Final task Report

1. **Project Name** - Create an Expenditure Calculator using MERN stack.

2. **Group Information –**

Name	Branch	GR no.
Ketan Gangwal	AI & DS	12010720
Nayan Bari	ENTC	12010451
Pavan Sanjay	Computer Science	12010017

3. **Project Github Link** – https://github.com/RoboSpark-2021/robospark-2021-Expenditure_Calculator.git

4. **Project Workflow** –
   - Finalizing Design of Frontend using Figma
   - Front Development using React js
   - Testing Frontend
   - Creating a Server
   - Creatin
   - Connecting React js with MongoDb Atlas (Cloud)
   - Creating API or routes for storing and fetching the data
   - Connecting the Frontend Form with Backend using API

- Showing Expenditure History in History Tab via calling the API
- Testing Whole Project (Frontend + Backend)

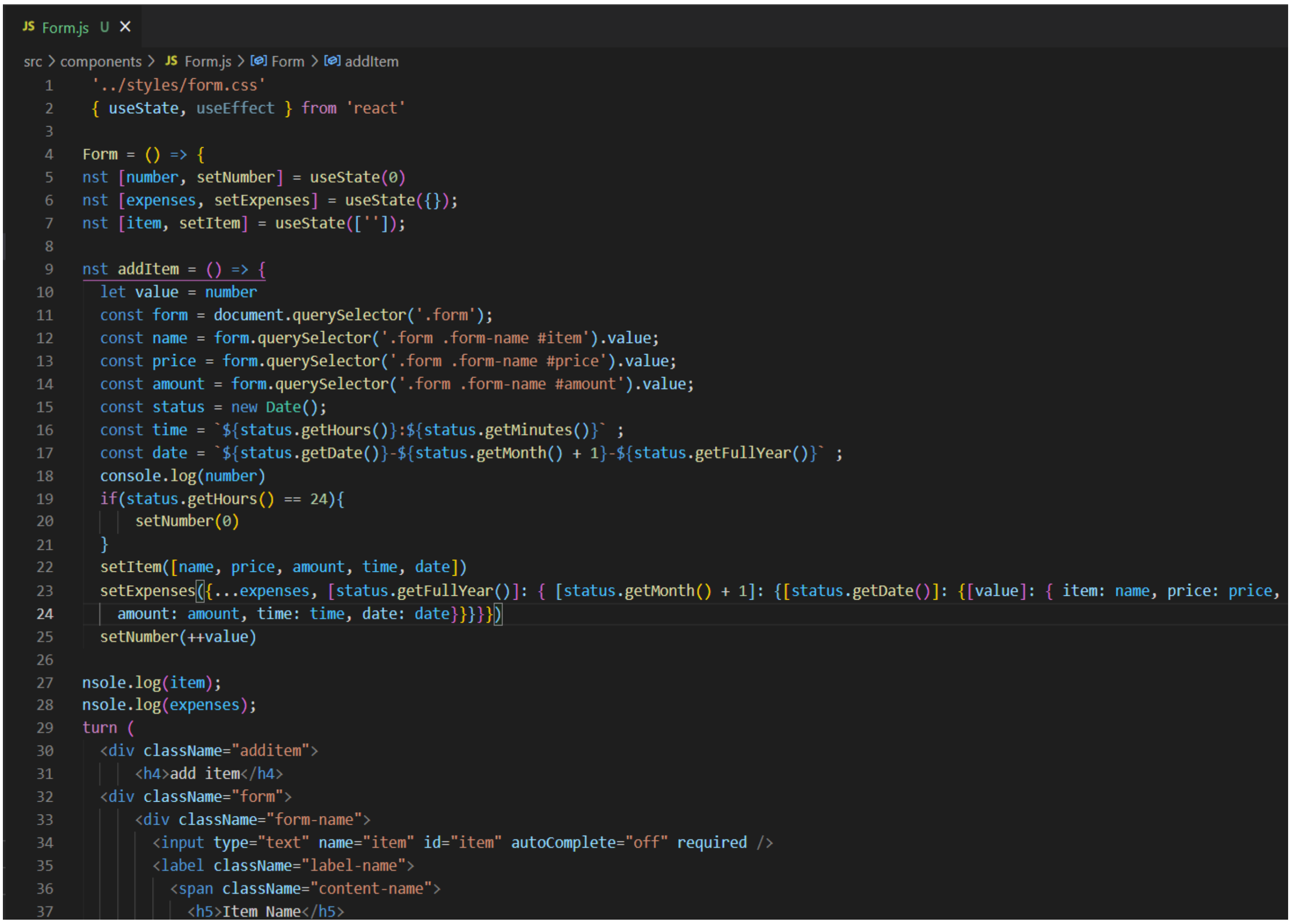
5. **Problems Faced** –
   - Problem facing while appending the Data in react state (frontend)
   - Connecting Server to the MongoDB Cluster
   - Repetition of Today's Expenditure Data while showing it in the Frontend

6. **Alternative Solutions found for problems mentioned above** –
   - Instead of Storing the Data in React state, we directly showed the Data using API Call
   - Instead of using Mongoose Package, we used MongoDB Package
   -

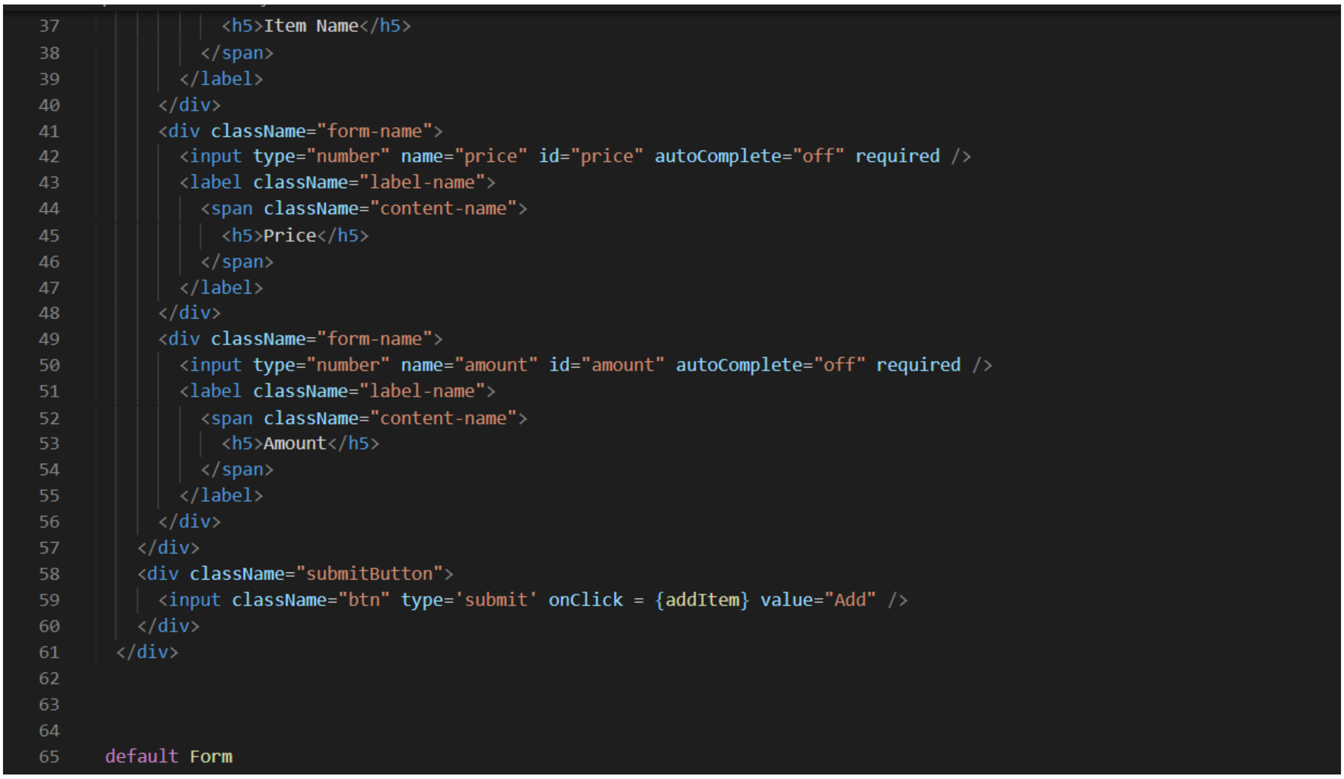
7. **Code Snippet ss** –

Frontend –

A. Form.js - -

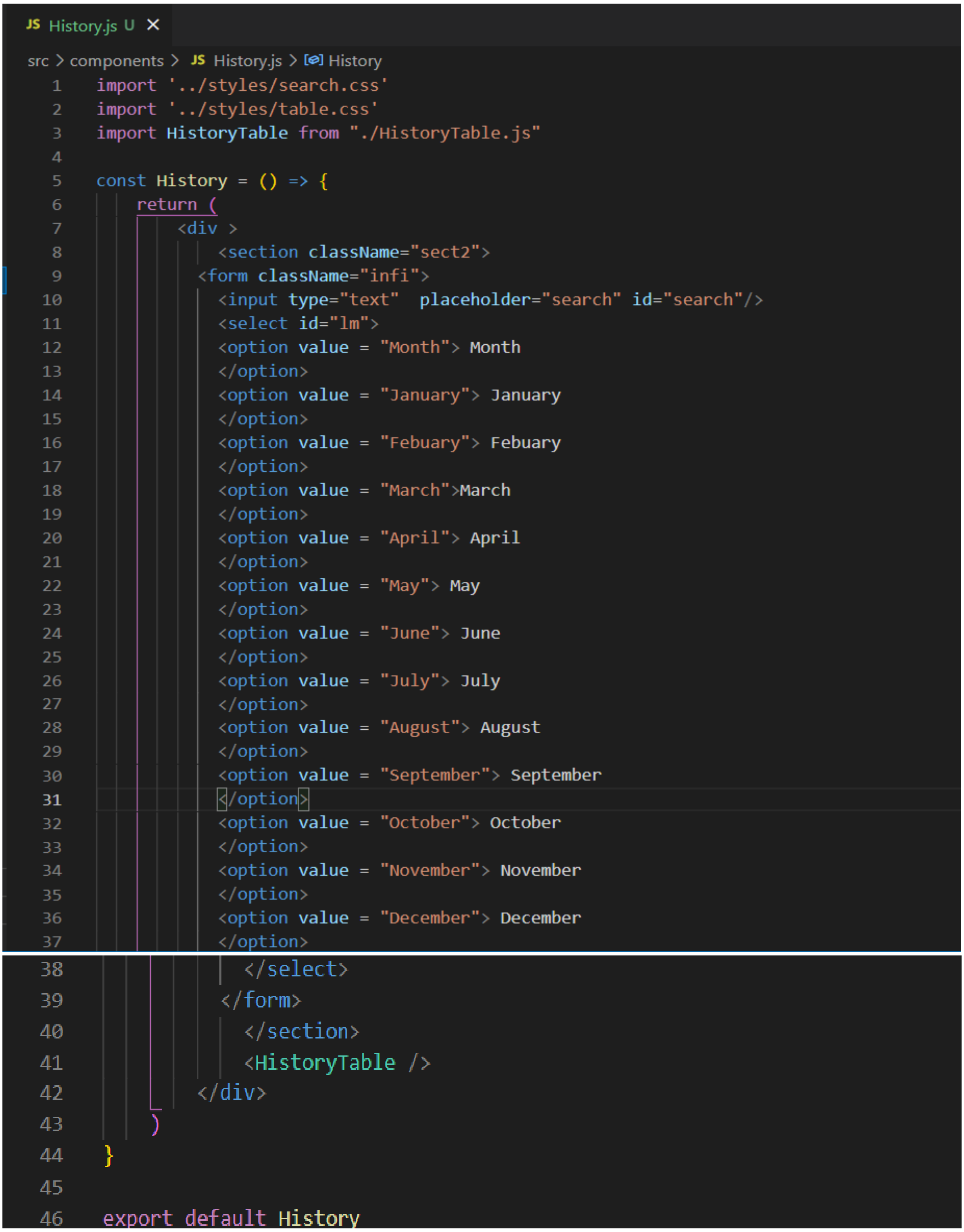


```
 '../styles/form.css'
 { useState, useEffect } from 'react'

Form = () => {
nst [number, setNumber] = useState(0)
nst [expenses, setExpenses] = useState({});
nst [item, setItem] = useState(['']);

nst addItem = () => {
  let value = number
  const form = document.querySelector('.form');
  const name = form.querySelector('.form .form-name #item').value;
  const price = form.querySelector('.form .form-name #price').value;
  const amount = form.querySelector('.form .form-name #amount').value;
  const status = new Date();
  const time = `${status.getHours()}:${status.getMinutes()}` ;
  const date = `${status.getDate()}-${status.getMonth() + 1}-${status.getFullYear()}` ;
  console.log(number)
  if(status.getHours() == 24){
      setNumber(0)
  }
  setItem([name, price, amount, time, date])
  setExpenses({...expenses, [status.getFullYear()]: { [status.getMonth() + 1]: {[status.getDate()]: {[value]: { item: name, price: price,
    amount: amount, time: time, date: date}}}}})
  setNumber(++value)

nsole.log(item);
nsole.log(expenses);
turn (
  <div className="additem">
      <h4>add item</h4>
  <div className="form">
      <div className="form-name">
        <input type="text" name="item" id="item" autoComplete="off" required />
        <label className="label-name">
          <span className="content-name">
            <h5>Item Name</h5>
```



