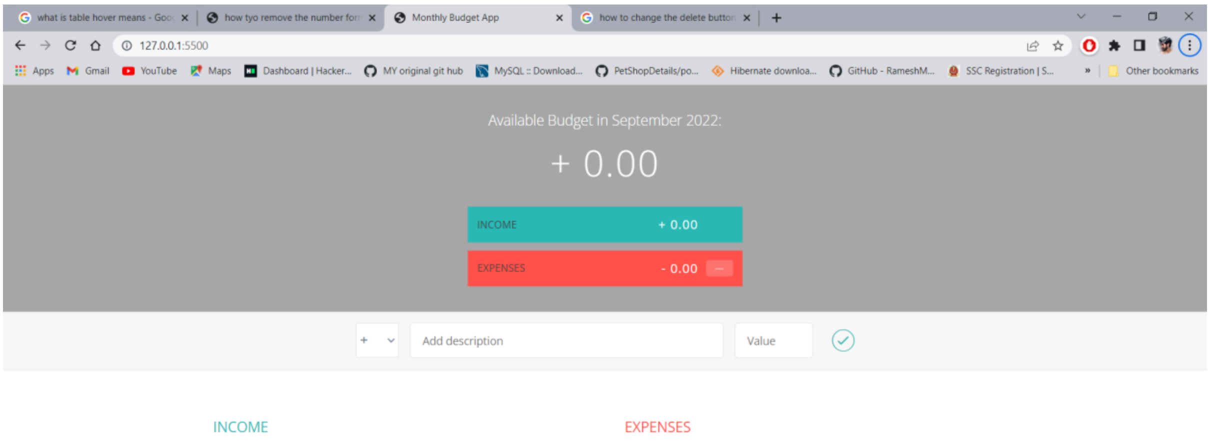
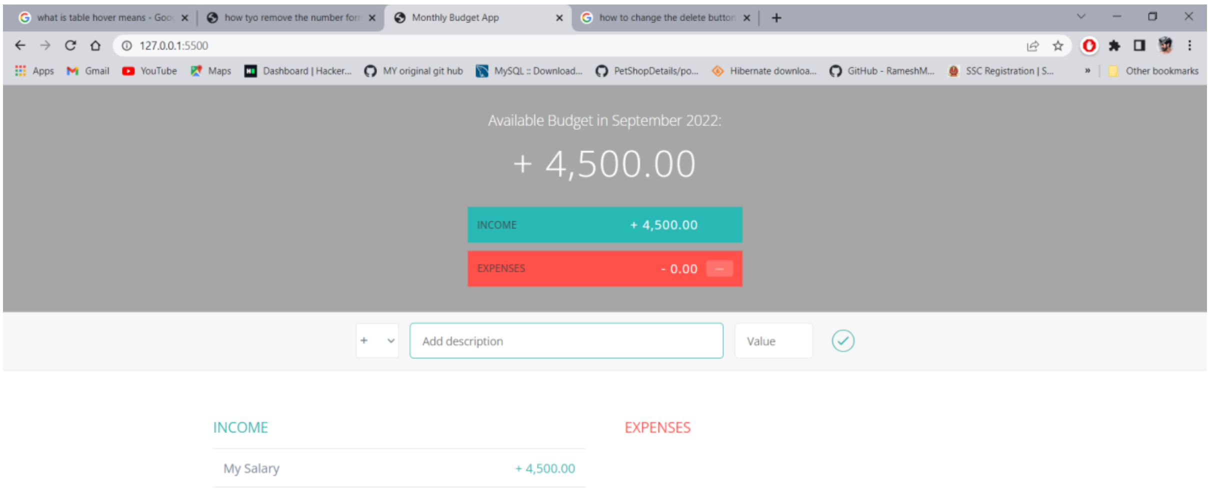
Budget App

- Created 1.Index.html in Vss
- Gave the 2.Style.css link in Index
- Gave the 3.app.js inside script tag of index

Here is our output Webpage.

We can add the income & expense .





Available Budget in September 2022:

+ 4,500.00

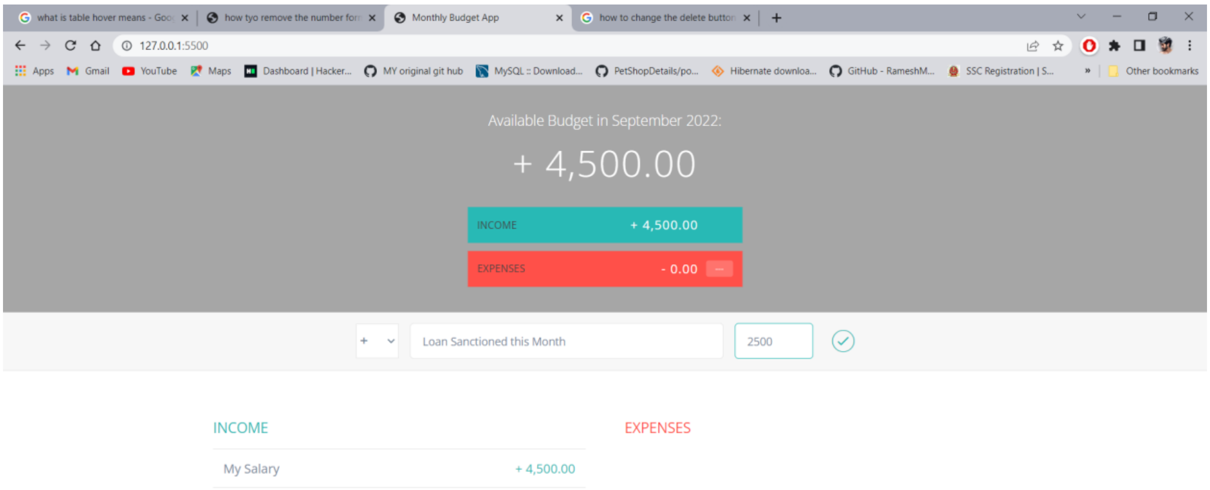
INCOME	+ 4,500.00
EXPENSES	- 0.00

+ | Add description | Value | ✓

INCOME

My Salary	+ 4,500.00

EXPENSES



Available Budget in September 2022:

