/"contact" and
display the message "welcome to Contact us page".

```

```

1182
1183 //pug
1184 doctype html
1185 html
1186   head
1187     title Express Pug Example
1188   body
1189     h1 Welcome to Express with Pug
1190     p Click #{a(href='/contact')} to contact us.
1191
1192 //app.js
1193 const express = require('express');
1194 const app = express();
1195
1196 app.set('view engine', 'pug');
1197 app.set('views', './views');
1198
1199 app.get('/', (req, res) =>
1200 {
1201   res.render('index');
1202 });
1203 app.get('/contact', (req, res) =>
1204 {
1205   res.render('contact', { message: 'Welcome to Contact Us page' });
1206 });
1207 app.listen(5000)
1208
1209
1210
1211
1212 //QB-273
1213 //Write an express.js script to send automatic mail to specified user.
1214
1215 const express = require('express');
1216 const app = express();
1217 const nodemailer = require('nodemailer');
1218
1219 app.use(express.json());
1220 app.use(express.urlencoded({ extended: true }));
1221
1222 app.post('/send-email', async (req, res) =>
1223 {
1224   const { to, subject, text } = req.body;
1225   try {
1226     let transporter = nodemailer.createTransport(
1227     {
1228       service: 'gmail',
1229       auth:
1230       {
1231         user: 'your-email@gmail.com', // Your email address
1232         pass: 'your-password'        // Your email password or app-specific
1233         password
1234       }
1235     });
1236
1237     let mailOptions =
1238     {
1239       from: 'your-email@gmail.com',
1240       to: to,
1241       subject: subject,
1242       text: text
1243     };
1244
1245     let info = await transporter.sendMail(mailOptions);
1246     console.log('Email sent:', info.messageId);
1247     res.send('Email sent successfully');
1248   }
1249   catch (error)
1250   {

```

```

1250         console.error('Error sending email:', error);
1251         res.status(500).send('Error sending email');
1252     }
1253 });
1254 app.listen(3000)
1255
1256 //QB-274
1257 //Write an express.js script to upload image.
1258 const express = require('express');
1259 const app = express();
1260 const multer = require('multer');
1261 const path = require('path');
1262
1263 const storage = multer.diskStorage({
1264     {
1265         destination: function(req, file, cb)
1266         {
1267             cb(null, './public/uploads/');
1268         },
1269         filename: function(req, file, cb)
1270         {
1271             cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname));
1272         }
1273     });
1274 const upload = multer({ storage: storage });
1275 app.use(express.static('public'));
1276 app.post('/upload', upload.single('image'), (req, res) =>
1277     {
1278         if (!req.file)
1279         {
1280             return res.status(400).send('No files were uploaded.');
```

```

1318
1319
1320 //QB-276
1321 // Create an Express.js route that accepts file uploads using the multer middleware and
1322 // resizes and saves the uploaded image in multiple sizes
1323 // (e.g., small, medium, large) to a specific directory on the server.
1324
1325
1326 //QB-277
1327 //Write a code to implement file uploading and downloading with Express ?
1328 const express = require('express');
1329 const app = express();
1330 const multer = require('multer');
1331 const path = require('path');
1332 const fs = require('fs');
1333
1334 const storage2 = multer.diskStorage(
1335   {
1336     destination: function(req, file, cb)
1337     {
1338       cb(null, 'public/uploads/');
1339     },
1340     filename: function(req, file, cb)
1341     {
1342       cb(null, Date.now() + '-' + file.originalname);
1343     }
1344   });
1345 const upload2 = multer({ storage: storage2 });
1346 app.use(express.static('public'));
1347
1348 app.post('/upload', upload2.single('file'), (req, res) =>
1349   {
1350     if (!req.file)
1351     {
1352       return res.status(400).send('No files were uploaded.');
```

1353 }

```

1354     res.send('File uploaded successfully');
1355   });
1356
1357 app.get('/download/:filename', (req, res) =>
1358   {
1359     const filename = req.params.filename;
1360     const filePath = path.join(__dirname, 'public/uploads/', filename);
1361
1362     if (fs.existsSync(filePath))
1363     {
1364       res.sendFile(filePath);
1365     }
1366     else
1367     {
1368       res.status(404).send('File not found');
```

1369 }