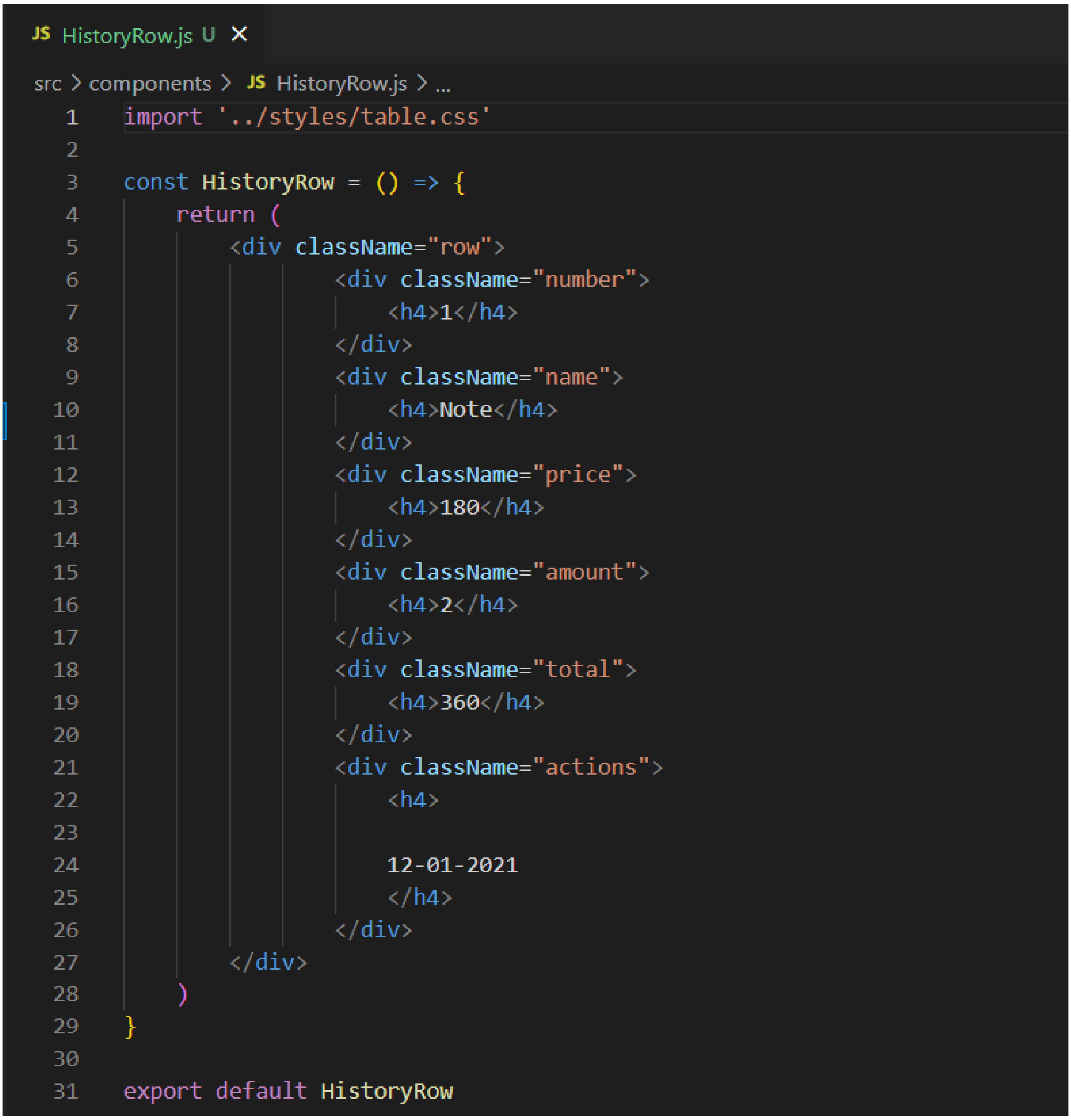
```
            <h5>Item Name</h5>
          </span>
        </label>
      </div>
      <div className="form-name">
        <input type="number" name="price" id="price" autoComplete="off" required />
        <label className="label-name">
          <span className="content-name">
            <h5>Price</h5>
          </span>
        </label>
      </div>
      <div className="form-name">
        <input type="number" name="amount" id="amount" autoComplete="off" required />
        <label className="label-name">
          <span className="content-name">
            <h5>Amount</h5>
          </span>
        </label>
      </div>
    </div>
    <div className="submitButton">
      <input className="btn" type='submit' onClick = {addItem} value="Add" />
    </div>
  </div>



default Form
```

B. History.js –



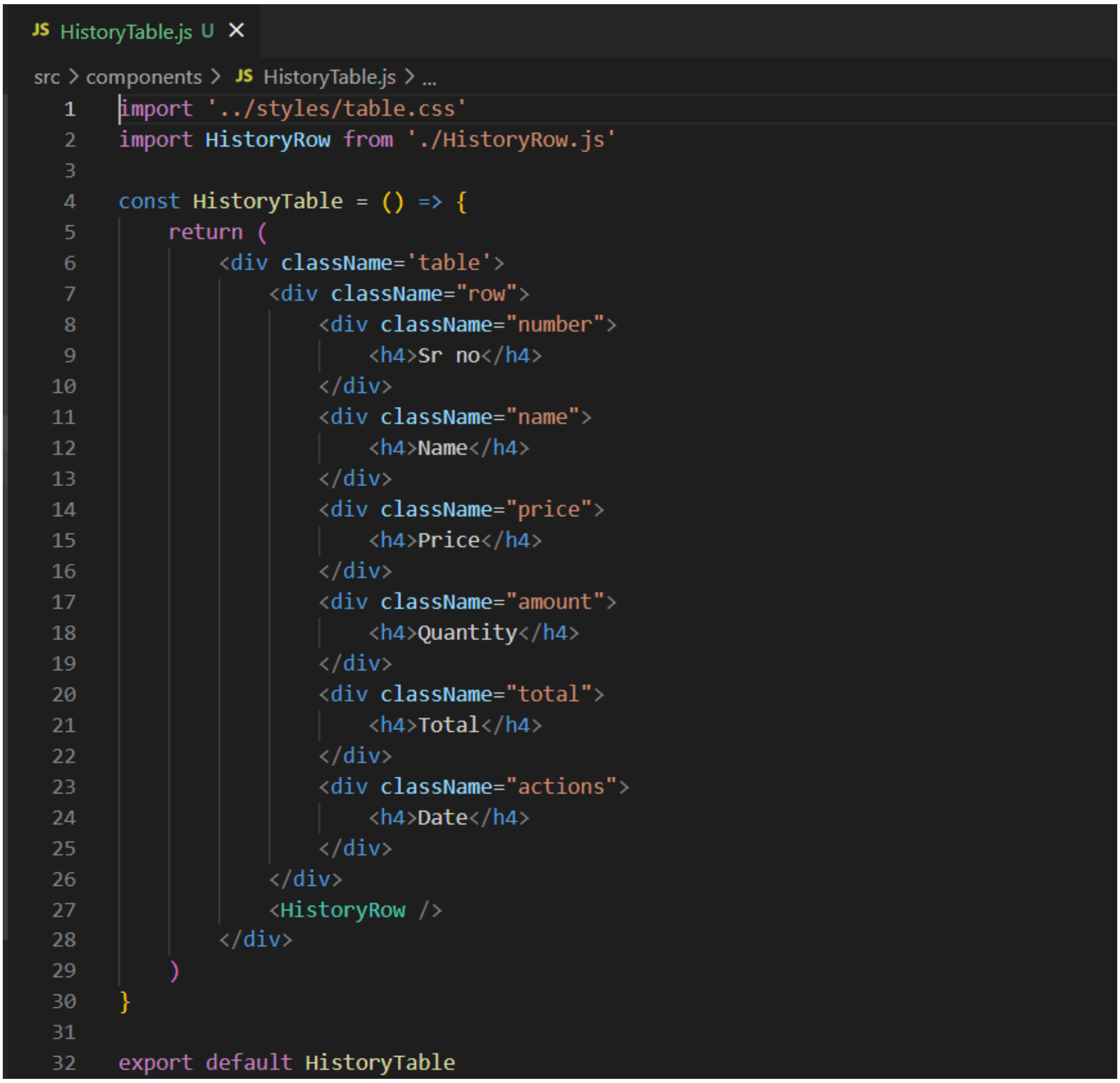
```
import '../styles/search.css'
import '../styles/table.css'
import HistoryTable from "./HistoryTable.js"

const History = () => {
    return (
        <div >
            <section className="sect2">
          <form className="infi">
            <input type="text"  placeholder="search" id="search"/>
            <select id="lm">
            <option value = "Month"> Month
            </option>
            <option value = "January"> January
            </option>
            <option value = "Febuary"> Febuary
            </option>
            <option value = "March">March
            </option>
            <option value = "April"> April
            </option>
            <option value = "May"> May
            </option>
            <option value = "June"> June
            </option>
            <option value = "July"> July
            </option>
            <option value = "August"> August
            </option>
            <option value = "September"> September
            </option>
            <option value = "October"> October
            </option>
            <option value = "November"> November
            </option>
            <option value = "December"> December
            </option>
            </select>
          </form>
            </section>
            <HistoryTable />
        </div>
    )
}

export default History
```

C. HistoryRow.js –



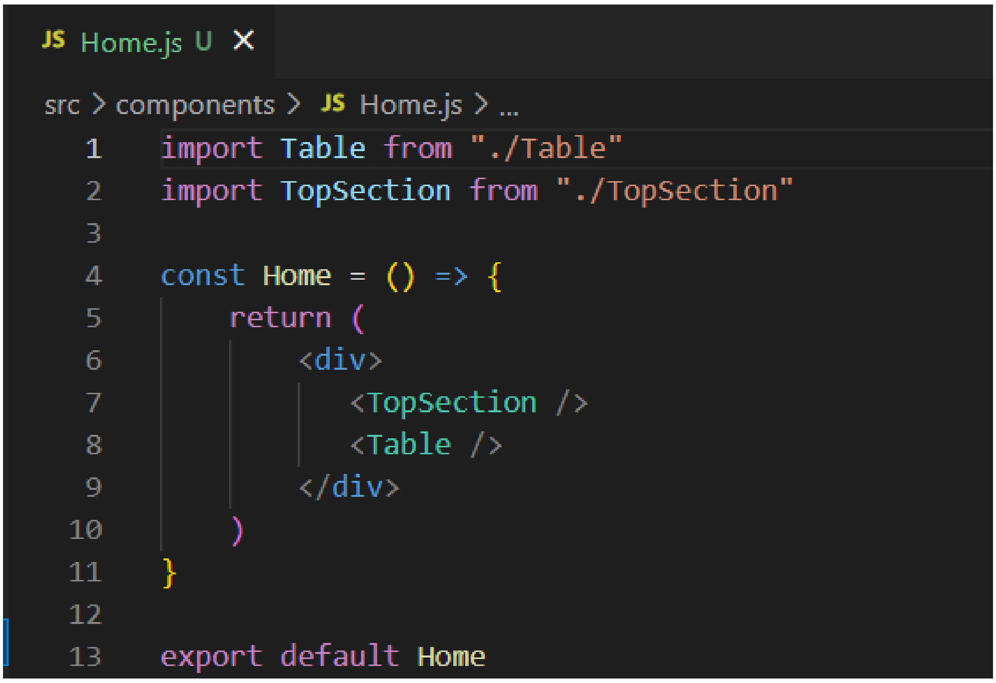
```jsx
import '../styles/table.css'

const HistoryRow = () => {
    return (
        <div className="row">
                <div className="number">
                    <h4>1</h4>
                </div>
                <div className="name">
                    <h4>Note</h4>
                </div>
                <div className="price">
                    <h4>180</h4>
                </div>
                <div className="amount">
                    <h4>2</h4>
                </div>
                <div className="total">
                    <h4>360</h4>
                </div>
                <div className="actions">
                    <h4>

                    12-01-2021
                    </h4>
                </div>
        </div>
    )
}

export default HistoryRow
```

D. HistoryTable.js –



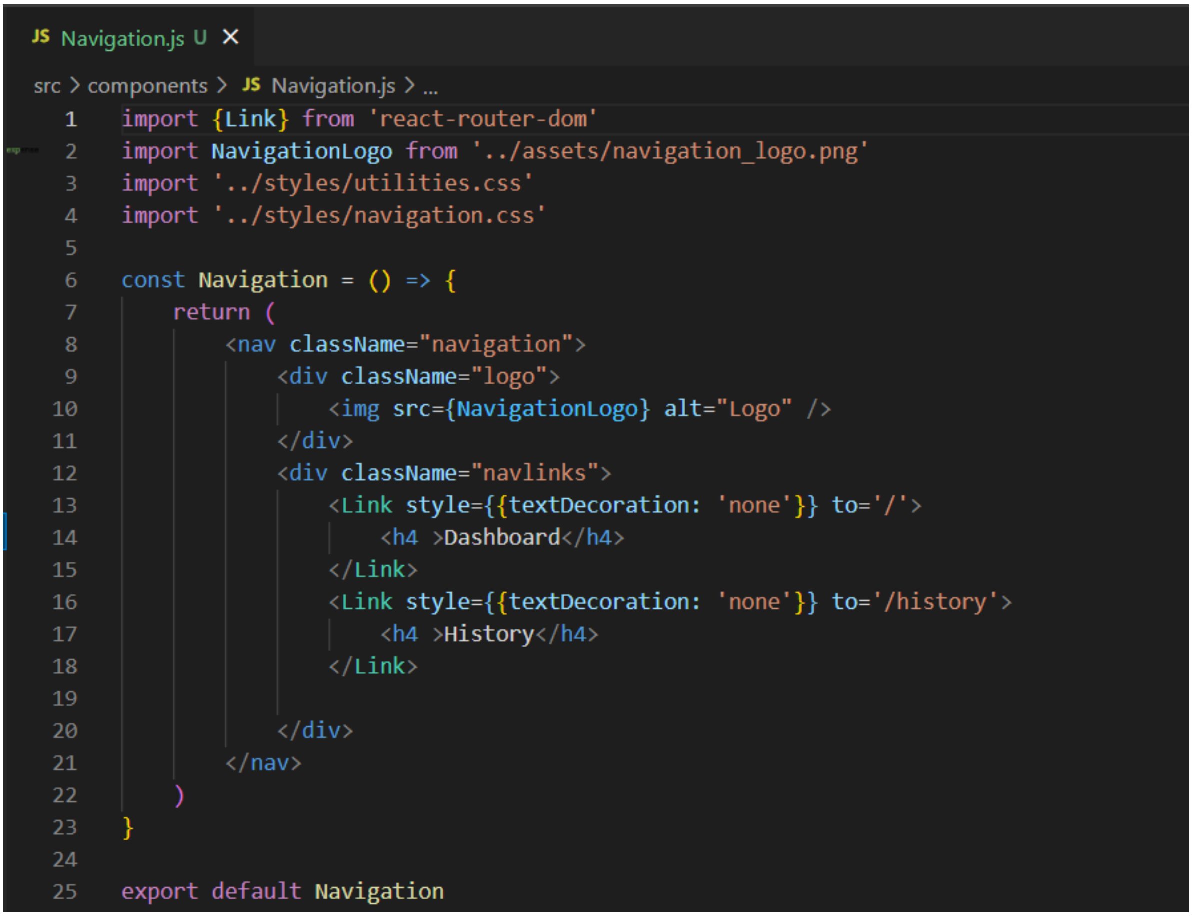
```
import '../styles/table.css'
import HistoryRow from './HistoryRow.js'

const HistoryTable = () => {
    return (
        <div className='table'>
            <div className="row">
                <div className="number">
                    <h4>Sr no</h4>
                </div>
                <div className="name">
                    <h4>Name</h4>
                </div>
                <div className="price">
                    <h4>Price</h4>
                </div>
                <div className="amount">
                    <h4>Quantity</h4>
                </div>
                <div className="total">
                    <h4>Total</h4>
                </div>
                <div className="actions">
                    <h4>Date</h4>
                </div>
            </div>
            <HistoryRow />
        </div>
    )
}

export default HistoryTable
```