+ 4,500.00

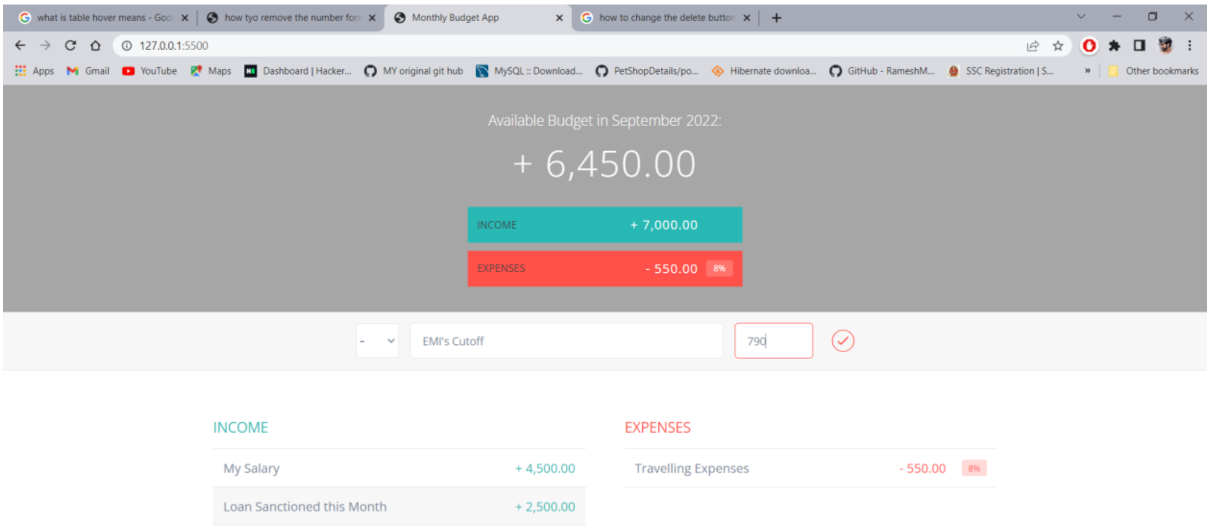
INCOME	+ 4,500.00
EXPENSES	- 0.00

+ | Loan Sanctioned this Month | 2500 | ✓

INCOME

My Salary	+ 4,500.00

EXPENSES



Available Budget in September 2022:

+ 6,450.00

INCOME	+ 7,000.00
EXPENSES	- 550.00 8%

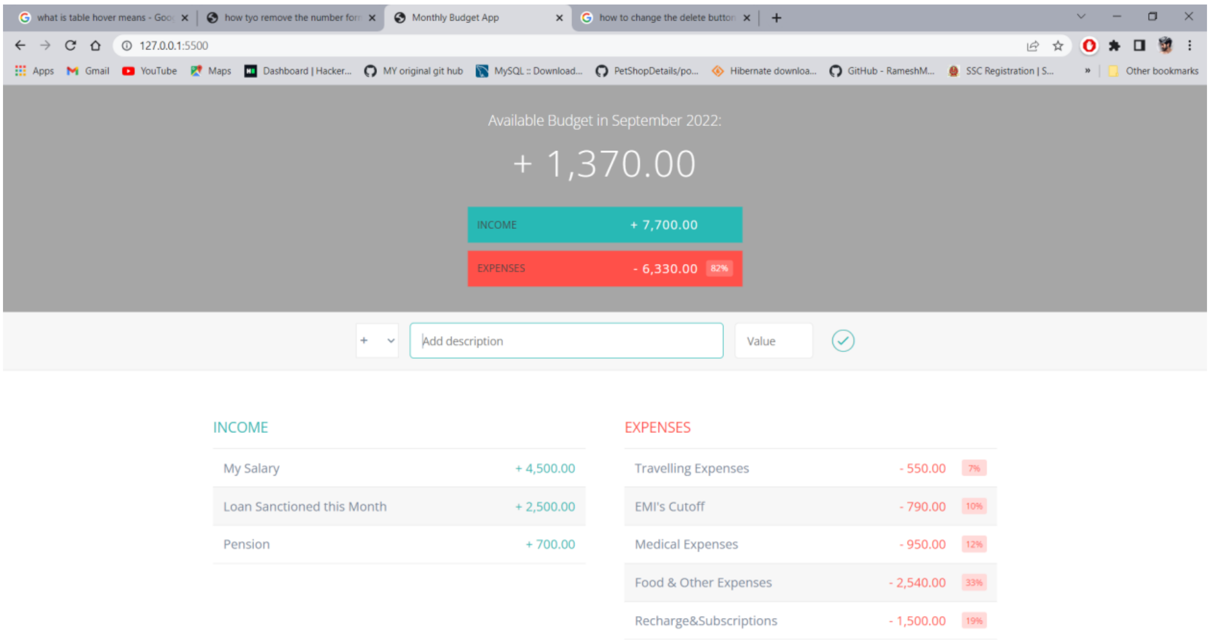
- | EMI's Cutoff | 790

INCOME

My Salary	+ 4,500.00
Loan Sanctioned this Month	+ 2,500.00

EXPENSES

Travelling Expenses	- 550.00	8%



Available Budget in September 2022:

+ 1,370.00

INCOME	+ 7,700.00
EXPENSES	- 6,330.00 82%

+ | Add description | Value

INCOME

My Salary	+ 4,500.00
Loan Sanctioned this Month	+ 2,500.00
Pension	+ 700.00

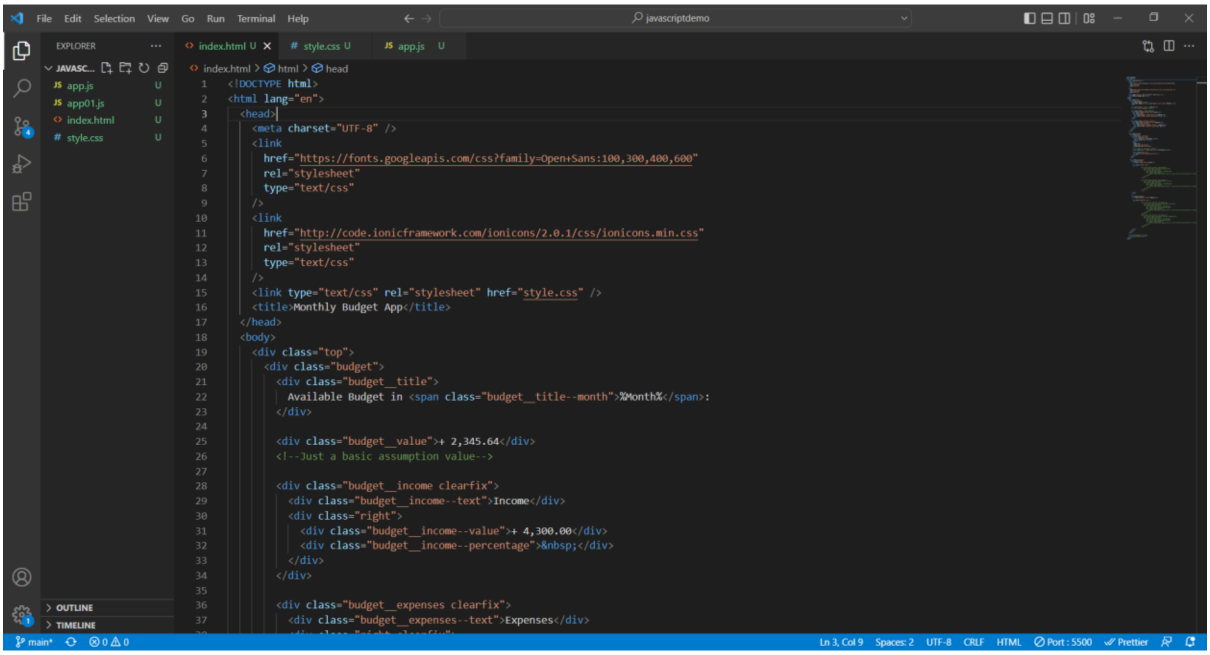
EXPENSES

Travelling Expenses	- 550.00	7%
EMI's Cutoff	- 790.00	10%
Medical Expenses	- 950.00	12%
Food & Other Expenses	- 2,540.00	33%
Recharge&Subscriptions	- 1,500.00	19%

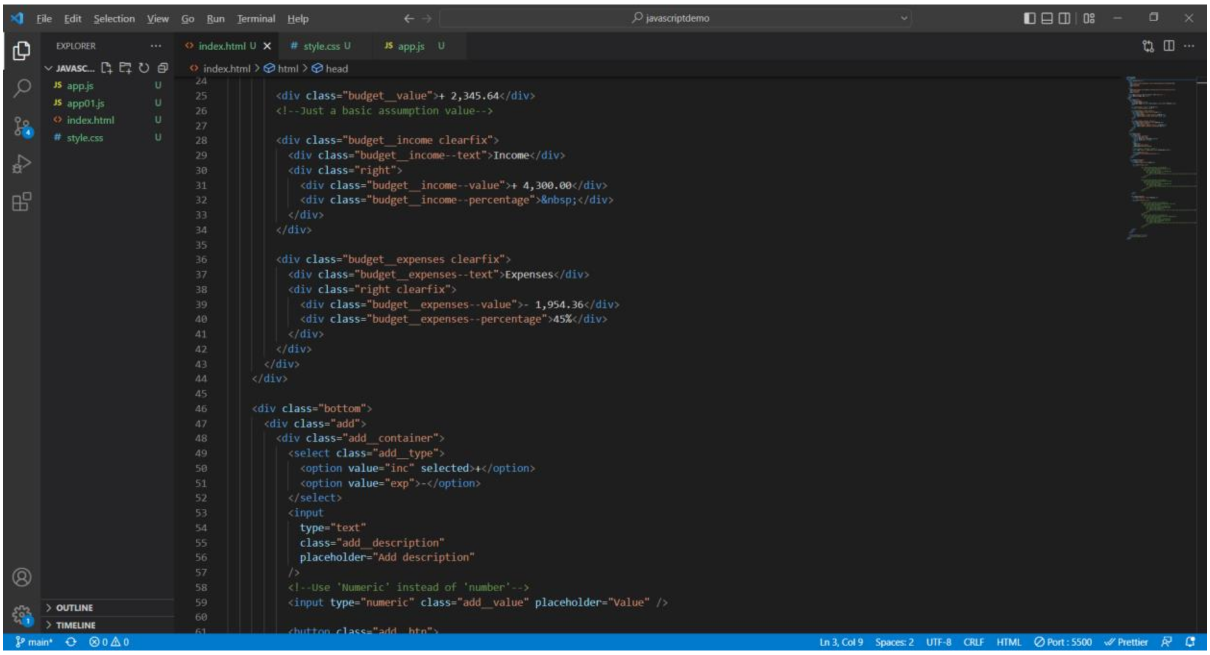
By using Cross-button When hovered over the list,

We can delete the expense or income from the list.