```

1370   });
1371 app.listen(4000)
1372
1373
1374 //QB-278
1375 //Write express js code to pass data from my Express.js application to a Pug template
1376 const express = require('express');
1377 const app = express();
1378
1379 app.set('view engine', 'pug');
1380 app.set('views', './views');
1381 const data =
1382   {
1383     title: 'Express.js and Pug Example',
1384     message: 'Hello from Express.js and Pug!',
1385     items: ['Apple', 'Banana', 'Cherry']

```

```

1386 };
1387 app.get('/', (req, res) =>
1388 {
1389     res.render('index', data);
1390 });
1391 app.listen(5000)
1392
1393 // .pug
1394 doctype html
1395 html
1396     head
1397         title= title
1398     body
1399         h1= message
1400         ul
1401             each item in items
1402                 li= item
1403
1404
1405 //QB-279
1406 //How to implement search and filtering in a REST API with Express.js
1407 your-app/
1408 |-- app.js
1409 |-- routes/
1410 |   |-- books.js
1411 |-- models/
1412 |   |-- Book.js
1413
1414 // app.js
1415 const express = require('express');
1416 const app = express();
1417 const bodyParser = require('body-parser');
1418 const booksRouter = require('./routes/books');
1419
1420
1421 app.use(bodyParser.json());
1422 app.use('/api/books', booksRouter);
1423
1424
1425 app.use((err, req, res, next) =>
1426 {
1427     console.error(err.stack);
1428     res.status(500).send('Something broke!');
1429 });
1430 app.listen(3000)
1431
1432 // model/book.js
1433 class Book {
1434     constructor(id, title, author, genre, publishedYear) {
1435         this.id = id;
1436         this.title = title;
1437         this.author = author;
1438         this.genre = genre;
1439         this.publishedYear = publishedYear;
1440     }
1441 }
1442 module.exports = Book;
1443
1444 //routes/book.js
1445 const express = require('express');
1446 const app = express();
1447 const router = express.Router();
1448 const Book = require('../models/Book');
1449 const books = [
1450     new Book(1, 'Book 1', 'Author A', 'Fiction', 2000),
1451     new Book(2, 'Book 2', 'Author B', 'Non-fiction', 2010),
1452     new Book(3, 'Book 3', 'Author A', 'Fantasy', 2015),
1453     new Book(4, 'Book 4', 'Author C', 'Science Fiction', 2020)
1454 ];

```

```

1455 router.get('/', (req, res) =>
1456 {
1457     let filteredBooks = [...books];
1458
1459     // Filter by title
1460     if (req.query.title)
1461     {
1462         filteredBooks = filteredBooks.filter(book =>
1463             book.title.toLowerCase().includes(req.query.title.toLowerCase())
1464         );
1465     }
1466
1467     // Filter by author
1468     if (req.query.author)
1469     {
1470         filteredBooks = filteredBooks.filter(book =>
1471             book.author.toLowerCase().includes(req.query.author.toLowerCase())
1472         );
1473     }
1474
1475     // Filter by genre
1476     if (req.query.genre)
1477     {
1478         filteredBooks = filteredBooks.filter(book =>
1479             book.genre.toLowerCase().includes(req.query.genre.toLowerCase())
1480         );
1481     }
1482
1483     // Filter by publishedYear
1484     if (req.query.publishedYear)
1485     {
1486         filteredBooks = filteredBooks.filter(book =>
1487             book.publishedYear === parseInt(req.query.publishedYear)
1488         );
1489     }
1490
1491     res.json(filteredBooks);
1492 });
1493
1494 module.exports = router;
1495
1496
1497
1498 //QB-280
1499 //Write a code to set up Pug in Express.js?
1500
1501 //pug
1502 doctype html
1503 html
1504     head
1505         title= title
1506     body
1507         h1= message
1508
1509 const express = require('express');
1510 const app = express();
1511 const path = require('path');
1512
1513 app.set('views', path.join(__dirname, 'views'));
1514 app.set('view engine', 'pug');
1515 app.get('/', (req, res) =>
1516 {
1517     res.render('index', { title: 'Express with Pug', message: 'Hello Pug!' });
1518 });
1519 app.listen(4000)
1520
1521
1522
1523 //QB-281

```