E. Home.js –



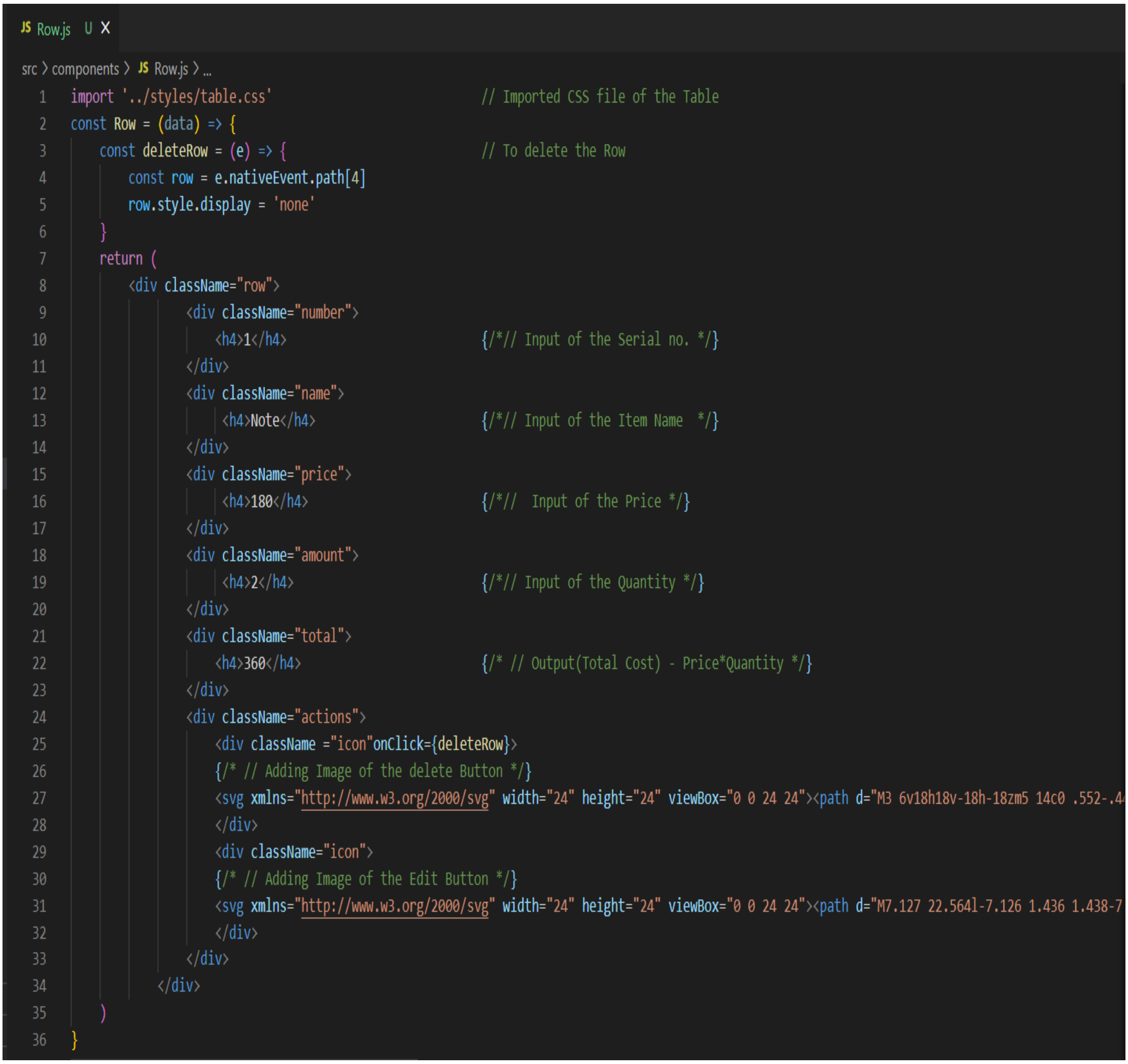
```
import Table from "./Table"
import TopSection from "./TopSection"

const Home = () => {
    return (
        <div>
            <TopSection />
            <Table />
        </div>
    )
}

export default Home
```

F. Navigation.js –

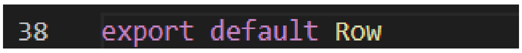


```
import {Link} from 'react-router-dom'
import NavigationLogo from '../assets/navigation_logo.png'
import '../styles/utilities.css'
import '../styles/navigation.css'

const Navigation = () => {
    return (
        <nav className="navigation">
            <div className="logo">
                <img src={NavigationLogo} alt="Logo" />
            </div>
            <div className="navlinks">
                <Link style={{textDecoration: 'none'}} to='/'>
                    <h4 >Dashboard</h4>
                </Link>
                <Link style={{textDecoration: 'none'}} to='/history'>
                    <h4 >History</h4>
                </Link>

            </div>
        </nav>
    )
}

export default Navigation
```

G. Row.js –

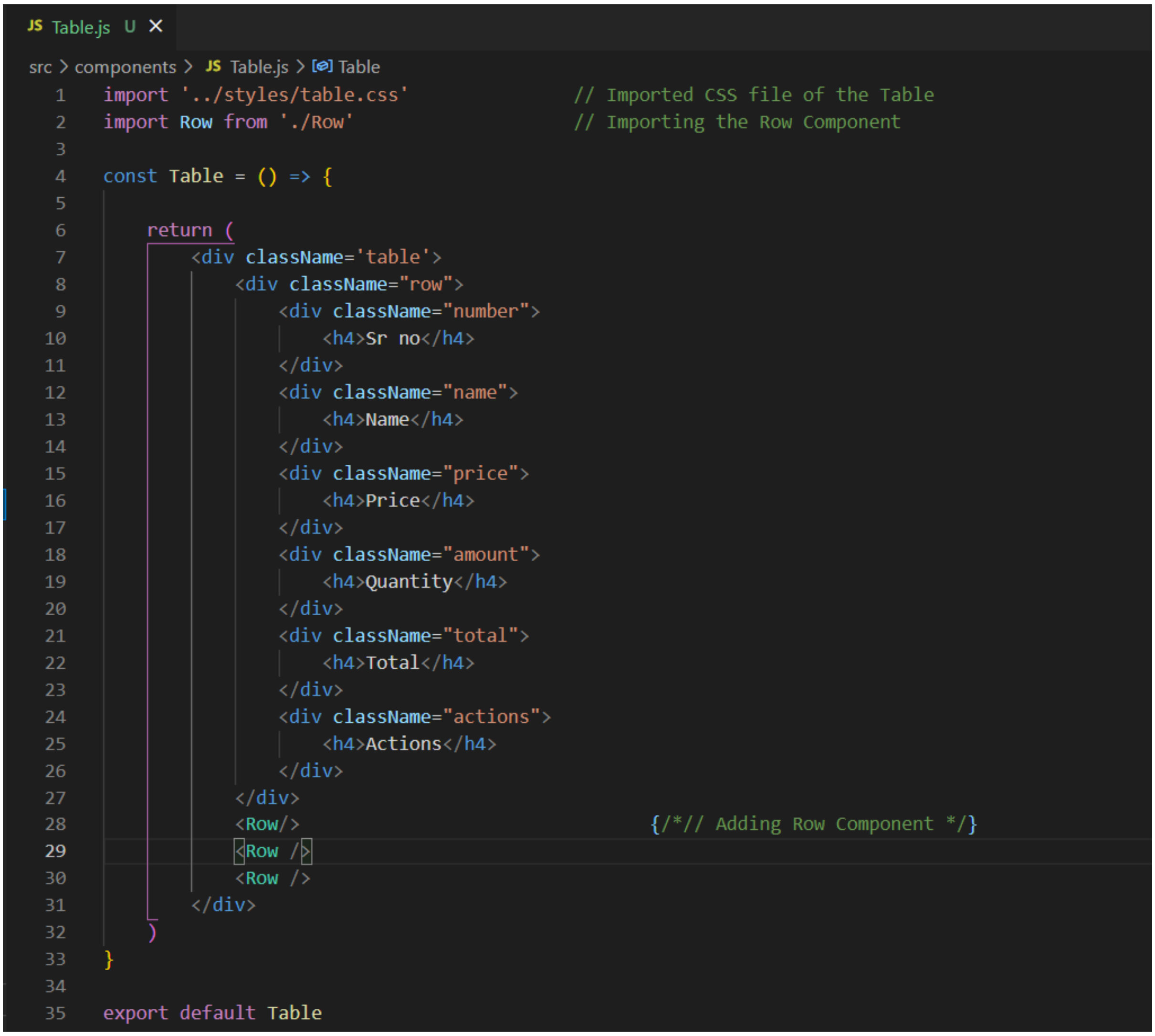


```
import '../styles/table.css'                             // Imported CSS file of the Table
const Row = (data) => {
    const deleteRow = (e) => {                           // To delete the Row
        const row = e.nativeEvent.path[4]
        row.style.display = 'none'
    }
    return (
        <div className="row">
                <div className="number">
                    <h4>1</h4>                           {/*// Input of the Serial no. */}
                </div>
                <div className="name">
                     <h4>Note</h4>                       {/*// Input of the Item Name  */}
                </div>
                <div className="price">
                     <h4>180</h4>                        {/*//  Input of the Price */}
                </div>
                <div className="amount">
                     <h4>2</h4>                          {/*// Input of the Quantity */}
                </div>
                <div className="total">
                    <h4>360</h4>                         {/* // Output(Total Cost) - Price*Quantity */}
                </div>
                <div className="actions">
                    <div className ="icon"onClick={deleteRow}>
                    {/* // Adding Image of the delete Button */}
                    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"><path d="M3 6v18h18v-18h-18zm5 14c0 .552-.4
                    </div>
                    <div className="icon">
                    {/* // Adding Image of the Edit Button */}
                    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"><path d="M7.127 22.564l-7.126 1.436 1.438-7
                    </div>
                </div>
            </div>
    )
}
```



