****Backend (Node.js and Express)**
Setup a new Node.js project**

mkdir csv-upload-app
cd csv-upload-app
npm init -y
npm install express multer csv-parser mongoose

****Create the server (server.js)****

```
const express = require('express');
const multer = require('multer');
const csv = require('csv-parser');
const mongoose = require('mongoose');
const fs = require('fs');

const app = express();
const PORT = 5000;

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/csvdata', { useNewUrlParser: true,
useUnifiedTopology: true });

const dataSchema = new mongoose.Schema({
   CreditScore: Number,
   CreditLines: Number,
});

const Data = mongoose.model('Data', dataSchema);

// Multer setup for file upload
const upload = multer({ dest: 'uploads/' });

app.use(express.json());

// Route for uploading CSV
app.post('/upload', upload.single('file'), (req, res) => {
   const results = [];
   fs.createReadStream(req.file.path)
      .pipe(csv())
      .on('data', (data) => results.push(data))
      .on('end', () => {
```

```
            Data.insertMany(results)
                .then(() => {
                    fs.unlinkSync(req.file.path); // Remove file after processing
                    res.json({ message: 'File uploaded and data saved!' });
                })
                .catch((err) => res.status(500).json({ error: err.message }));
        });
});

// Route for fetching data with pagination
app.get('/data', async (req, res) => {
    const { page = 1, limit = 10 } = req.query;
    try {
        const data = await Data.find()
            .limit(limit * 1)
            .skip((page - 1) * limit)
            .exec();
        const count = await Data.countDocuments();
        res.json({
            data,
            totalPages: Math.ceil(count / limit),
            currentPage: page
        });
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

// Route for calculating subscription pricing
app.post('/calculate', async (req, res) => {
    const { basePrice, pricePerCreditLine, pricePerCreditScorePoint } = req.body;
    const data = await Data.find();
    const results = data.map(entry => {
        const subscriptionPrice = basePrice + (pricePerCreditLine * entry.CreditLines) +
(pricePerCreditScorePoint * entry.CreditScore);
        return { ...entry._doc, subscriptionPrice };
    });
    res.json(results);
});

app.listen(PORT, () => {
    console.log(`Server is running on http://localhost:${PORT}`);
});
```

****Frontend (React)**
Setup a new React project**

npx create-react-app csv-upload-app-client
cd csv-upload-app-client
npm install axios react-bootstrap bootstrap

****Update src/App.js****

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import { Container, Row, Col, Button, Form, Table, Pagination } from 'react-bootstrap';
import 'bootstrap/dist/css/bootstrap.min.css';

function App() {
  const [file, setFile] = useState(null);
  const [data, setData] = useState([]);
  const [page, setPage] = useState(1);
  const [totalPages, setTotalPages] = useState(1);
  const [basePrice, setBasePrice] = useState(100);
  const [pricePerCreditLine, setPricePerCreditLine] = useState(10);
  const [pricePerCreditScorePoint, setPricePerCreditScorePoint] = useState(0.5);
  const [calculatedData, setCalculatedData] = useState([]);

  useEffect(() => {
    fetchData();
  }, [page]);

  const fetchData = async () => {
    const response = await axios.get(`http://localhost:5000/data?page=${page}&limit=10`);
    setData(response.data.data);
    setTotalPages(response.data.totalPages);
  };

  const handleFileChange = (e) => {
    setFile(e.target.files[0]);
  };
```

```
    const handleUpload = async () => {
        const formData = new FormData();
        formData.append('file', file);
        await axios.post('http://localhost:5000/upload', formData, {
            onUploadProgress: progressEvent => {
                console.log('Upload Progress: ' + Math.round((progressEvent.loaded /
progressEvent.total) * 100) + '%');
            }
        });
        fetchData();
    };

    const handleCalculate = async () => {
        const response = await axios.post('http://localhost:5000/calculate', {
            basePrice,
            pricePerCreditLine,
            pricePerCreditScorePoint
        });
        setCalculatedData(response.data);
    };

    return (
        <Container>
            <Row className="my-4">
                <Col>
                    <Form>
                        <Form.Group>
                            <Form.File label="Upload CSV File" onChange={handleFileChange} />
                        </Form.Group>
                        <Button onClick={handleUpload}>Upload</Button>
                    </Form>
                </Col>
            </Row>
            <Row className="my-4">
                <Col>
                    <h3>Data</h3>
                    <Table striped bordered hover>
                        <thead>
                            <tr>
                                <th>Credit Score</th>
                                <th>Credit Lines</th>
                            </tr>
                        </thead>
                        <tbody>
```

```
                    {data.map((row, index) => (
                        <tr key={index}>
                            <td>{row.CreditScore}</td>
                            <td>{row.CreditLines}</td>
                        </tr>
                    ))}
                </tbody>
            </Table>
            <Pagination>
                {[...Array(totalPages).keys()].map(number => (
                    <Pagination.Item key={number + 1} active={number + 1 === page} onClick={()
=> setPage(number + 1)}>
                        {number + 1}
                    </Pagination.Item>
                ))}
            </Pagination>
        </Col>
    </Row>
    <Row className="my-4">
        <Col>
            <h3>Subscription Pricing Calculator</h3>
            <Form>
                <Form.Group>
                    <Form.Label>Base Price</Form.Label>
                    <Form.Control type="number" value={basePrice} onChange={(e) =>
setBasePrice(e.target.value)} />
                </Form.Group>
                <Form.Group>
                    <Form.Label>Price Per Credit Line</Form.Label>
                    <Form.Control type="number" value={pricePerCreditLine} onChange={(e) =>
setPricePerCreditLine(e.target.value)} />
                </Form.Group>
                <Form.Group>
                    <Form.Label>Price Per Credit Score Point</Form.Label>
                    <Form.Control type="number" value={pricePerCreditScorePoint}
onChange={(e) => setPricePerCreditScorePoint(e.target.value)} />
                </Form.Group>
                <Button onClick={handleCalculate}>Calculate</Button>
            </Form>
        </Col>
    </Row>
    <Row className="my-4">
        <Col>
            <h3>Calculated Subscription Prices</h3>
```

```
                <Table striped bordered hover>
                    <thead>
                        <tr>
                            <th>Credit Score</th>
                            <th>Credit Lines</th>
                            <th>Subscription Price</th>
                        </tr>
                    </thead>
                    <tbody>
                        {calculatedData.map((row, index) => (
                            <tr key={index}>
                                <td>{row.CreditScore}</td>
                                <td>{row.CreditLines}</td>
                                <td>{row.subscriptionPrice}</td>
                            </tr>
                        ))}
                    </tbody>
                </Table>
            </Col>
        </Row>
    </Container>
  );
}

export default App;
```

****Update src/index.js to import Bootstrap CSS****

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import 'bootstrap/dist/css/bootstrap.min.css';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

****Running the Application****

1) Start the Backend:
   node server.js
2) Start the Frontend:
   npm start