1. Here I attached the index.html code screen shots

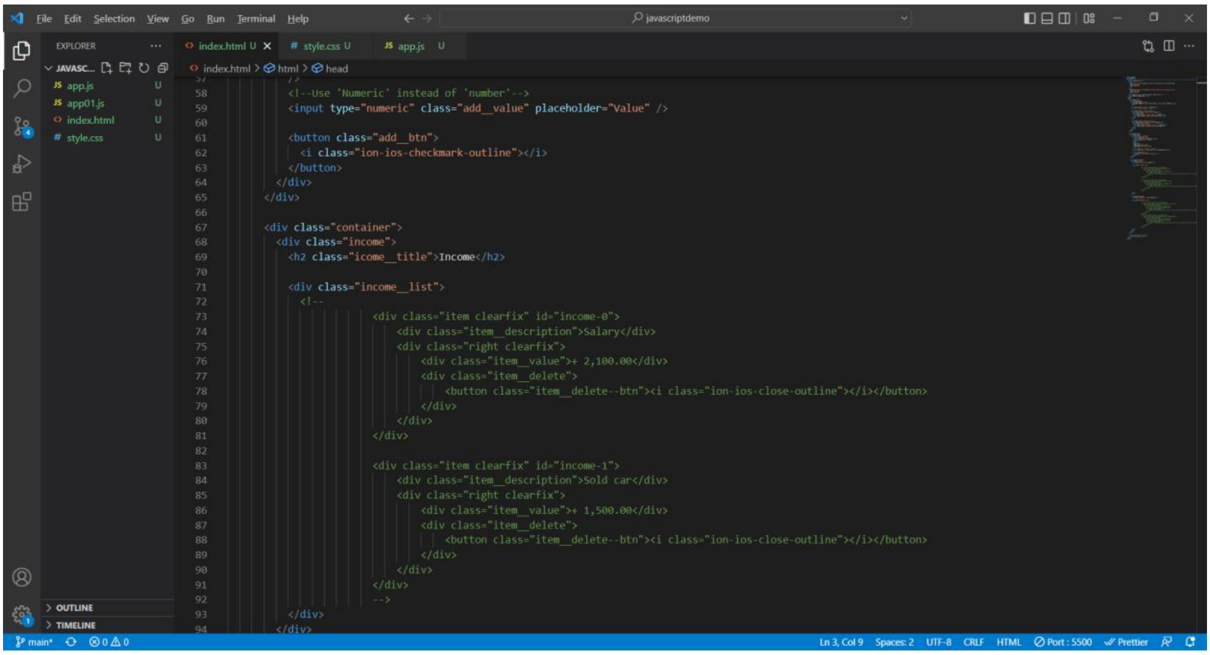


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link
      href="https://fonts.googleapis.com/css?family=Open+Sans:100,300,400,600"
      rel="stylesheet"
      type="text/css"
    />
    <link
      href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
      rel="stylesheet"
      type="text/css"
    />
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Monthly Budget App</title>
  </head>
  <body>
    <div class="top">
      <div class="budget">
        <div class="budget__title">
          Available Budget in <span class="budget__title--month">%Month%</span>:
        </div>

        <div class="budget__value">+ 2,345.64</div>
        <!--Just a basic assumption value-->

        <div class="budget__income clearfix">
          <div class="budget__income--text">Income</div>
          <div class="right">
            <div class="budget__income--value">+ 4,300.00</div>
            <div class="budget__income--percentage">&nbsp;</div>
          </div>
        </div>

        <div class="budget__expenses clearfix">
          <div class="budget__expenses--text">Expenses</div>
```

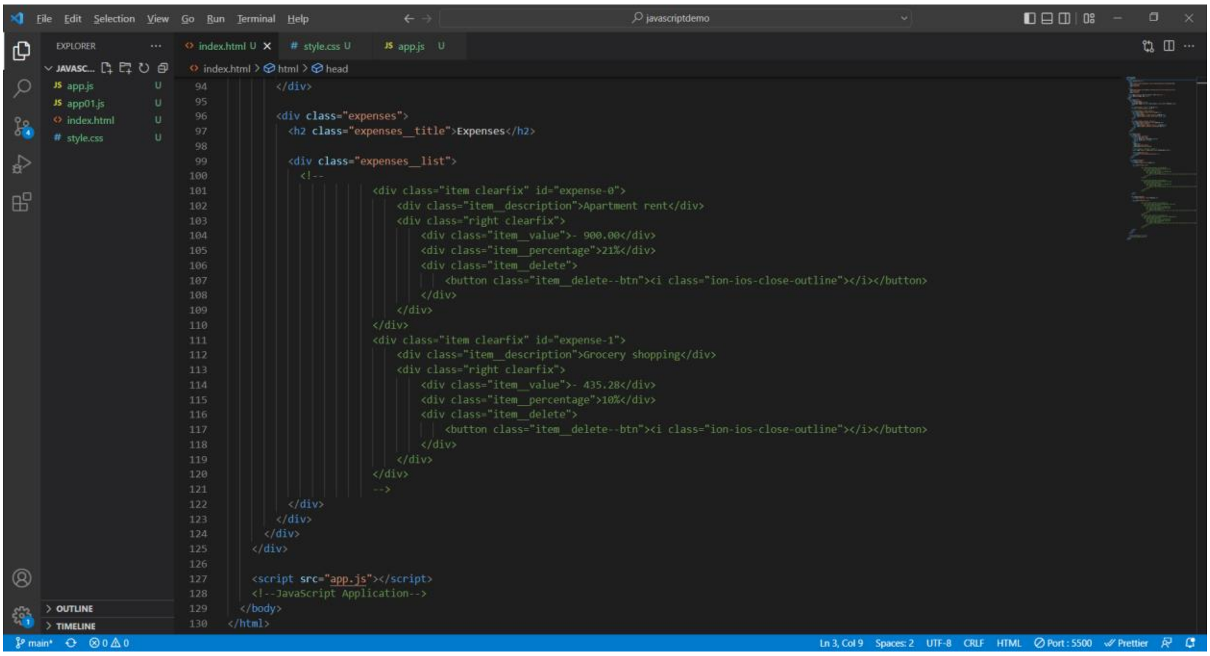


```
        <div class="budget__value">+ 2,345.64</div>
        <!--Just a basic assumption value-->

        <div class="budget__income clearfix">
          <div class="budget__income--text">Income</div>
          <div class="right">
            <div class="budget__income--value">+ 4,300.00</div>
            <div class="budget__income--percentage">&nbsp;</div>
          </div>
        </div>

        <div class="budget__expenses clearfix">
          <div class="budget__expenses--text">Expenses</div>
```

```
          <div class="right clearfix">
            <div class="budget__expenses--value">- 1,954.36</div>
            <div class="budget__expenses--percentage">45%</div>
          </div>
        </div>
      </div>
    </div>

    <div class="bottom">
      <div class="add">
        <div class="add__container">
          <select class="add__type">
            <option value="inc" selected>+</option>
            <option value="exp">-</option>
          </select>
          <input
            type="text"
            class="add__description"
            placeholder="Add description"
          />
          <!--Use 'Numeric' instead of 'number'-->
          <input type="numeric" class="add__value" placeholder="Value" />