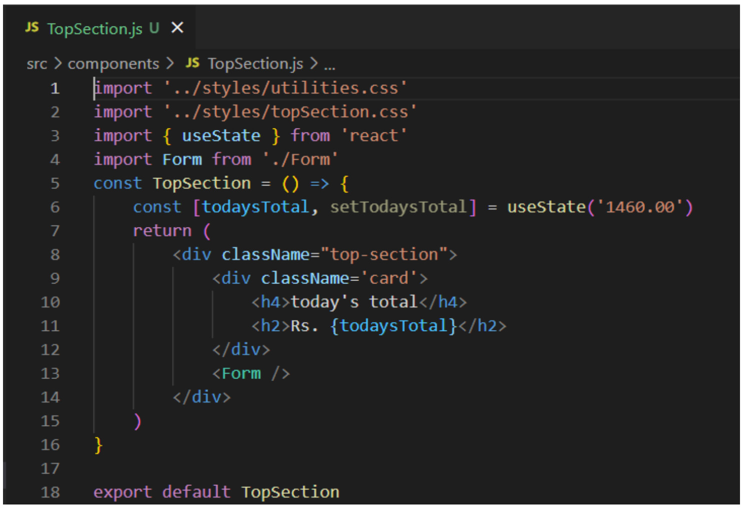
```
export default Row
```

H. Table.js –



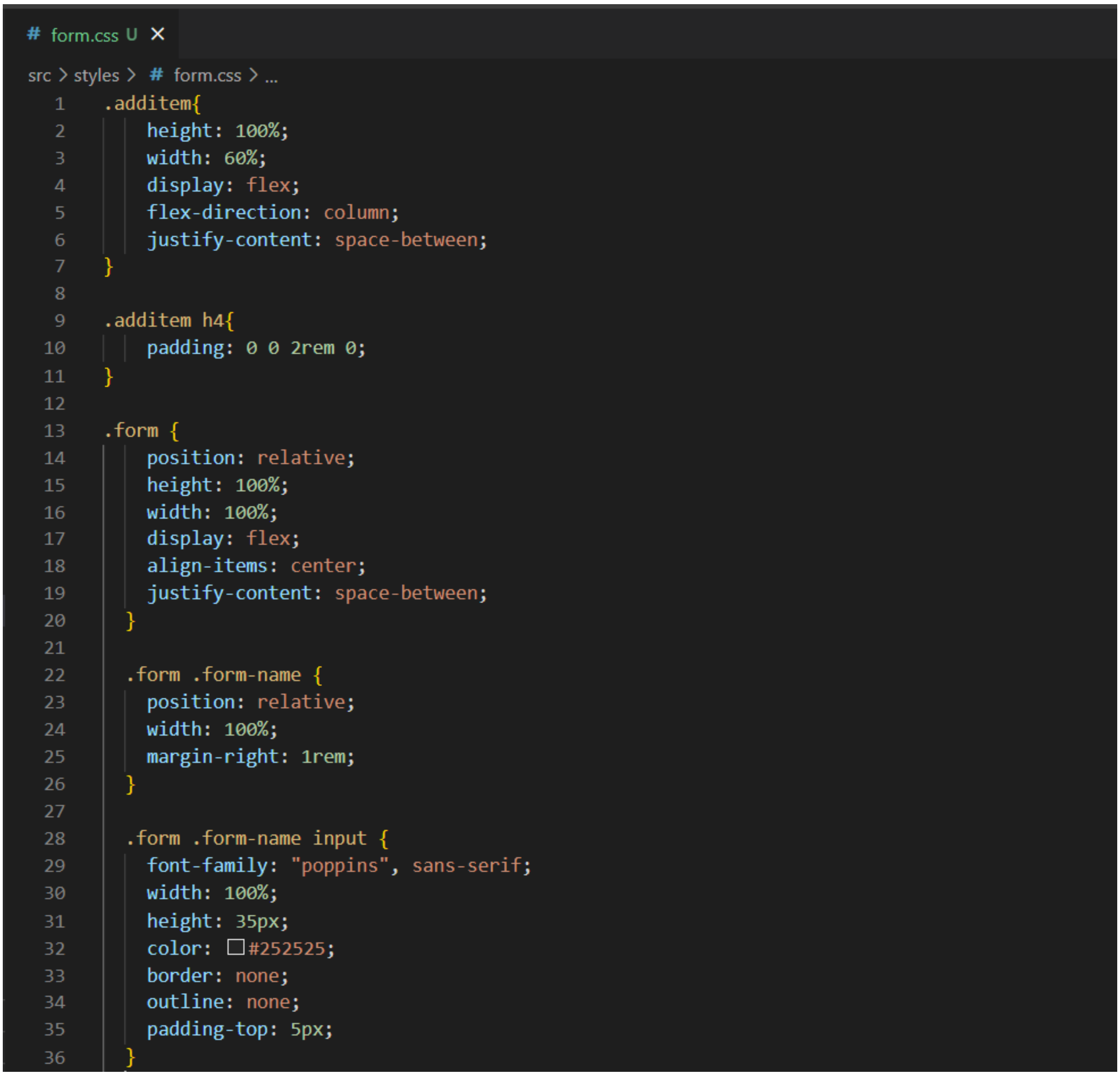
```
import '../styles/table.css'                 // Imported CSS file of the Table
import Row from './Row'                      // Importing the Row Component

const Table = () => {

    return (
        <div className='table'>
            <div className="row">
                <div className="number">
                    <h4>Sr no</h4>
                </div>
                <div className="name">
                    <h4>Name</h4>
                </div>
                <div className="price">
                    <h4>Price</h4>
                </div>
                <div className="amount">
                    <h4>Quantity</h4>
                </div>
                <div className="total">
                    <h4>Total</h4>
                </div>
                <div className="actions">
                    <h4>Actions</h4>
                </div>
            </div>
            <Row/>                                   {/*// Adding Row Component */}
            <Row />
            <Row />
        </div>
    )
}

export default Table
```

I. TopSection.js –

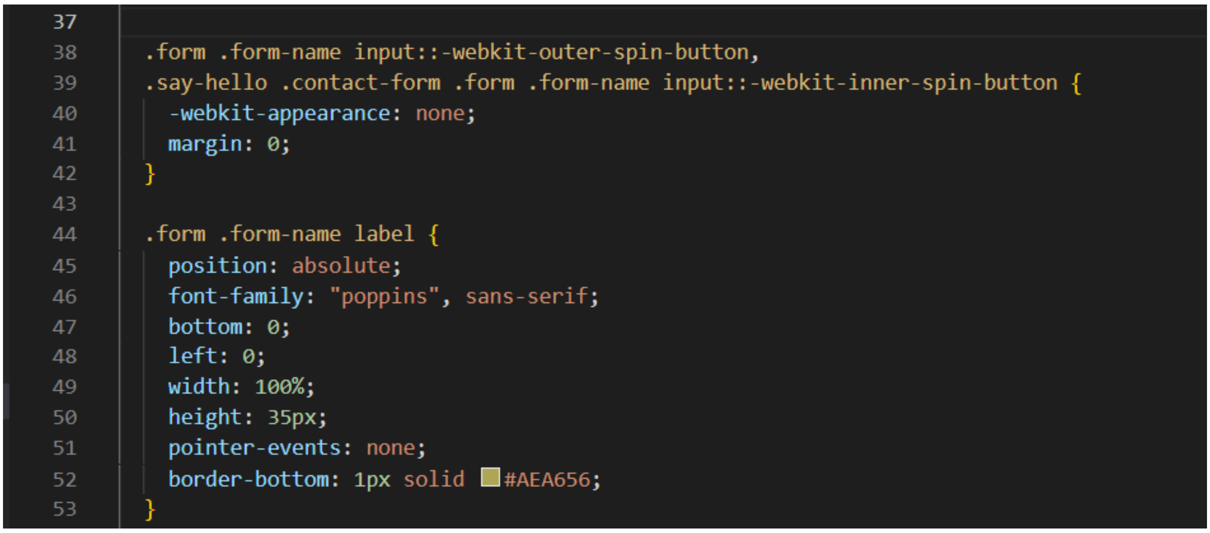


```
import '../styles/utilities.css'
import '../styles/topSection.css'
import { useState } from 'react'
import Form from './Form'
const TopSection = () => {
    const [todaysTotal, setTodaysTotal] = useState('1460.00')
    return (
        <div className="top-section">
            <div className='card'>
                <h4>today's total</h4>
                <h2>Rs. {todaysTotal}</h2>
            </div>
            <Form />
        </div>
    )
}

export default TopSection
```

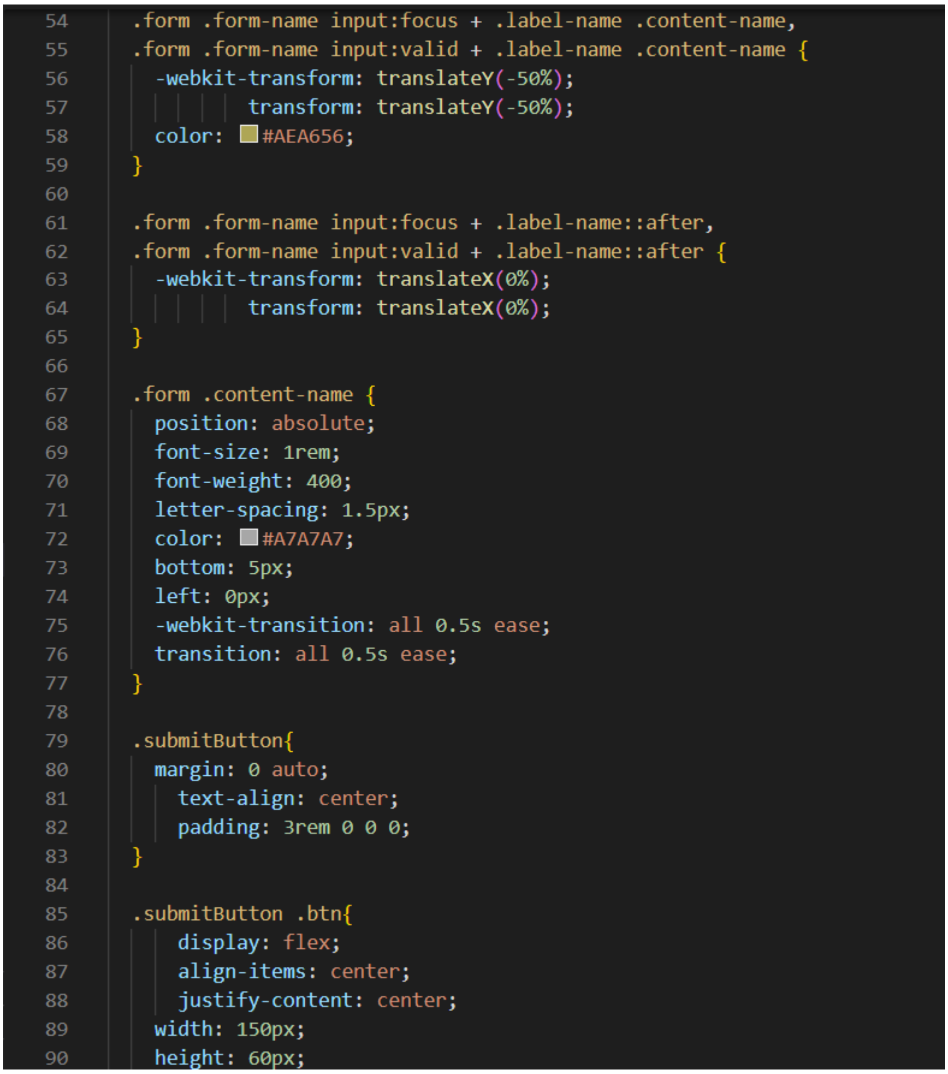
J. Form.css –



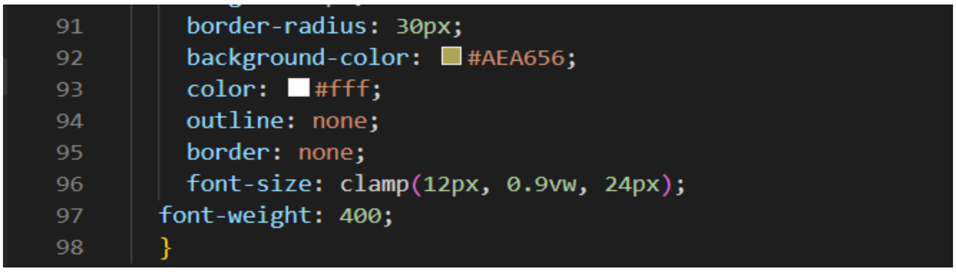
```css
.additem{
    height: 100%;
    width: 60%;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
}

.additem h4{
    padding: 0 0 2rem 0;
}

.form {
    position: relative;
    height: 100%;
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: space-between;
  }

  .form .form-name {
    position: relative;
    width: 100%;
    margin-right: 1rem;
  }

  .form .form-name input {
    font-family: "poppins", sans-serif;
    width: 100%;
    height: 35px;
    color: #252525;
    border: none;
    outline: none;
    padding-top: 5px;
  }
```



```css

  .form .form-name input::-webkit-outer-spin-button,
  .say-hello .contact-form .form .form-name input::-webkit-inner-spin-button {
    -webkit-appearance: none;
    margin: 0;
  }

  .form .form-name label {
    position: absolute;
    font-family: "poppins", sans-serif;
    bottom: 0;
    left: 0;
    width: 100%;
    height: 35px;
    pointer-events: none;
    border-bottom: 1px solid #AEA656;
  }
```

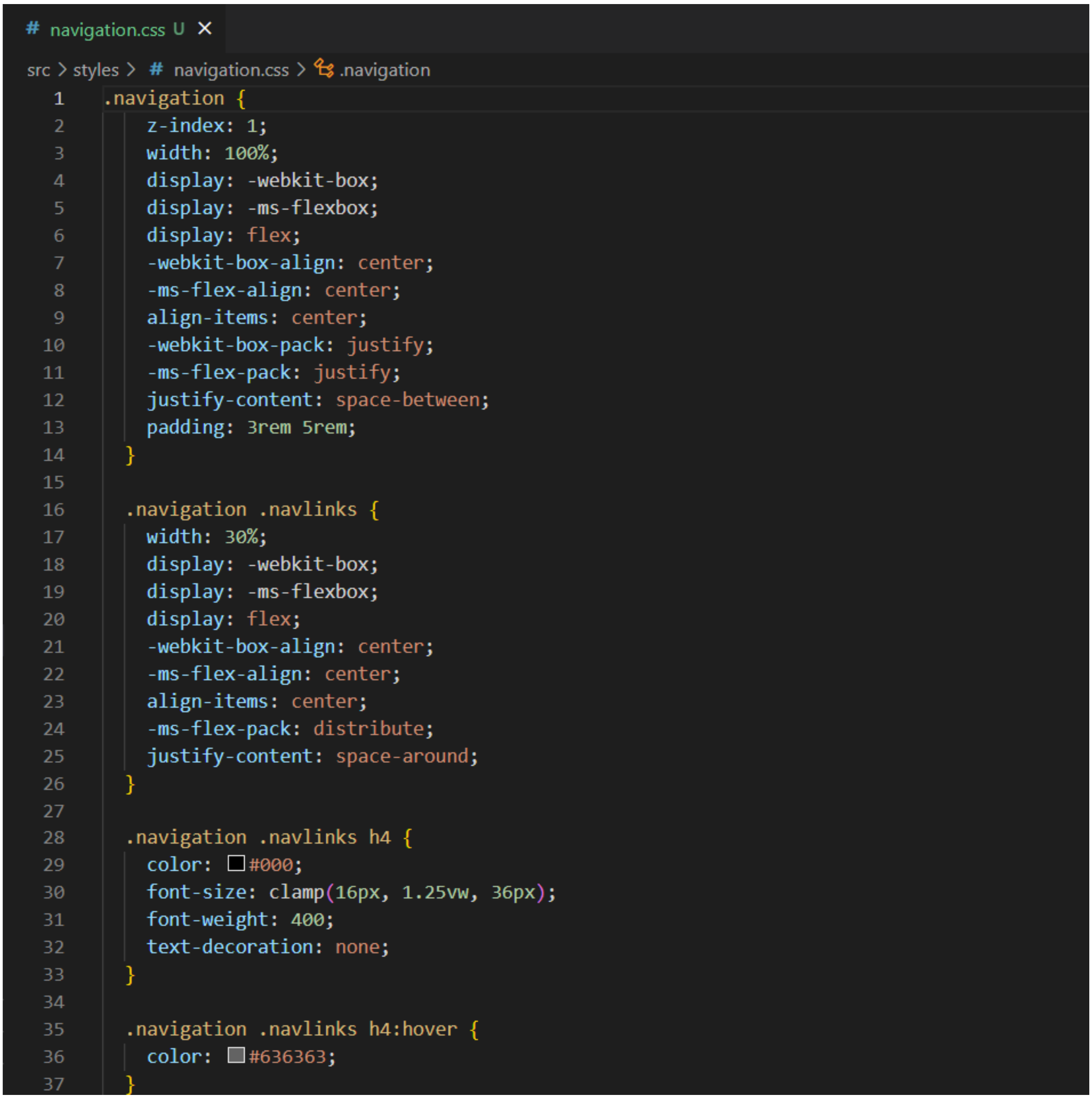


```css
.form .form-name input:focus + .label-name .content-name,
.form .form-name input:valid + .label-name .content-name {
  -webkit-transform: translateY(-50%);
          transform: translateY(-50%);
  color: #AEA656;
}

.form .form-name input:focus + .label-name::after,
.form .form-name input:valid + .label-name::after {
  -webkit-transform: translateX(0%);
          transform: translateX(0%);
}

.form .content-name {
  position: absolute;
  font-size: 1rem;
  font-weight: 400;
  letter-spacing: 1.5px;
  color: #A7A7A7;
  bottom: 5px;
  left: 0px;
  -webkit-transition: all 0.5s ease;
  transition: all 0.5s ease;
}

.submitButton{
  margin: 0 auto;
    text-align: center;
    padding: 3rem 0 0 0;
}

.submitButton .btn{
    display: flex;
    align-items: center;
    justify-content: center;
  width: 150px;
  height: 60px;
```