```

1524 //Write a code to set up nodemailer in Express.js
1525 const express = require('express');
1526 const app = express();
1527 const nodemailer = require('nodemailer');
1528 require('dotenv').config();
1529
1530 app.use(express.json());
1531 app.use(express.urlencoded({ extended: true }));
1532
1533 const transporter = nodemailer.createTransport(
1534   {
1535     host: process.env.SMTP_HOST,
1536     port: process.env.SMTP_PORT,
1537     secure: true, // true for 465, false for other ports
1538     auth:
1539       {
1540         user: process.env.SMTP_USER,
1541         pass: process.env.SMTP_PASS
1542       }
1543   });
1544
1545 app.post('/send-email', (req, res) =>
1546 {
1547   const { to, subject, text } = req.body;
1548   const mailOptions =
1549     {
1550       from: process.env.SMTP_USER,
1551       to: to,
1552       subject: subject,
1553       text: text
1554     };
1555
1556   // Send email
1557   transporter.sendMail(mailOptions, (error, info) =>
1558   {
1559     if (error)
1560     {
1561       console.log(error);
1562       res.status(500).send('Error sending email');
1563     }
1564     else
1565     {
1566       console.log('Email sent: ' + info.response);
1567       res.send('Email sent successfully');
1568     }
1569   });
1570 });
1571 app.listen(4000)
1572
1573
1574 //QB-282
1575 //Write a code to upload file and access uploaded file in express js.
1576 const express = require('express');
1577 const app = express();
1578 const multer = require('multer');
1579 const path = require('path');
1580
1581 const storag = multer.diskStorage({
1582   destination: function (req, file, cb)
1583   {
1584     cb(null, 'uploads/');
1585   },
1586   filename: function (req, file, cb) {
1587     cb(null, Date.now() + '-' + file.originalname); // File name
1588   }
1589 });
1590
1591 // Initialize multer upload
1592 const uploa = multer({ storage: storag });

```



```

1593
1594 // Endpoint to upload a single file
1595 app.post('/upload', uploa.single('file'), (req, res) =>
1596 {
1597     res.send('File uploaded successfully');
1598 });
1599
1600 app.get('/uploads/:filename', (req, res) =>
1601 {
1602     const fileName = req.params.filename;
1603     res.sendFile(path.join(__dirname, 'uploads', fileName));
1604 });
1605 app.listen(4000)
1606
1607
1608
1609 //QB-283
1610 // Write express js script to load student form using pug file which contains following
1611 // fields Name(text) Email(email) Course(radio : CE, IT, CSE)
1612 // Once form submitted then data must be displayed on '/data' page using pug file. Means
1613 // data should be submitted from express application to PUG file
1614
1615 const express = require('express');
1616 const app = express();
1617 const path = require('path');
1618 const bodyParser = require('body-parser');
1619
1620 app.use(bodyParser.json());
1621 app.use(bodyParser.urlencoded({ extended: true }));
1622
1623 app.set('views', path.join(__dirname, 'views'));
1624
1625 app.set('view engine', 'pug');
1626
1627 app.get('/', (req, res) =>
1628 {
1629     res.render('studentForm');
1630 });
1631
1632 app.post('/data', (req, res) =>
1633 {
1634     const { name, email, course } = req.body;
1635     res.render('data', { name, email, course });
1636 });
1637
1638 app.listen(4000)
1639
1640 // .pug
1641 doctype html
1642 html
1643     head
1644         title Student Form
1645     body
1646         h1 Student Form
1647         form(action='/data', method='post')
1648             label(for='name') Name:
1649             input(type='text', id='name', name='name')
1650
1651             br
1652
1653             label(for='email') Email:
1654             input(type='email', id='email', name='email')
1655
1656             br
1657
1658             label Course:
1659             br
1660             input(type='radio', id='ce', name='course', value='CE')

```