          <button class="add__btn">
```

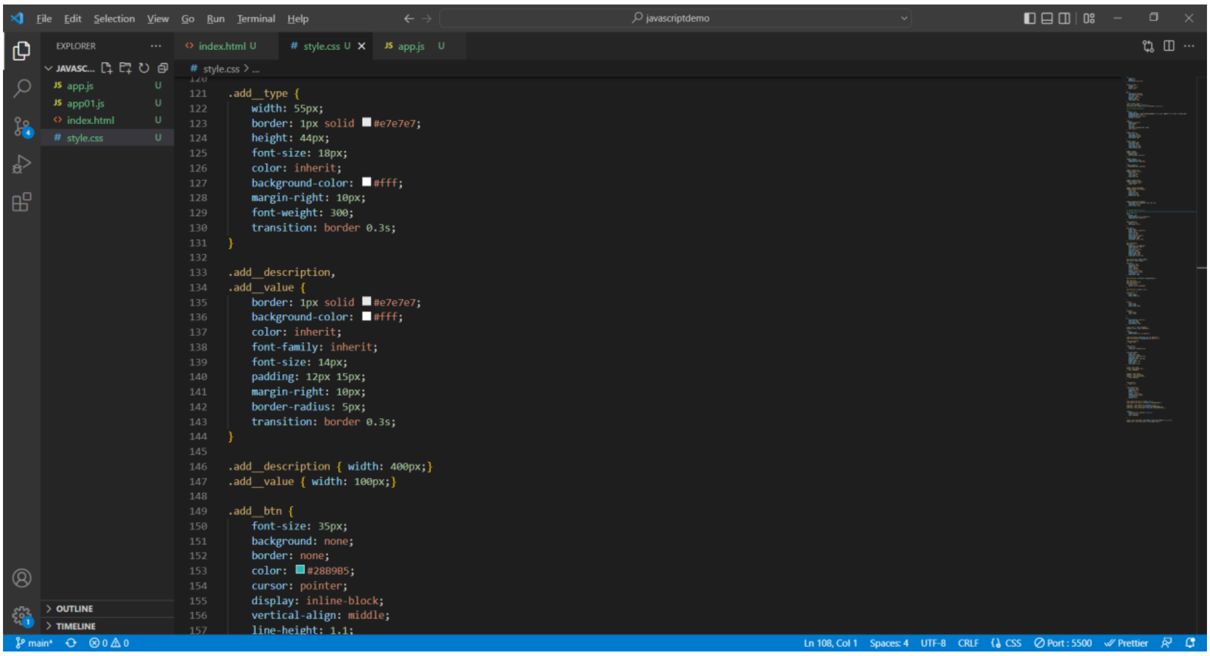


```html
          />
          <!--Use 'Numeric' instead of 'number'-->
          <input type="numeric" class="add__value" placeholder="Value" />

          <button class="add__btn">
            <i class="ion-ios-checkmark-outline"></i>
          </button>
        </div>
      </div>

      <div class="container">
        <div class="income">
          <h2 class="icome__title">Income</h2>

          <div class="income__list">
            <!--
                        <div class="item clearfix" id="income-0">
                            <div class="item__description">Salary</div>
                            <div class="right clearfix">
                                <div class="item__value">+ 2,100.00</div>
                                <div class="item__delete">
                                    <button class="item__delete--btn"><i class="ion-ios-close-outline"></i></button>
                                </div>
                            </div>
                        </div>

                        <div class="item clearfix" id="income-1">
                            <div class="item__description">Sold car</div>
                            <div class="right clearfix">
                                <div class="item__value">+ 1,500.00</div>
                                <div class="item__delete">
                                    <button class="item__delete--btn"><i class="ion-ios-close-outline"></i></button>
                                </div>
                            </div>
                        </div>
                        -->
          </div>
        </div>
```

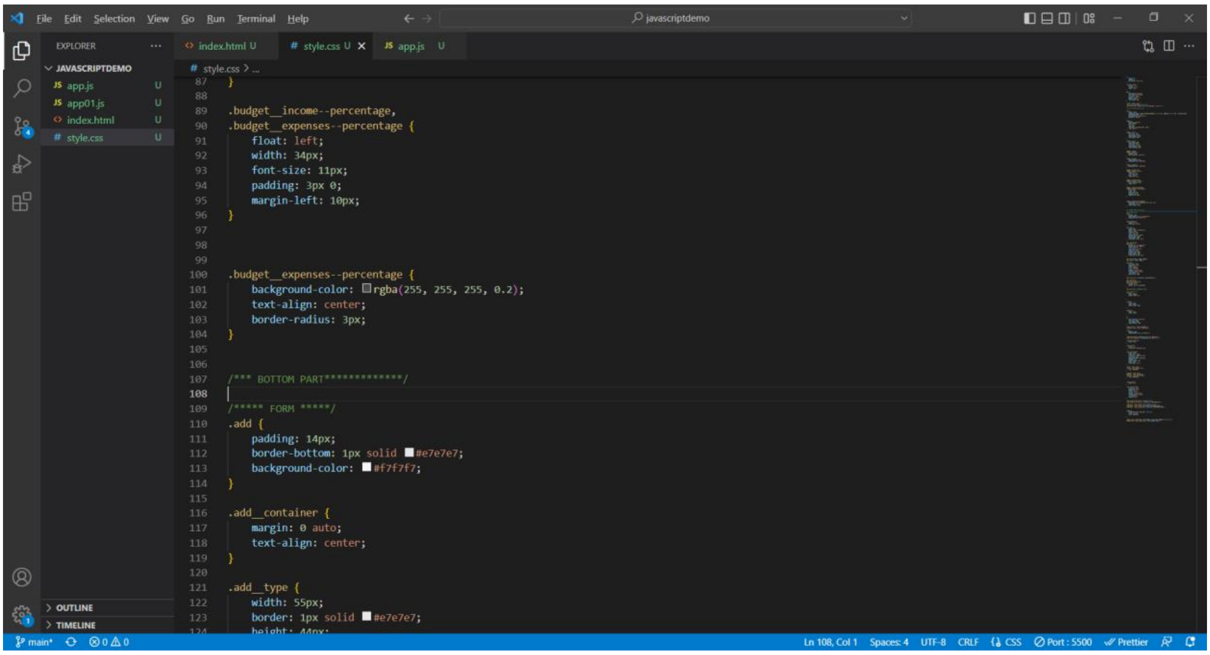