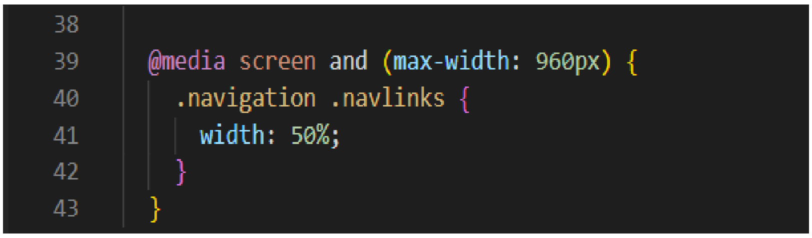


```css
    border-radius: 30px;
    background-color: #AEA656;
    color: #fff;
    outline: none;
    border: none;
    font-size: clamp(12px, 0.9vw, 24px);
  font-weight: 400;
  }
```

K. navigation.css –

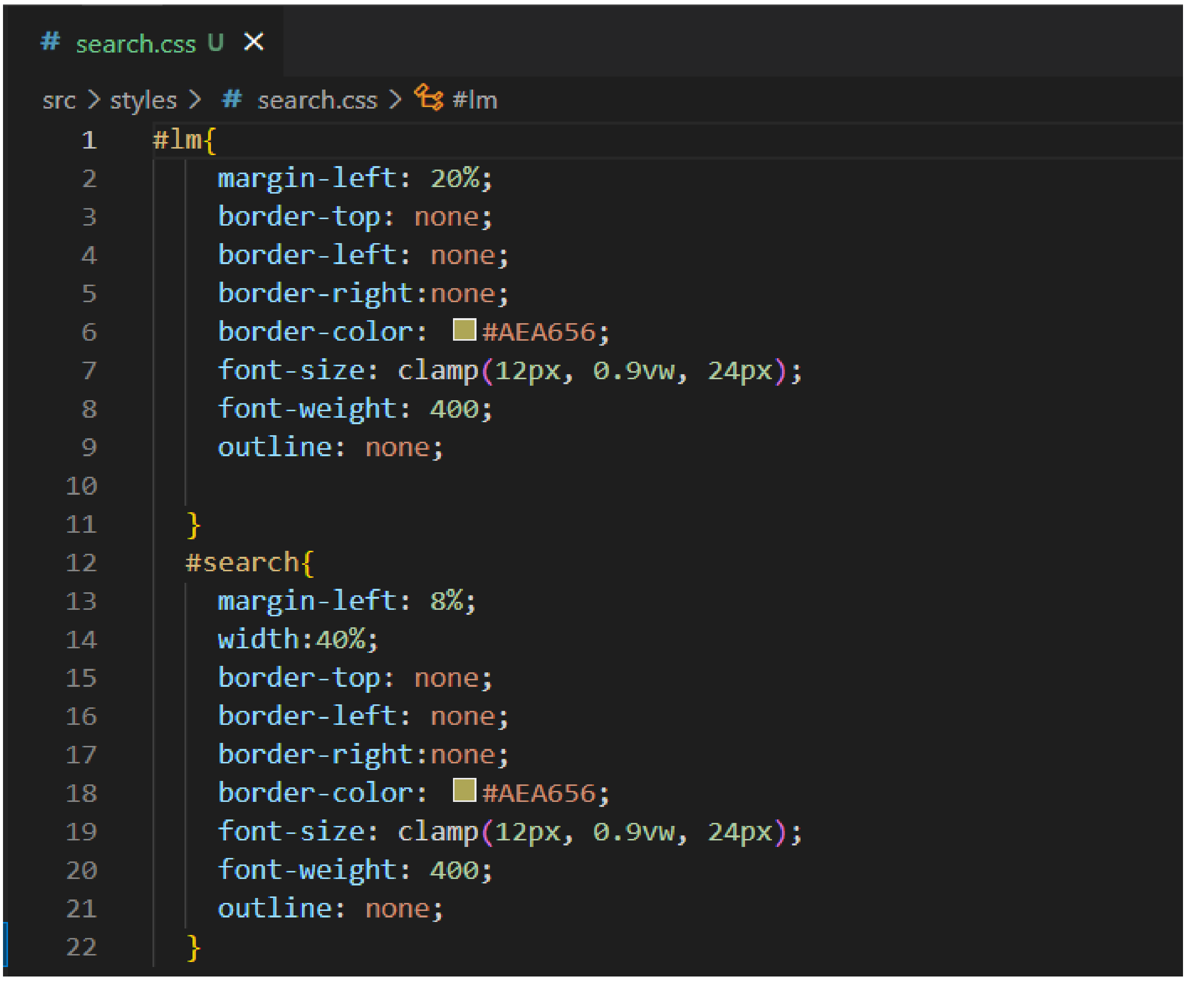


```css
.navigation {
    z-index: 1;
    width: 100%;
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-box-align: center;
    -ms-flex-align: center;
    align-items: center;
    -webkit-box-pack: justify;
    -ms-flex-pack: justify;
    justify-content: space-between;
    padding: 3rem 5rem;
  }

  .navigation .navlinks {
    width: 30%;
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-box-align: center;
    -ms-flex-align: center;
    align-items: center;
    -ms-flex-pack: distribute;
    justify-content: space-around;
  }

  .navigation .navlinks h4 {
    color: #000;
    font-size: clamp(16px, 1.25vw, 36px);
    font-weight: 400;
    text-decoration: none;
  }

  .navigation .navlinks h4:hover {
    color: #636363;
  }
```



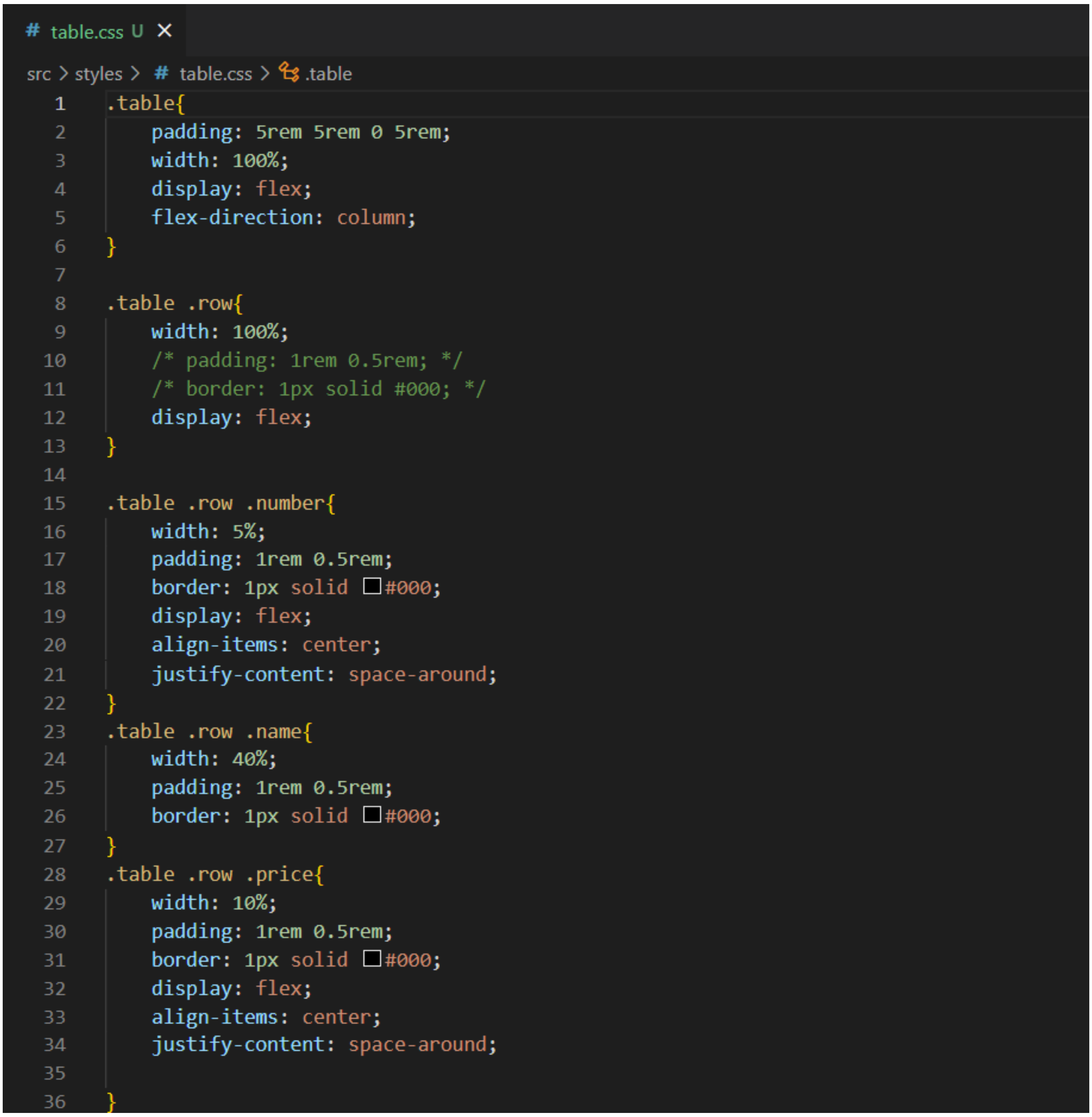
```css

  @media screen and (max-width: 960px) {
    .navigation .navlinks {
      width: 50%;
    }
  }
```

L. search.css –

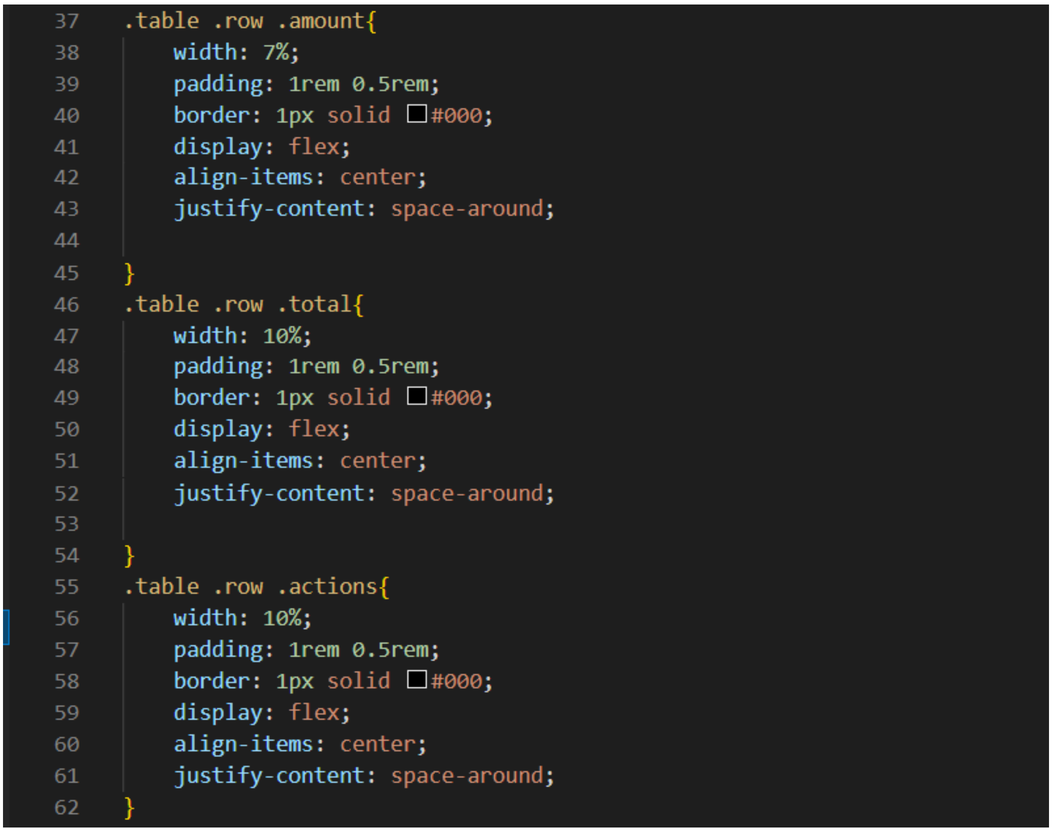


```css
#lm{
    margin-left: 20%;
    border-top: none;
    border-left: none;
    border-right:none;
    border-color: #AEA656;
    font-size: clamp(12px, 0.9vw, 24px);
    font-weight: 400;
    outline: none;

  }
  #search{
    margin-left: 8%;
    width:40%;
    border-top: none;
    border-left: none;
    border-right:none;
    border-color: #AEA656;
    font-size: clamp(12px, 0.9vw, 24px);
    font-weight: 400;
    outline: none;
  }
```