```

1660         label(for='ce') CE
1661     br
1662     input(type='radio', id='it', name='course', value='IT')
1663     label(for='it') IT
1664     br
1665     input(type='radio', id='cse', name='course', value='CSE')
1666     label(for='cse') CSE
1667
1668     br
1669
1670     button(type='submit') Submit
1671
1672 //view.pug
1673 doctype html
1674 html
1675     head
1676         title Student Data
1677     body
1678         h1 Student Data
1679         ul
1680             li Name: #{name}
1681             li Email: #{email}
1682             li Course: #{course}
1683
1684
1685
1686
1687 //QB-284
1688 // Write an express js script that allows only image type file to be uploaded using the
1689 // "multer" middleware and saves the file to the specific directory called
1690 // "IMAGES". If file other than image has been uploaded then it will give an error
1691 // message that "Upload only image file".
1692
1693 //html
1694 <!-- index.html -->
1695 <!DOCTYPE html>
1696 <html lang="en">
1697 <head>
1698     <meta charset="UTF-8">
1699     <meta name="viewport" content="width=device-width, initial-scale=1.0">
1700     <title>Image Upload</title>
1701 </head>
1702 <body>
1703     <h1>Image Upload</h1>
1704     <form action="http://localhost:3000/upload" method="POST" enctype=
1705     "multipart/form-data">
1706         <input type="file" name="image">
1707         <button type="submit">Upload</button>
1708     </form>
1709 </body>
1710 </html>
1711
1712 //app.js
1713 const express = require('express')
1714 const app = express();
1715 const multer = require('multer');
1716 const path = require('path');
1717
1718 // Multer storage configuration
1719 const storagea = multer.diskStorage({
1720     destination: function (req, file, cb)
1721     {
1722         cb(null, 'IMAGES/');
1723     },
1724     filename: function (req, file, cb)
1725     {
1726         cb(null, Date.now() + '-' + file.originalname);
1727     }
1728 });

```

```

1726 });
1727
1728 const fileFilter = (req, file, cb) =>
1729 {
1730   if (file.mimetype.startsWith('image'))
1731   {
1732     cb(null, true);
1733   }
1734   else
1735   {
1736     cb(new Error('Upload only image file'), false);
1737   }
1738 };
1739
1740 const uploada = multer({
1741   storage: storagea,
1742   fileFilter: fileFilter
1743 });
1744
1745 app.post('/upload', uploada.single('image'), (req, res, next) =>
1746 {
1747   if (!req.file)
1748   {
1749     return res.status(400).send('No file uploaded');
1750   }
1751   res.send('File uploaded successfully');
1752 });
1753
1754 app.use((err, req, res, next) =>
1755 {
1756   if (err instanceof multer.MulterError)
1757   {
1758     res.status(400).send('Error uploading file: ' + err.message);
1759   }
1760   else
1761   {
1762     res.status(500).send('Server error: ' + err.message);
1763   }
1764 });
1765 app.listen(3000)
1766
1767
1768
1769 //QB-285
1770 // Write an express JS script to upload any type of file of size up to 50KB only.
1771 const express = require('express');
1772 const app = express();
1773 const multer = require('multer');
1774 const path = require('path');
1775
1776 const storageb = multer.diskStorage({
1777   destination: function (req, file, cb)
1778   {
1779     cb(null, 'uploads/');
1780   },
1781   filename: function (req, file, cb)
1782   {
1783     cb(null, Date.now() + '-' + file.originalname);
1784   }
1785 });
1786
1787 const fileSizeLimit = 50 * 1024; // 50 KB
1788
1789 const uploadb = multer({
1790   storage: storageb,
1791   limits:
1792   {
1793     fileSize: fileSizeLimit
1794   },

```

```

1795     fileFilter: function (req, file, cb)
1796     {
1797
1798         cb(null, true);
1799     }
1800 });
1801
1802 app.post('/upload', upload.single('file'), (req, res) =>
1803 {
1804     if (!req.file)
1805     {
1806         return res.status(400).send('No file uploaded');
1807     }
1808     res.send('File uploaded successfully');
1809 });
1810
1811 app.use((err, req, res, next) =>
1812 {
1813     if (err instanceof multer.MulterError)
1814     {
1815         res.status(400).send('Error uploading file: ' + err.message);
1816     }
1817     else
1818     {
1819         res.status(500).send('Server error: ' + err.message);
1820     }
1821 });
1822
1823 app.listen(3000)
1824
1825
1826 //QB-286
1827 // write a code to create a link named "About Us" using a Pug template engine inside
1828 // Express code when you click on
1829 // "About Us" it will redirect to the next page "/about" and display the message
1830 // "welcome to About us page".
1831
1832 // .pug
1833 doctype html
1834 html
1835     head
1836         title Home Page
1837     body
1838         h1 Welcome to Home Page
1839         a(href='/about') About Us
1840
1841 //view.pug
1842 doctype html
1843 html
1844     head
1845         title About Us Page
1846     body
1847         h1 Welcome to About Us Page
1848
1849 const express = require('express');
1850 const app = express();
1851 const path = require('path');
1852
1853 app.set('views', path.join(__dirname, 'views'));
1854 app.set('view engine', 'pug');
1855
1856 app.get('/', (req, res) =>
1857 {
1858     res.render('index');
1859 });
1860
1861 app.get('/about', (req, res) =>
1862 {

```