```html
        </div>

        <div class="expenses">
          <h2 class="expenses__title">Expenses</h2>

          <div class="expenses__list">
            <!--
                        <div class="item clearfix" id="expense-0">
                            <div class="item__description">Apartment rent</div>
                            <div class="right clearfix">
                                <div class="item__value">- 900.00</div>
                                <div class="item__percentage">21%</div>
                                <div class="item__delete">
                                    <button class="item__delete--btn"><i class="ion-ios-close-outline"></i></button>
                                </div>
                            </div>
                        </div>
                        <div class="item clearfix" id="expense-1">
                            <div class="item__description">Grocery shopping</div>
                            <div class="right clearfix">
                                <div class="item__value">- 435.28</div>
                                <div class="item__percentage">10%</div>
                                <div class="item__delete">
                                    <button class="item__delete--btn"><i class="ion-ios-close-outline"></i></button>
                                </div>
                            </div>
                        </div>
                        -->
          </div>
        </div>
      </div>
    </div>

    <script src="app.js"></script>
    <!--JavaScript Application-->
  </body>
</html>
```

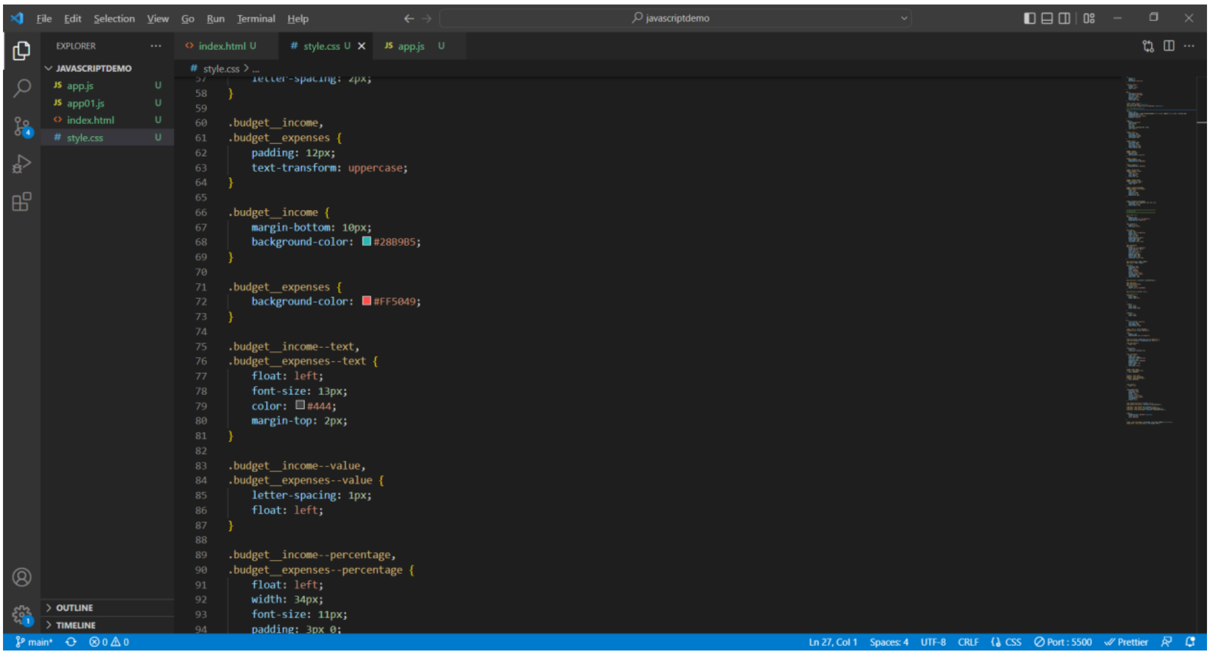
Here I added only few screenshots of style.css code Because File Exceeding limit . So, I added (full code) the source code in another pdf.



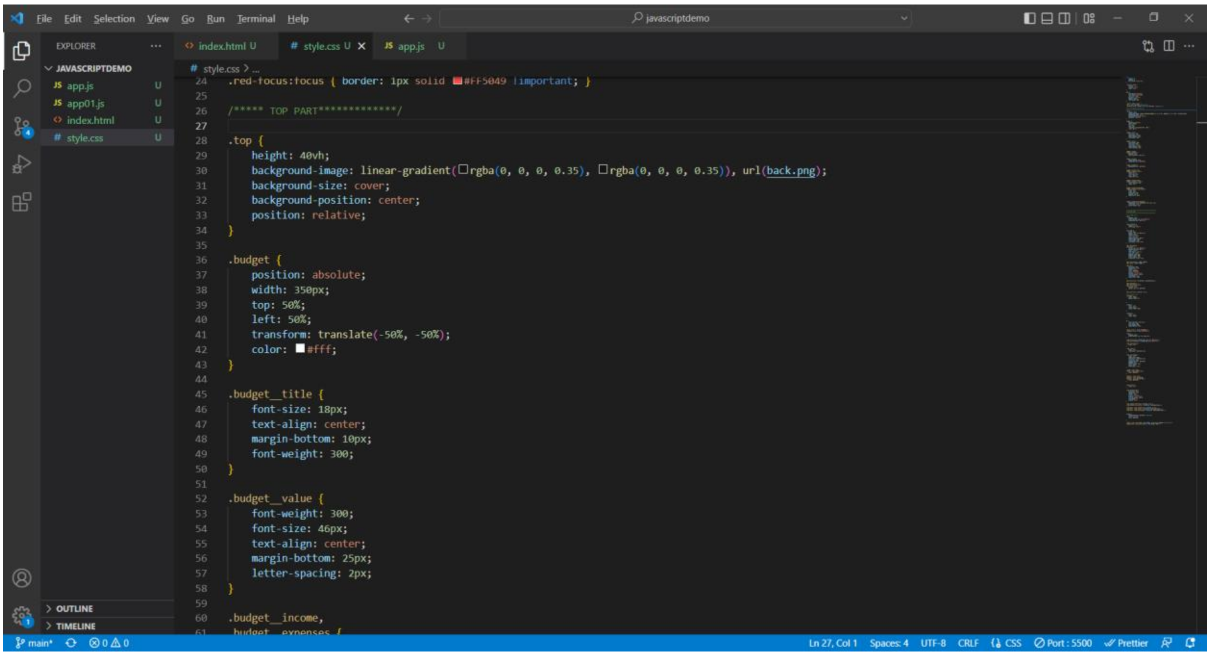
```css
.add__type {
    width: 55px;
    border: 1px solid #e7e7e7;
    height: 44px;
    font-size: 18px;
    color: inherit;
    background-color: #fff;
    margin-right: 10px;
    font-weight: 300;
    transition: border 0.3s;
}