M. table.css –

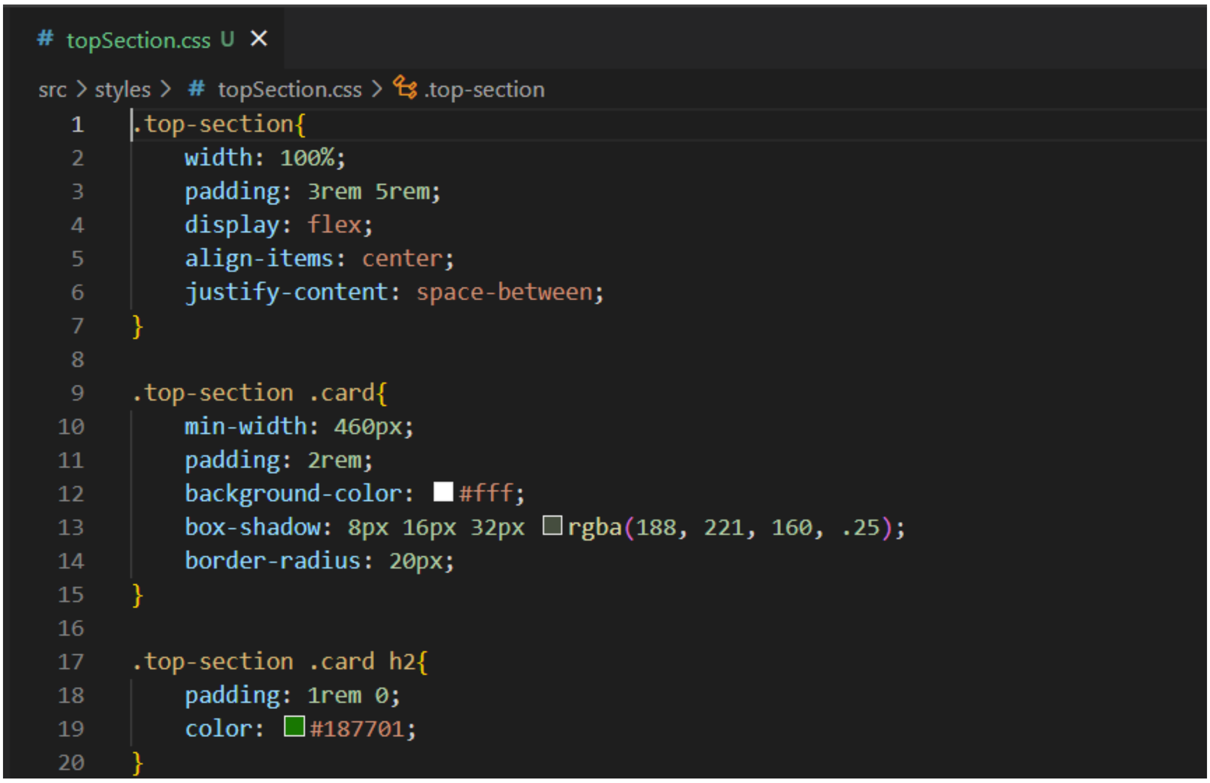


```css
.table{
    padding: 5rem 5rem 0 5rem;
    width: 100%;
    display: flex;
    flex-direction: column;
}

.table .row{
    width: 100%;
    /* padding: 1rem 0.5rem; */
    /* border: 1px solid #000; */
    display: flex;
}

.table .row .number{
    width: 5%;
    padding: 1rem 0.5rem;
    border: 1px solid #000;
    display: flex;
    align-items: center;
    justify-content: space-around;
}
.table .row .name{
    width: 40%;
    padding: 1rem 0.5rem;
    border: 1px solid #000;
}
.table .row .price{
    width: 10%;
    padding: 1rem 0.5rem;
    border: 1px solid #000;
    display: flex;
    align-items: center;
    justify-content: space-around;

}
```



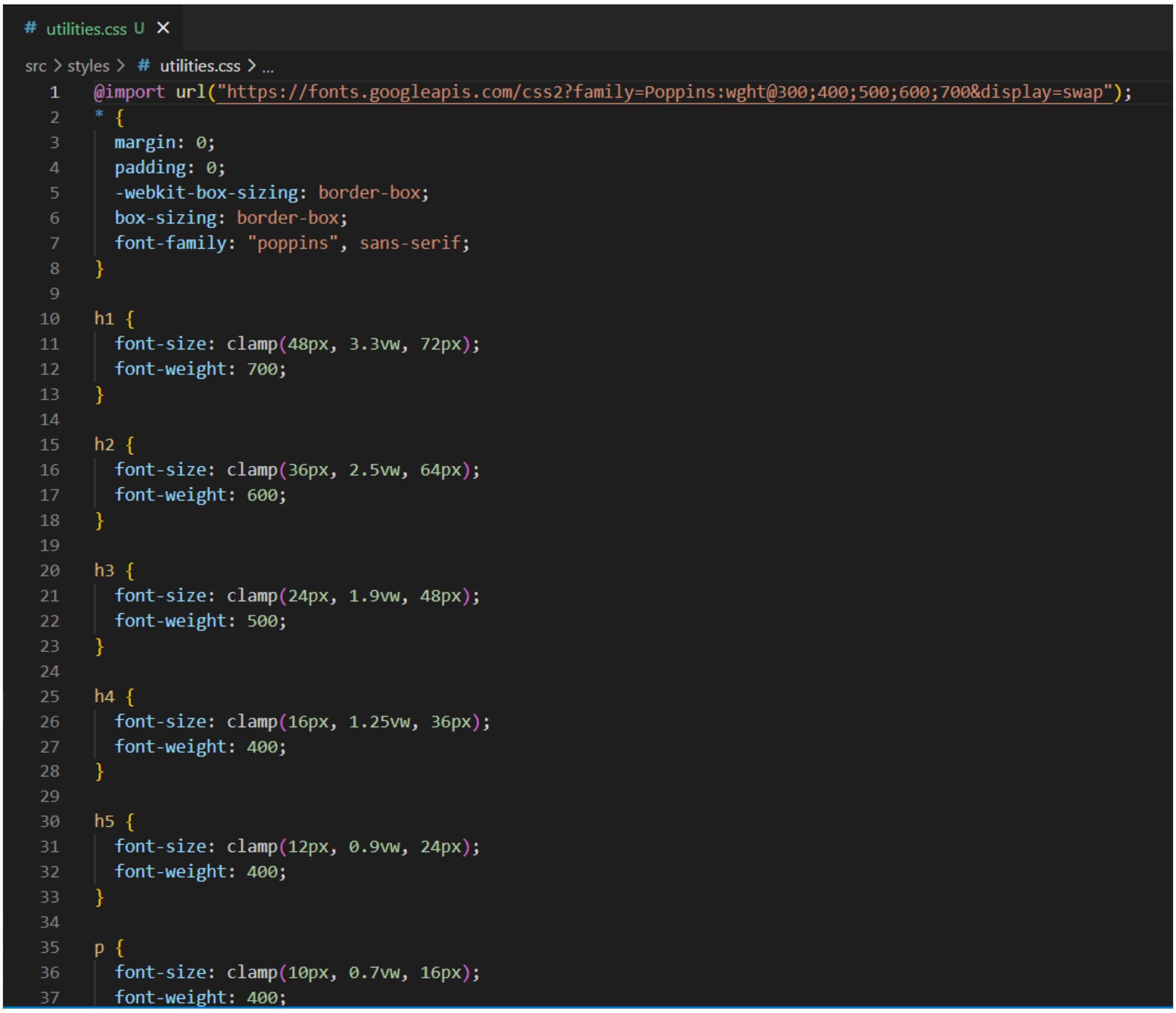
```css
.table .row .amount{
    width: 7%;
    padding: 1rem 0.5rem;
    border: 1px solid #000;
    display: flex;
    align-items: center;
    justify-content: space-around;

}
.table .row .total{
    width: 10%;
    padding: 1rem 0.5rem;
    border: 1px solid #000;
    display: flex;
    align-items: center;
    justify-content: space-around;

}
.table .row .actions{
    width: 10%;
    padding: 1rem 0.5rem;
    border: 1px solid #000;
    display: flex;
    align-items: center;
    justify-content: space-around;
}
```

N. topSection.css –



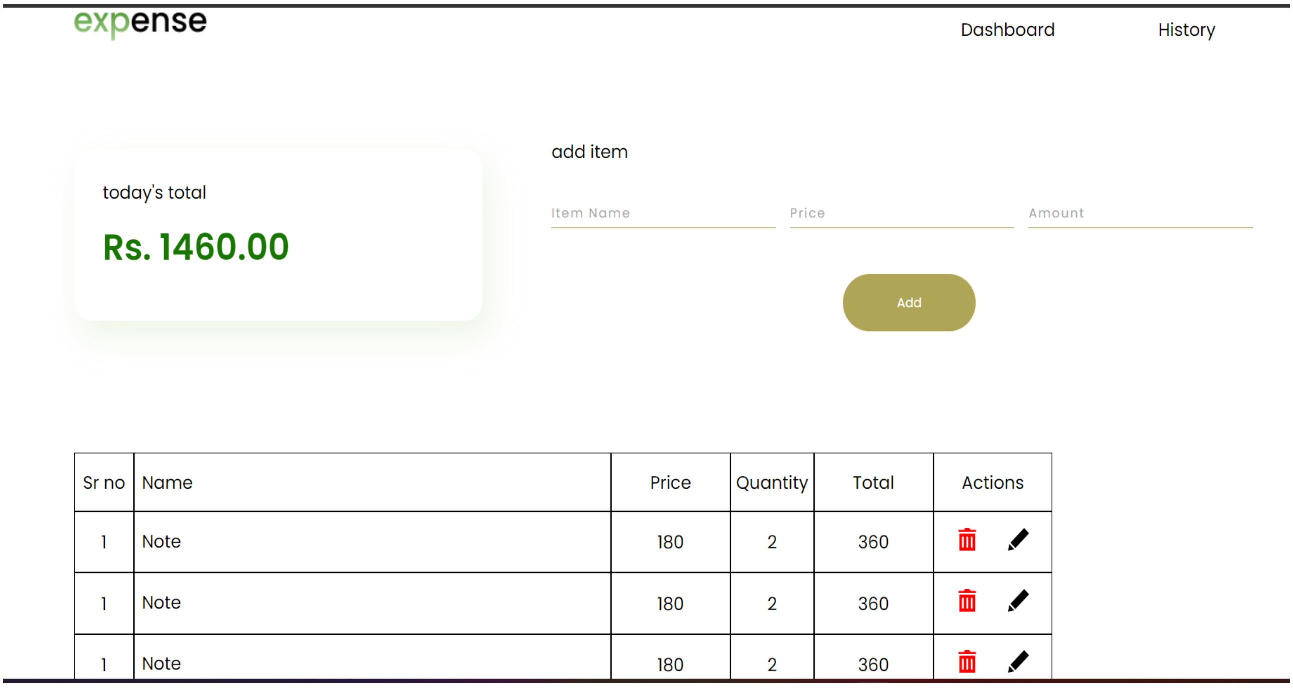
```css
.top-section{
    width: 100%;
    padding: 3rem 5rem;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.top-section .card{
    min-width: 460px;
    padding: 2rem;
    background-color: #fff;
    box-shadow: 8px 16px 32px rgba(188, 221, 160, .25);
    border-radius: 20px;
}

.top-section .card h2{
    padding: 1rem 0;
    color: #187701;
}
```

O. Utilities.css –



```css
@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap");
* {
  margin: 0;
  padding: 0;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
  font-family: "poppins", sans-serif;
}

h1 {
  font-size: clamp(48px, 3.3vw, 72px);
  font-weight: 700;
}

h2 {
  font-size: clamp(36px, 2.5vw, 64px);
  font-weight: 600;
}

h3 {
  font-size: clamp(24px, 1.9vw, 48px);
  font-weight: 500;
}

h4 {
  font-size: clamp(16px, 1.25vw, 36px);
  font-weight: 400;
}

h5 {
  font-size: clamp(12px, 0.9vw, 24px);
  font-weight: 400;
}

p {
  font-size: clamp(10px, 0.7vw, 16px);
  font-weight: 400;
```