```

1862     res.render('about');
1863 });
1864
1865 app.listen(3000)
1866
1867
1868
1869
1870 //QB-287
1871 // Perform the following tasks as asked:
1872 // 1) Create a HTML file for response form and this file should be loaded on home('/')
page. •Fields are : Name, Email and Submit button.
1873 // 2) Once Response is submitted, message "Thank you for your response." Will be
displayed on page '/response' and
1874 // also send mail to the entered email id with the submitted response.
1875
1876 //html
1877 <!DOCTYPE html>
1878 <html lang="en">
1879 <head>
1880     <meta charset="UTF-8">
1881     <meta name="viewport" content="width=device-width, initial-scale=1.0">
1882     <title>Response Form</title>
1883 </head>
1884 <body>
1885     <h1>Response Form</h1>
1886     <form action="/response" method="POST">
1887         <label for="name">Name:</label>
1888         <input type="text" id="name" name="name" required<br><br>
1889
1890         <label for="email">Email:</label>
1891         <input type="email" id="email" name="email" required<br><br>
1892
1893         <button type="submit">Submit</button>
1894     </form>
1895 </body>
1896 </html>
1897
1898 //app.js
1899 const express = require('express');
1900 const app = express();
1901 const bodyParser = require('body-parser');
1902 const nodemailer = require('nodemailer');
1903 const path = require('path');
1904
1905 app.use(bodyParser.urlencoded({ extended: false }));
1906 app.use(bodyParser.json());
1907 app.use(express.static(path.join(__dirname, 'public')));
1908
1909 app.get('/', (req, res) =>
1910 {
1911     res.sendFile(path.join(__dirname, 'public', 'index.html'));
1912 });
1913 app.post('/response', (req, res) =>
1914 {
1915     const { name, email } = req.body;
1916     sendEmail(email);
1917     res.send('Thank you for your response.');
```

```

1929     {
1930         from: 'your_email@gmail.com',
1931         to: email,
1932         subject: 'Thank you for your response',
1933         text: 'Thank you for submitting your response.'
1934     };
1935     transporter.sendMail(mailOptions, (error, info) =>
1936     {
1937         if (error)
1938         {
1939             console.log('Error occurred while sending email:', error.message);
1940         }
1941         else
1942         {
1943             console.log('Email sent:', info.response);
1944         }
1945     });
1946 }
1947 app.listen(3000)
1948
1949
1950
1951
1952 //QB-288
1953 // Write a code to set up nodemailer in Express.js.
1954 // Sender email id: "lju@gmail.com".
1955 // Receiver email ids: "student@gmail.com and faculty@gmail.com". Mail subject should be
1956 "LJ University"
1957 // Mail body contains "Welcome Student" in h3 tag and in table display data Date
1958 28/06/23, Exam name FSD-2.
1959 const express = require('express');
1960 const app = express();
1961 const nodemailer = require('nodemailer');
1962
1963 const transporterr = nodemailer.createTransport({
1964     service: 'gmail',
1965     auth:
1966     {
1967         user: 'lju@gmail.com', // Sender email id
1968         pass: 'your_password' // Replace with your email password
1969     }
1970 });
1971
1972 app.get('/sendEmail', (req, res) =>
1973 {
1974     const receivers = ['student@gmail.com', 'faculty@gmail.com'];
1975
1976     const mailOptions =
1977     {
1978         from: 'lju@gmail.com',
1979         to: receivers.join(', '), // Join receiver emails with comma
1980         subject: 'LJ University',
1981         html: `
1982             <h3>Welcome Student</h3>
1983             <table border="1">
1984                 <tr><td>Date</td><td>28/06/23</td></tr>
1985                 <tr><td>Exam name</td><td>FSD-2</td></tr>
1986             </table>
1987         `
1988     };
1989
1990     transporter.sendMail(mailOptions, (error, info) =>
1991     {
1992         if (error)
1993         {
1994             console.log('Error occurred while sending email:', error.message);
1995             res.send('Error occurred while sending email');
1996         } else {

```