.add__description,
.add__value {
    border: 1px solid #e7e7e7;
    background-color: #fff;
    color: inherit;
    font-family: inherit;
    font-size: 14px;
    padding: 12px 15px;
    margin-right: 10px;
    border-radius: 5px;
    transition: border 0.3s;
}

.add__description { width: 400px;}
.add__value { width: 100px;}

.add__btn {
    font-size: 35px;
    background: none;
    border: none;
    color: #28B9B5;
    cursor: pointer;
    display: inline-block;
    vertical-align: middle;
    line-height: 1.1;
```



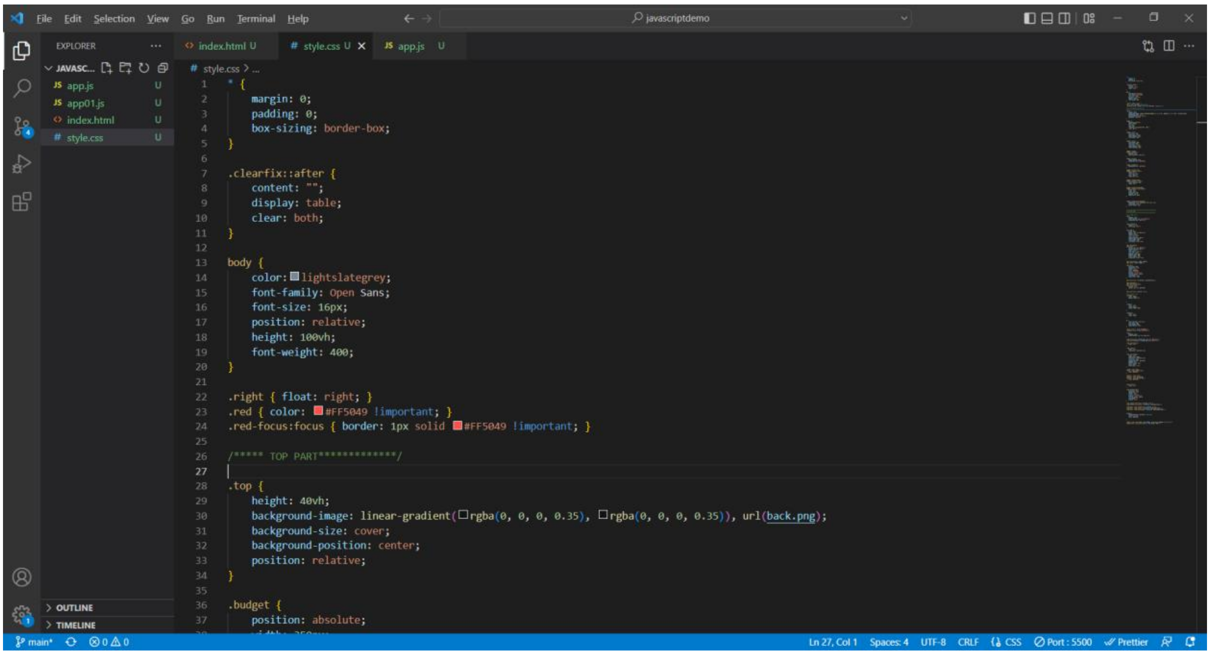
```css
}

.budget__income--percentage,
.budget__expenses--percentage {
    float: left;
    width: 34px;
    font-size: 11px;
    padding: 3px 0;
    margin-left: 10px;
}



.budget__expenses--percentage {
    background-color: rgba(255, 255, 255, 0.2);
    text-align: center;
    border-radius: 3px;
}


/*** BOTTOM PART*************/

/***** FORM *****/
.add {
    padding: 14px;
    border-bottom: 1px solid #e7e7e7;
    background-color: #f7f7f7;
}

