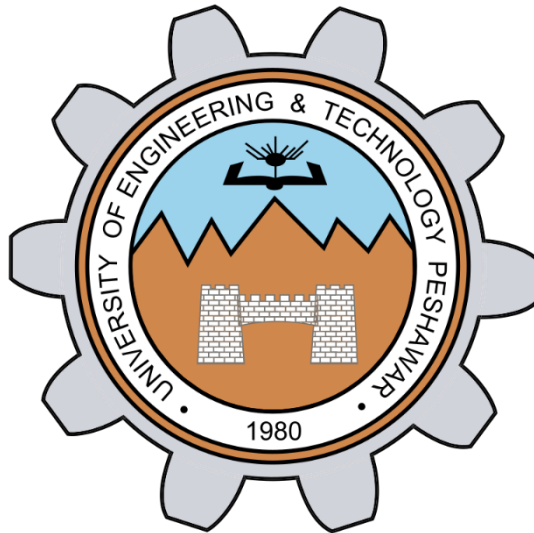


UNIVERSITY OF ENGINEERING AND TECHNOLOGY, PESHAWAR  
PAKISTAN  
*Main Campus*



**CS-311 Web Programming**  
**Practical Task 07**

**Submitted By**

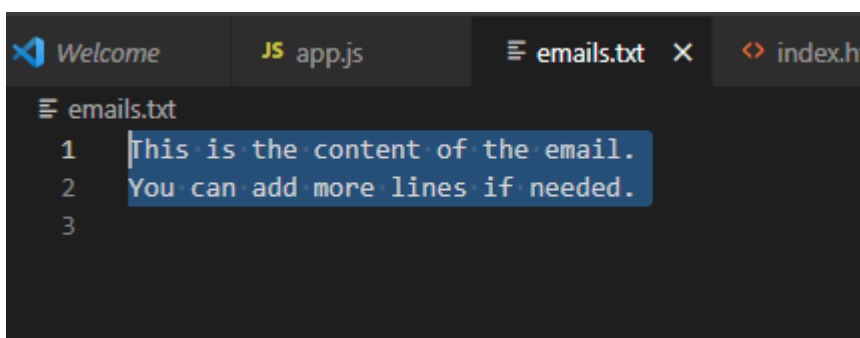
<b>Name:</b>	<b>Sahibzada Abdullah ahmad</b>
<b>Registration No.</b>	<b>21pwbc0861</b>
<b>Semester:</b>	<b>BS CS 5<sup>th</sup></b>
<b>Section:</b>	<b>B</b>
<b>Submission Date:</b>	<b>27th November 2023</b>

**Submitted To**  
**Sir Mohammad**

## • File Handling:

- o Create a text file named emails.txt with the email content.
- o Use Node.js to read the contents of emails.txt.

```
JS app.js > app.get('/send-email') callback > transporter
1 // Import required modules
2 const fs = require('fs');
3 const express = require('express');
4 const nodemailer = require('nodemailer');
5
6 // Create an Express app
7 const app = express();
8 const port = 3000;
9
10 // Read the contents of emails.txt
11 fs.readFile('emails.txt', 'utf8', (err, data) => {
12     if (err) {
13         console.error('Error reading emails.txt:', err);
14         return;
15     }
16
17     // Store the email content for later use
18     const emailContent = data;
19 });
```



The screenshot shows a code editor with three tabs: 'Welcome', 'JS app.js', and 'emails.txt'. The 'emails.txt' tab is active, displaying the following text:

```
1 This is the content of the email.
2 You can add more lines if needed.
3
```

## • Express Server:

- o Set up an Express server that serves a simple HTML page.
- o Display the contents of emails.txt on the webpage.
- o Use a route like /emails to serve the HTML page

```
// Set up a route to serve the HTML page
app.get('/emails', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});

// Set up a route to handle the "Send Email" button click
app.get('/send-email', (req, res) => {
  // Create a nodemailer transporter
  const transporter = nodemailer.createTransport({
    service: 'gmail',
    auth: {
      user: 'sahibzadaabdullah8@gmail.com', // your Gmail address
      pass: 's@h!b0099', // your Gmail password
    },
  });

  // Define email options
  const mailOptions = {
    from: 'sahibzadaabdullah8@gmail.com', // sender address
    to: '21pwbc0861@uetpeshawar.edu.pk', // recipient address
```

```
    subject: 'Read data from the Text file', // subject line
    text: 'This is the content of the email. You can add more lines if needed.', // email body
  };

  // Send the email
  transporter.sendMail(mailOptions, (error, info) => {
    if (error) {
      console.error('Error sending email:', error);
      res.send('Error sending email.');
```

```
    } else {
      console.log('Email sent:', info.response);
      res.send('Email sent successfully.');
```

```
    }
  });

  // Start the Express server
  app.listen(port, () => {
    console.log(`Server is running at http://localhost:${port}/emails`);
  });
}
```

## • Send Email Button:

- o Add a button on the webpage labeled "Send Email."
- o When the button is clicked, use Node.js to read the contents of emails.txt and send an email to a predefined address.

o The email subject should be "Read data from the Text file," and the body should contain the content from emails.txt.

---

## Email Content

Send Email

### • Notification:

o After sending the email, display a notification on the webpage indicating that the email has been sent successfully.

## Email Content

Send Email

Error sending email.

### • Error Handling:

o Implement error handling for file reading and email sending.  
o Display error messages on the webpage when an error occurs.

```
be: readable.push((node.internal?/src/asmjs/readable.js:26:116) {
code: 'EAUTH',
response: '535-5.7.8 Username and Password not accepted. Learn more at\n' +
'535 5.7.8  https://support.google.com/mail/?p=BadCredentials u4-20020a17090282c400b001bb1f0605b2sm8017242plz.214 - gmail',
responseCode: 535,
command: 'AUTH PLAIN'
```