```

1996         console.log('Email sent:', info.response);
1997         res.send('Email sent successfully');
1998     }
1999 });
2000 });
2001 app.listen(3000)
2002
2003
2004 //QB-289
2005 // Write an express js script to configure the multer middleware. Perform following
2006 // tasks.
2007 // 1) Create a pug file named "file.pug". This file contains heading(h3) "Upload your
2008 // CV" in red color. And,
2009 // a form with input type file(to browse and select file) and submit(to upload the
2010 // file).
2011 // 2) Create a js file named "file.js" and link this js and pug file to browse pug file
2012 // on "/home" page.
2013 // 3) After uploading a file display message on "/upload" page "(file original name) has
2014 // been uploaded".
2015 // 4) Save uploaded files to specific directory named "upload". And in this folder file
2016 // must be stored in format of "lju-file.pdf" where "lju" is the field name.
2017
2018 const express = require('express');
2019 const app = express();
2020 const path = require('path');
2021 const multer = require('multer');
2022
2023 app.set('views', path.join(__dirname, 'public'));
2024 app.set('view engine', 'pug');
2025
2026 app.use(express.static(path.join(__dirname, 'public')));
2027
2028 const storagec = multer.diskStorage({
2029     destination: function (req, file, cb)
2030     {
2031         cb(null, 'uploads/'); // Directory to save uploaded files
2032     },
2033     filename: function (req, file, cb)
2034     {
2035         cb(null, 'lju-' + file.originalname); // File naming format
2036     }
2037 });
2038
2039 const uploadc = multer({ storage: storagec });
2040
2041 app.get('/home', (req, res) =>
2042 {
2043     res.render('file');
2044 });
2045
2046 app.post('/upload', uploadc.single('lju'), (req, res) =>
2047 {
2048     if (!req.file)
2049     {
2050         return res.status(400).send('No files were uploaded.');

```

```

2059     }
2060     body
2061         h3 Upload your CV
2062         form(action='/upload', method='post', enctype='multipart/form-data')
2063             input(type='file', name='lju')
2064             br
2065             button(type='submit') Upload
2066
2067
2068
2069 //QB-290
2070 //Write express js script to upload file with size limit of 1 MB to a specific directory
named "Data" on the server.
2071
2072 const express = require('express');
2073 const app = express();
2074 const multer = require('multer');
2075 const path = require('path');
2076
2077
2078 const storaged = multer.diskStorage({
2079     destination: function (req, file, cb)
2080     {
2081         cb(null, 'uploads/Data/'); // Directory to save uploaded files
2082     },
2083     filename: function (req, file, cb)
2084     {
2085         cb(null, file.originalname); // Use original filename
2086     }
2087 });
2088
2089 const upload = multer({
2090     storage: storaged,
2091     limits: { fileSize: 1000000 } // 1 MB file size limit
2092 });
2093
2094 app.post('/upload', upload.single('file'), (req, res) =>
2095 {
2096     if (!req.file)
2097     {
2098         return res.status(400).send('No file uploaded.');

```



```
2122 <body>
2123   <div class="container">
2124     <h3 class="heading">Upload your File</h3>
2125     <form action="/upload" method="post" enctype="multipart/form-data">
2126       <input type="file" name="data">
2127       <br>
2128       <button type="submit">Upload</button>
2129     </form>
2130   </div>
2131 </body>
2132 </html>
2133
2134
2135 const express = require('express');
2136 const app = express();
2137 const path = require('path');
2138 const multer = require('multer');
2139
2140 app.use(express.static(path.join(__dirname, 'public')));
2141
2142 const storage = multer.diskStorage({
2143   destination: function (req, file, cb)
2144     {
2145       cb(null, 'uploads/File/');
2146     },
2147   filename: function (req, file, cb) {
2148     cb(null, 'data-' + file.originalname);
2149   }
2150 });
2151
2152 const upload = multer({ storage: storage });
2153
2154 app.get('/home', (req, res) =>
2155   {
2156     res.sendFile(path.join(__dirname, 'public', 'form.html'));
2157   });
2158
2159 app.post('/upload', upload.single('data'), (req, res) =>
2160   {
2161     if (!req.file)
2162     {
2163       return res.status(400).send('No file uploaded.');
```