8. **Output ss** (Frontend)–

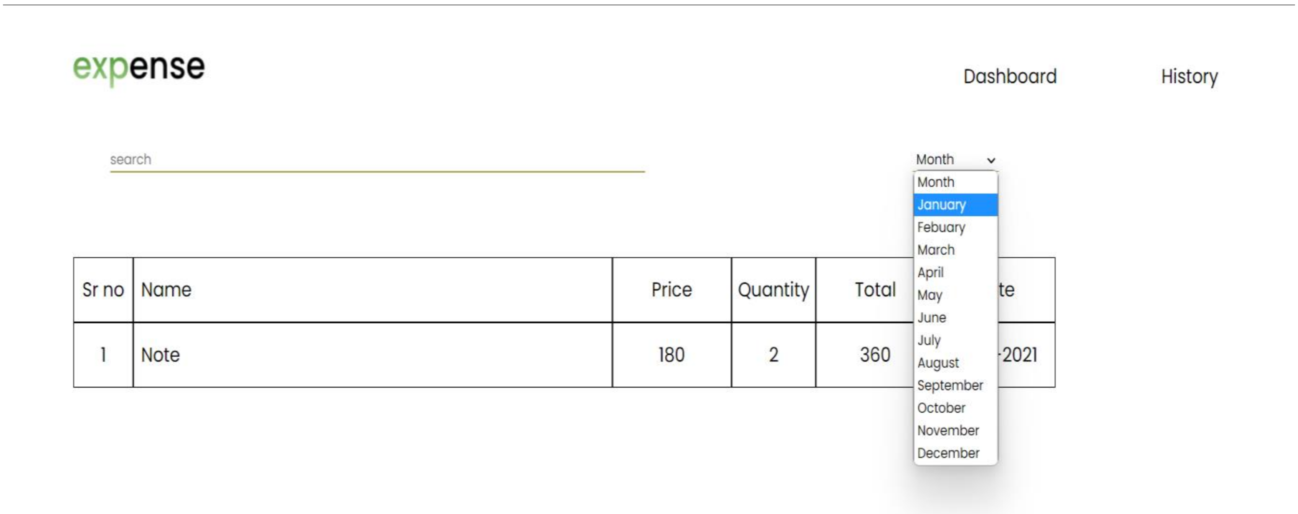


today's total

Rs. 1460.00

add item

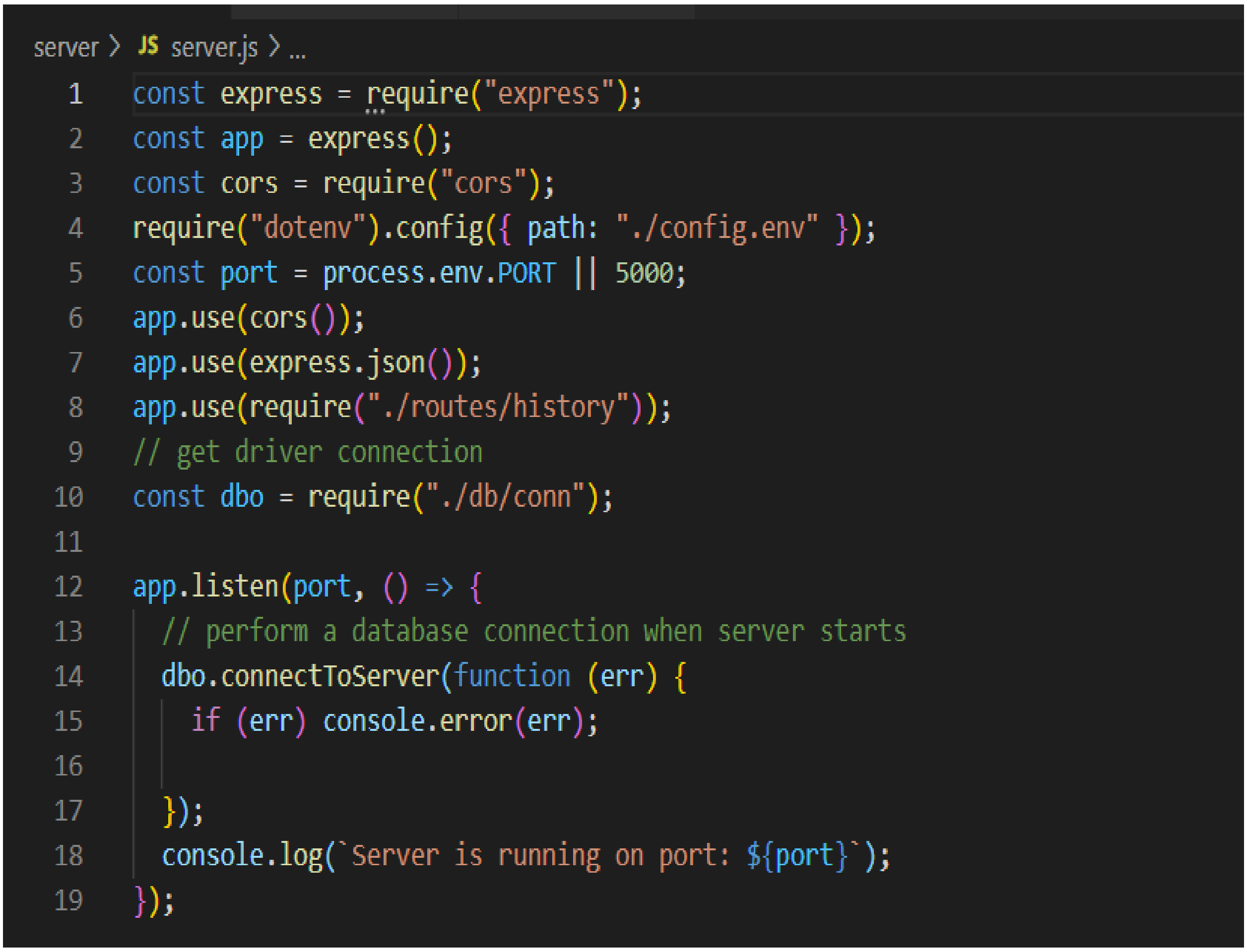
Sr no	Name	Price	Quantity	Total	Actions
1	Note	180	2	360	🗑 ✎
1	Note	180	2	360	🗑 ✎
1	Note	180	2	360	🗑 ✎



Sr no	Name	Price	Quantity	Total	te
1	Note	180	2	360	-2021

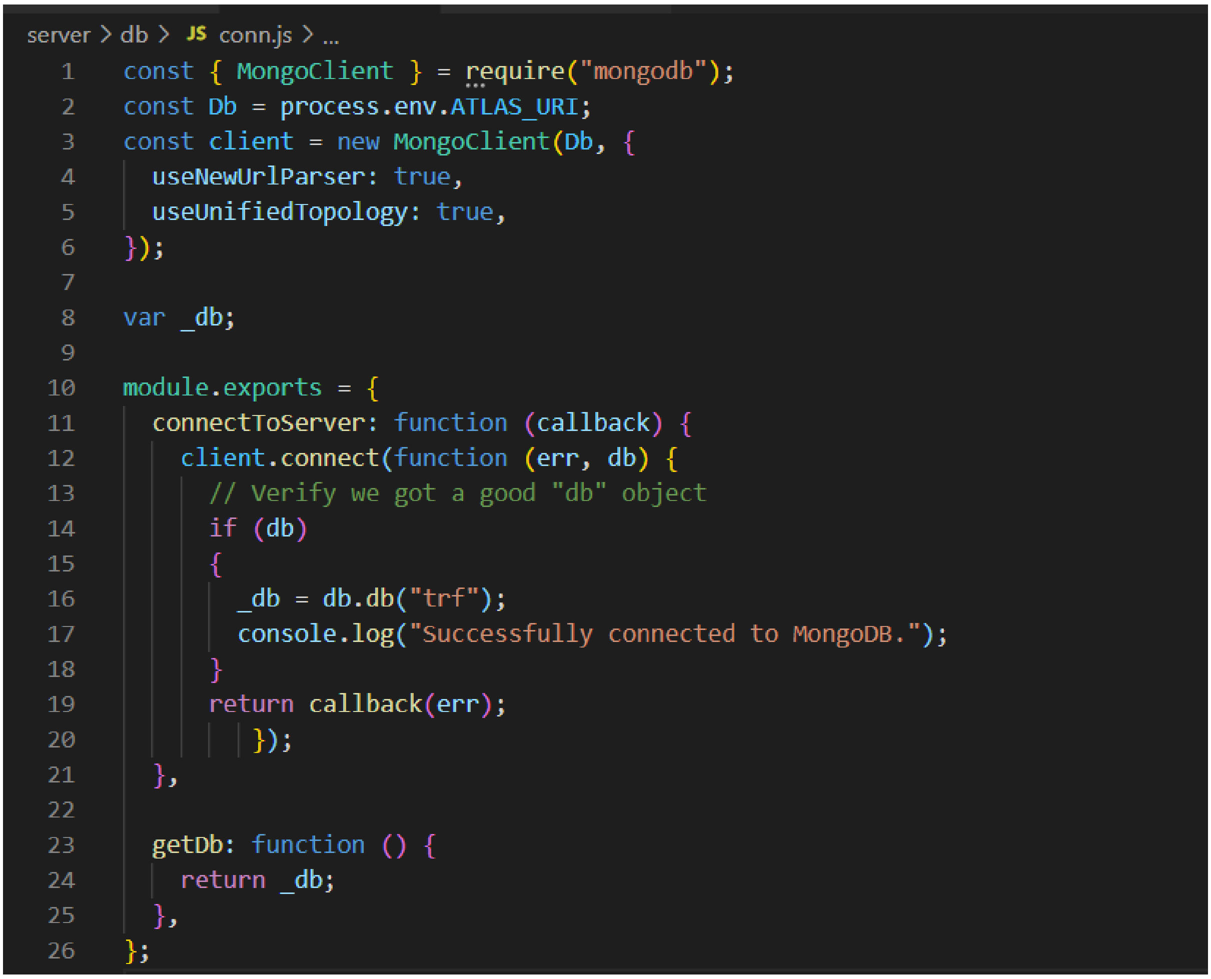
Backend –

A. Server.js –



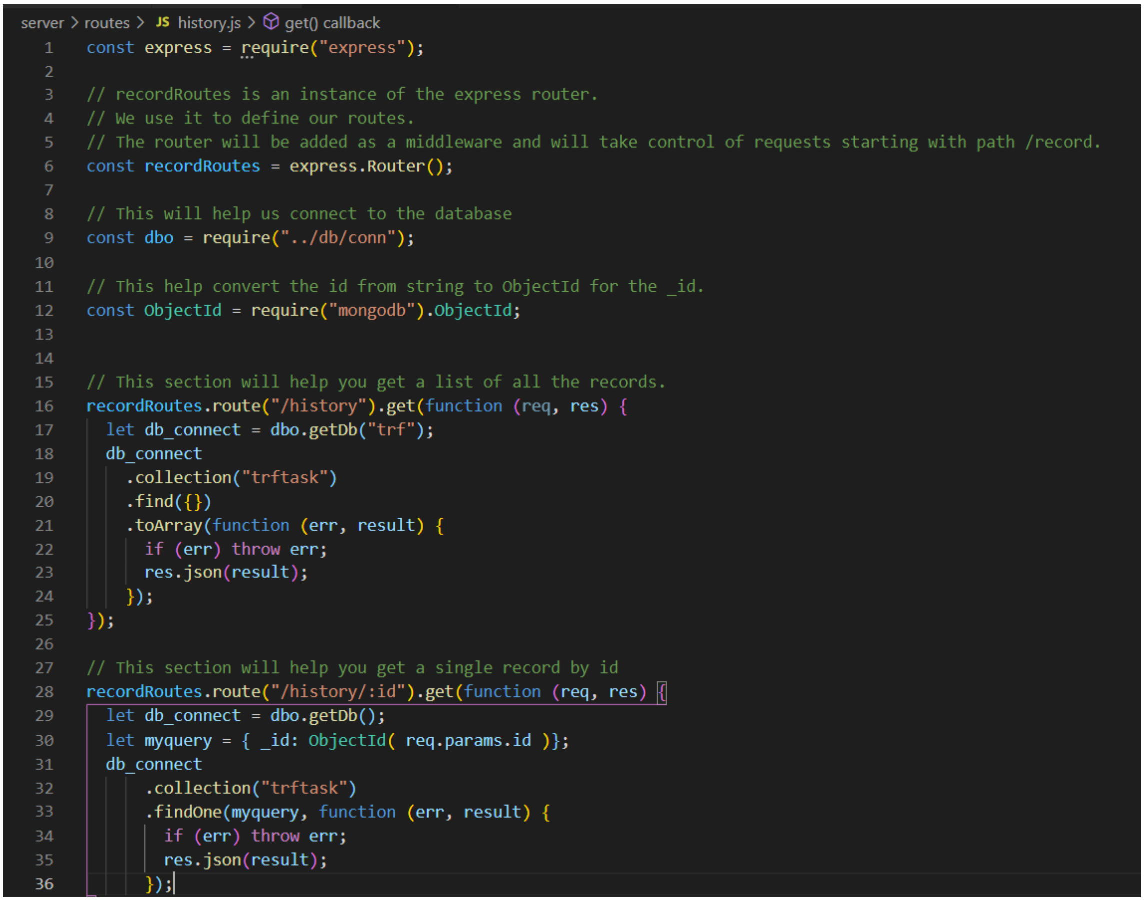
```javascript
const express = require("express");
const app = express();
const cors = require("cors");
require("dotenv").config({ path: "./config.env" });
const port = process.env.PORT || 5000;
app.use(cors());
app.use(express.json());
app.use(require("./routes/history"));
// get driver connection
const dbo = require("./db/conn");

app.listen(port, () => {
  // perform a database connection when server starts
  dbo.connectToServer(function (err) {
    if (err) console.error(err);

  });
  console.log(`Server is running on port: ${port}`);
});
```

B. Conn.js –

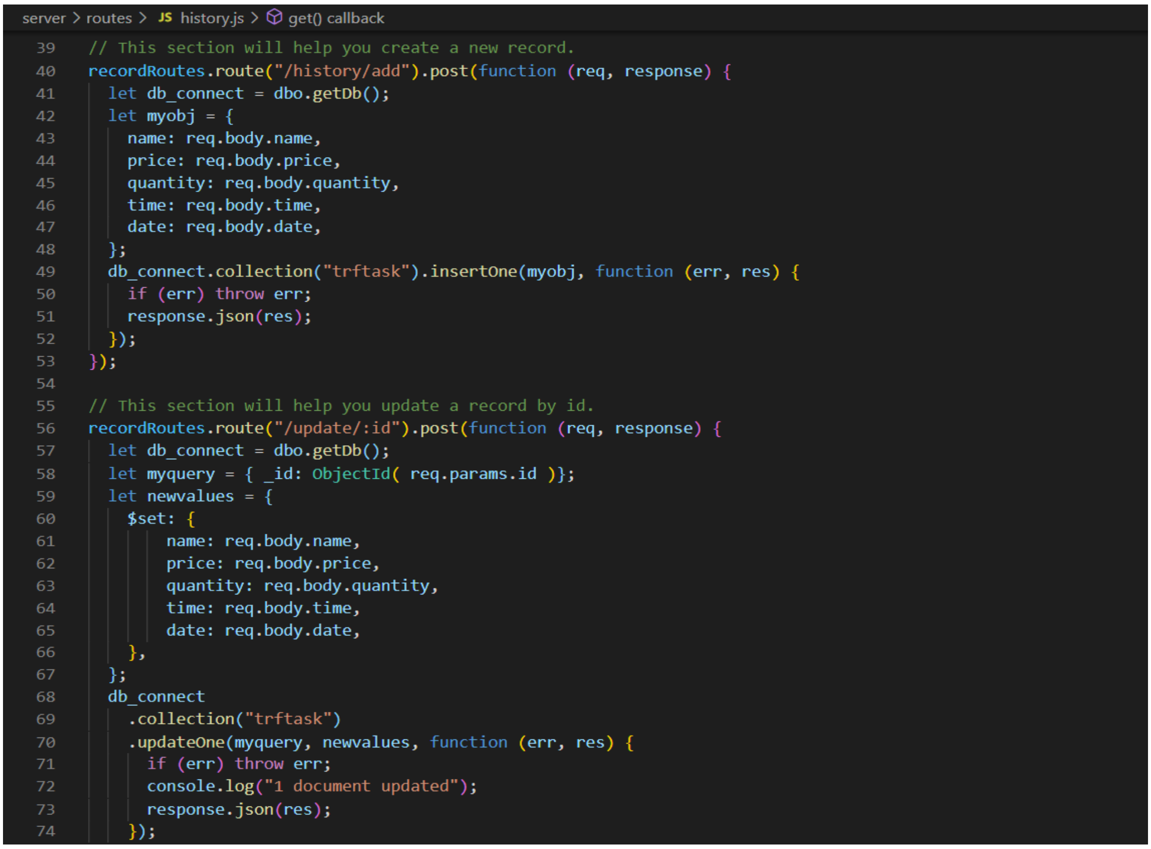


```
const { MongoClient } = require("mongodb");
const Db = process.env.ATLAS_URI;
const client = new MongoClient(Db, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

var _db;

module.exports = {
  connectToServer: function (callback) {
    client.connect(function (err, db) {
      // Verify we got a good "db" object
      if (db)
      {
        _db = db.db("trf");
        console.log("Successfully connected to MongoDB.");
      }
      return callback(err);
         });
  },

  getDb: function () {
    return _db;
  },
};
```

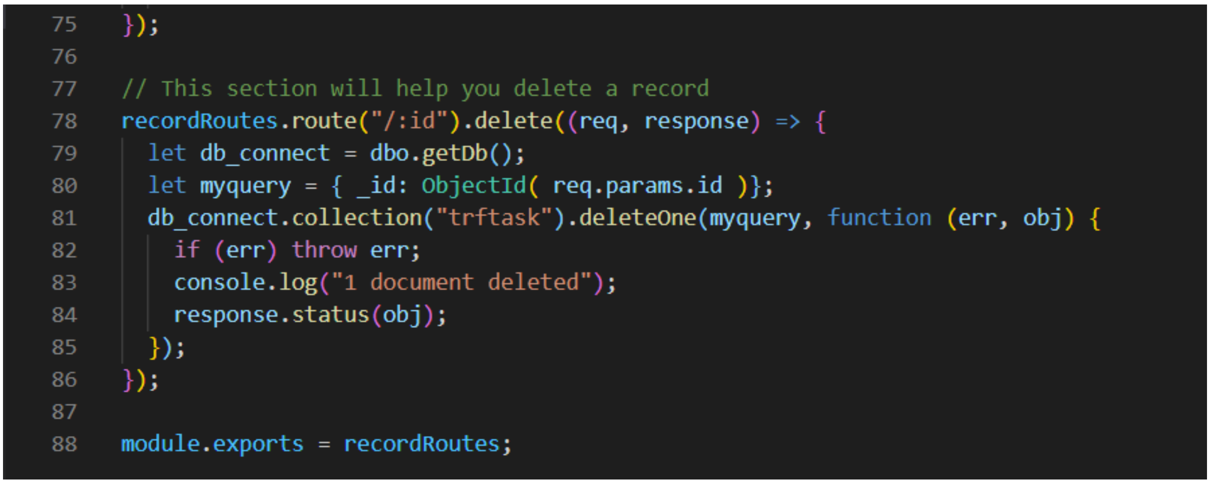
C. History.js –



```javascript
const express = require("express");

// recordRoutes is an instance of the express router.
// We use it to define our routes.
// The router will be added as a middleware and will take control of requests starting with path /record.
const recordRoutes = express.Router();

// This will help us connect to the database
const dbo = require("../db/conn");

// This help convert the id from string to ObjectId for the _id.
const ObjectId = require("mongodb").ObjectId;


// This section will help you get a list of all the records.
recordRoutes.route("/history").get(function (req, res) {
  let db_connect = dbo.getDb("trf");
  db_connect
    .collection("trftask")
    .find({})
    .toArray(function (err, result) {
      if (err) throw err;
      res.json(result);
    });
});

// This section will help you get a single record by id
recordRoutes.route("/history/:id").get(function (req, res) {
  let db_connect = dbo.getDb();
  let myquery = { _id: ObjectId( req.params.id )};
  db_connect
      .collection("trftask")
      .findOne(myquery, function (err, result) {
        if (err) throw err;
        res.json(result);
      });
```



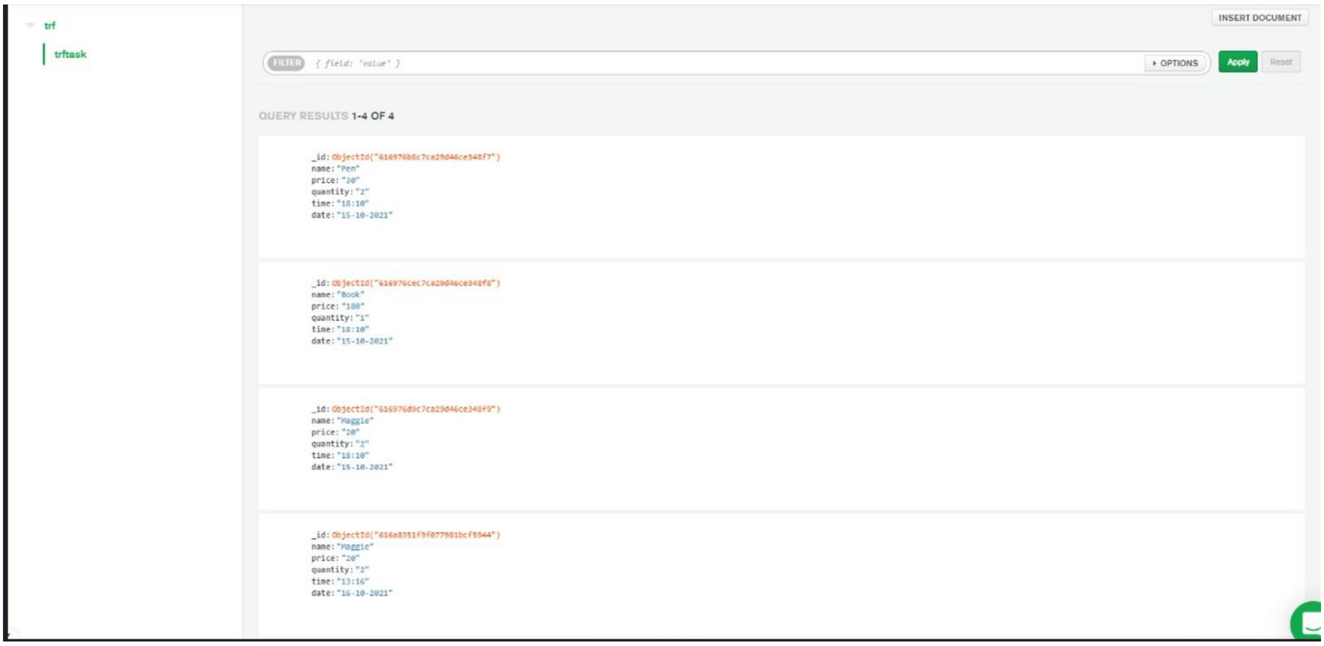
```javascript
// This section will help you create a new record.
recordRoutes.route("/history/add").post(function (req, response) {
  let db_connect = dbo.getDb();
  let myobj = {
    name: req.body.name,
    price: req.body.price,
    quantity: req.body.quantity,
    time: req.body.time,
    date: req.body.date,
  };
  db_connect.collection("trftask").insertOne(myobj, function (err, res) {
    if (err) throw err;
    response.json(res);
  });
});

// This section will help you update a record by id.
recordRoutes.route("/update/:id").post(function (req, response) {
  let db_connect = dbo.getDb();
  let myquery = { _id: ObjectId( req.params.id )};
  let newvalues = {
    $set: {
        name: req.body.name,
        price: req.body.price,
        quantity: req.body.quantity,
        time: req.body.time,
        date: req.body.date,
    },
  };
  db_connect
    .collection("trftask")
    .updateOne(myquery, newvalues, function (err, res) {
      if (err) throw err;
      console.log("1 document updated");
      response.json(res);
    });
```



```
});

// This section will help you delete a record
recordRoutes.route("/:id").delete((req, response) => {
  let db_connect = dbo.getDb();
  let myquery = { _id: ObjectId( req.params.id )};
  db_connect.collection("trftask").deleteOne(myquery, function (err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    response.status(obj);
  });
});

module.exports = recordRoutes;
```

Output ss – (Backend) –

A. MongoDb – Data Collection -



QUERY RESULTS 1-4 OF 4

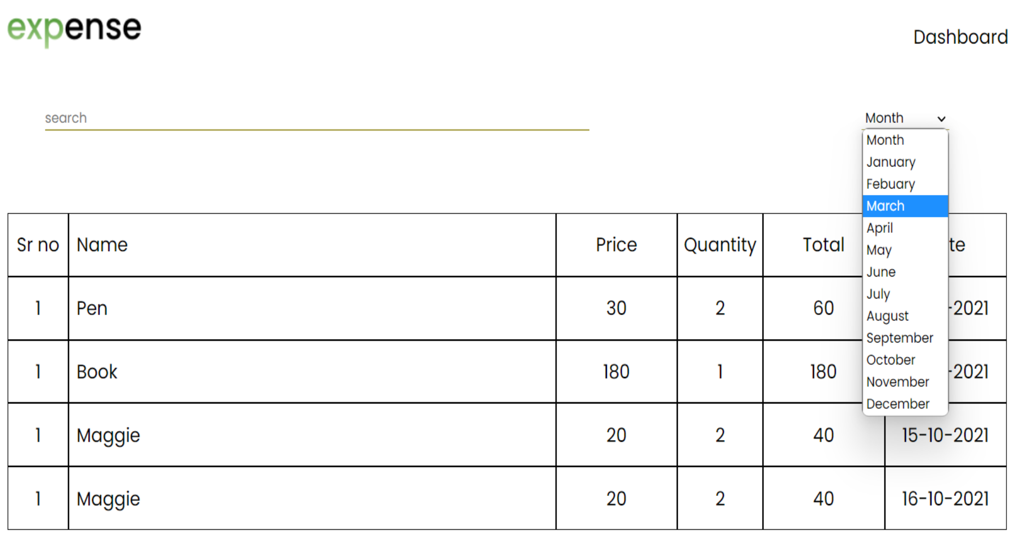
```
_id: ObjectId("616976b8c7ca29d46ce348f7")
name: "Pen"
price: "30"
quantity: "2"
time: "18:10"
date: "15-10-2021"
```

```
_id: ObjectId("616976cec7ca29d46ce348f8")
name: "Book"
price: "100"
quantity: "1"
time: "18:10"
date: "15-10-2021"
```

```
_id: ObjectId("616976d9c7ca29d46ce348f9")
name: "Maggie"
price: "20"
quantity: "2"
time: "18:10"
date: "15-10-2021"
```

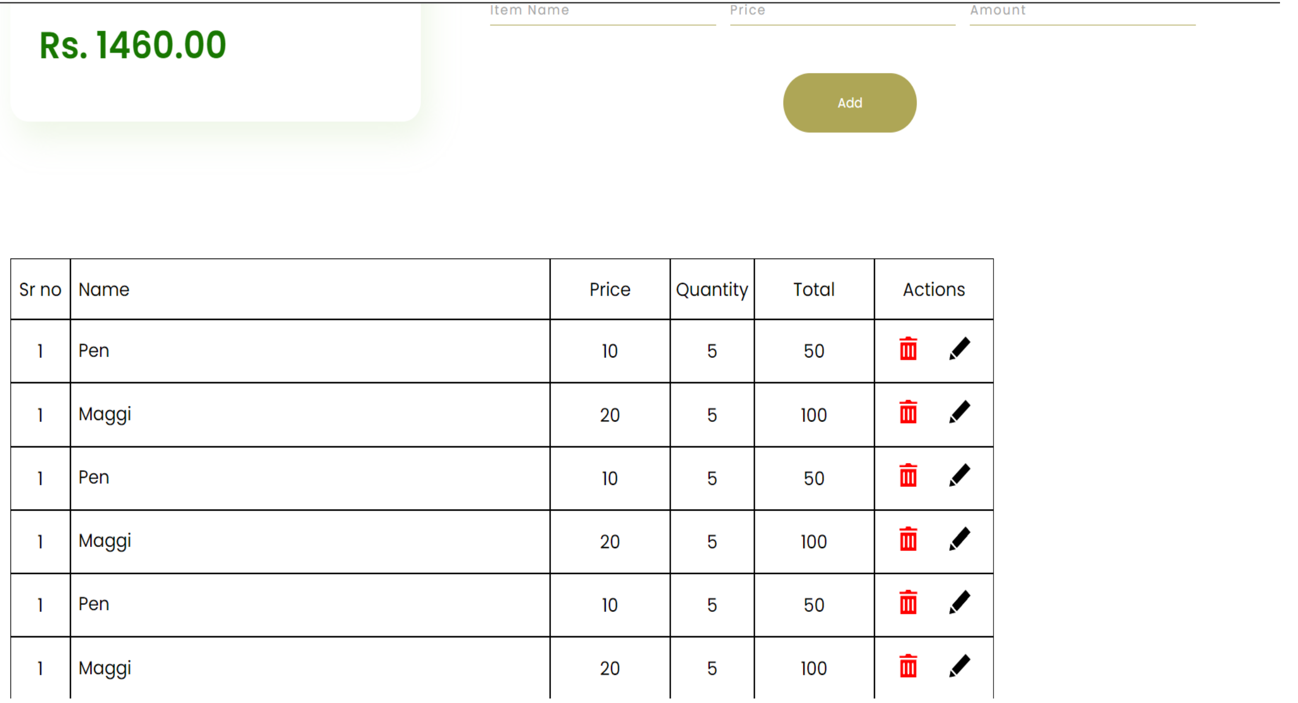
```
_id: ObjectId("616e8351f9f077981bcf5944")
name: "Maggie"
price: "20"
quantity: "2"
time: "13:16"
date: "16-10-2021"
```

History –



Sr no	Name	Price	Quantity	Total	te
1	Pen	30	2	60	-2021
1	Book	180	1	180	-2021
1	Maggie	20	2	40	15-10-2021
1	Maggie	20	2	40	16-10-2021

Entering the Data –



Rs. 1460.00

Sr no	Name	Price	Quantity	Total	Actions
1	Pen	10	5	50	🗑 ✏
1	Maggi	20	5	100	🗑 ✏
1	Pen	10	5	50	🗑 ✏
1	Maggi	20	5	100	🗑 ✏
1	Pen	10	5	50	🗑 ✏
1	Maggi	20	5	100	🗑 ✏