.add__container {
    margin: 0 auto;
    text-align: center;
}

.add__type {
    width: 55px;
    border: 1px solid #e7e7e7;
    height: 44px;
```



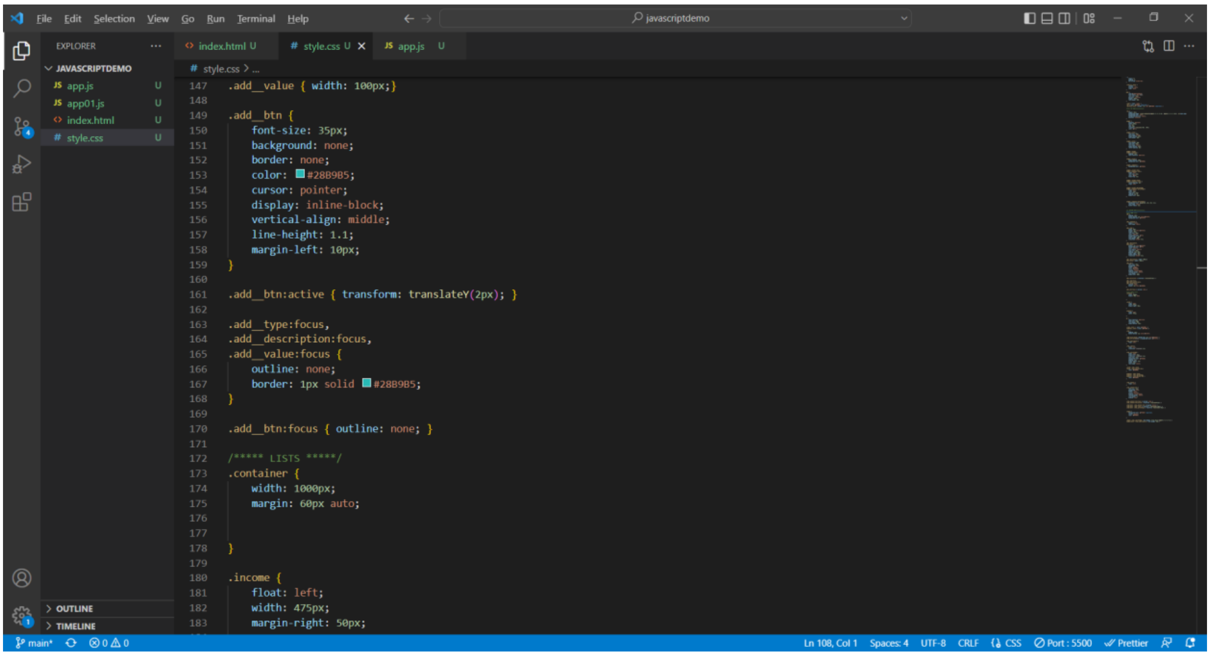
```css
    letter-spacing: 2px;
}

.budget__income,
.budget__expenses {
    padding: 12px;
    text-transform: uppercase;
}

.budget__income {
    margin-bottom: 10px;
    background-color: #28B9B5;
}

.budget__expenses {
    background-color: #FF5049;
}

.budget__income--text,
.budget__expenses--text {
    float: left;
    font-size: 13px;
    color: #444;
    margin-top: 2px;
}

.budget__income--value,
.budget__expenses--value {
    letter-spacing: 1px;
    float: left;
}

.budget__income--percentage,
.budget__expenses--percentage {
    float: left;
    width: 34px;
    font-size: 11px;
    padding: 3px 0;
```



```css
.red-focus:focus { border: 1px solid #FF5049 !important; }

/***** TOP PART*************/

.top {
    height: 40vh;
    background-image: linear-gradient(rgba(0, 0, 0, 0.35), rgba(0, 0, 0, 0.35)), url(back.png);
    background-size: cover;
    background-position: center;
    position: relative;
}

.budget {
    position: absolute;
    width: 350px;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    color: #fff;
}

.budget__title {
    font-size: 18px;
    text-align: center;
    margin-bottom: 10px;
    font-weight: 300;
}

.budget__value {
    font-weight: 300;
    font-size: 46px;
    text-align: center;
    margin-bottom: 25px;
    letter-spacing: 2px;
}

.budget__income,
.budget__expenses {
```

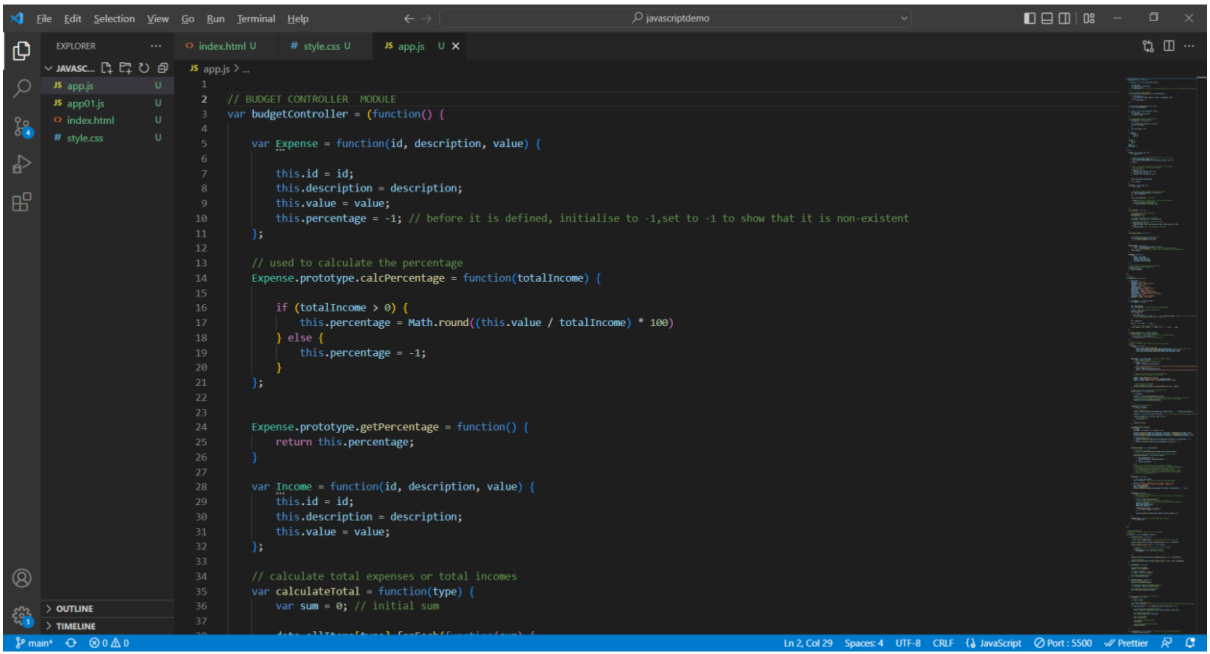


```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.clearfix::after {
    content: "";
    display: table;
    clear: both;
}

body {
    color: lightslategrey;
    font-family: Open Sans;
    font-size: 16px;
    position: relative;
    height: 100vh;
    font-weight: 400;
}

.right { float: right; }
.red { color: #FF5049 !important; }
.red-focus:focus { border: 1px solid #FF5049 !important; }

/***** TOP PART*************/

.top {
    height: 40vh;
    background-image: linear-gradient(rgba(0, 0, 0, 0.35), rgba(0, 0, 0, 0.35)), url(back.png);
    background-size: cover;
    background-position: center;
    position: relative;
}

.budget {
    position: absolute;
```

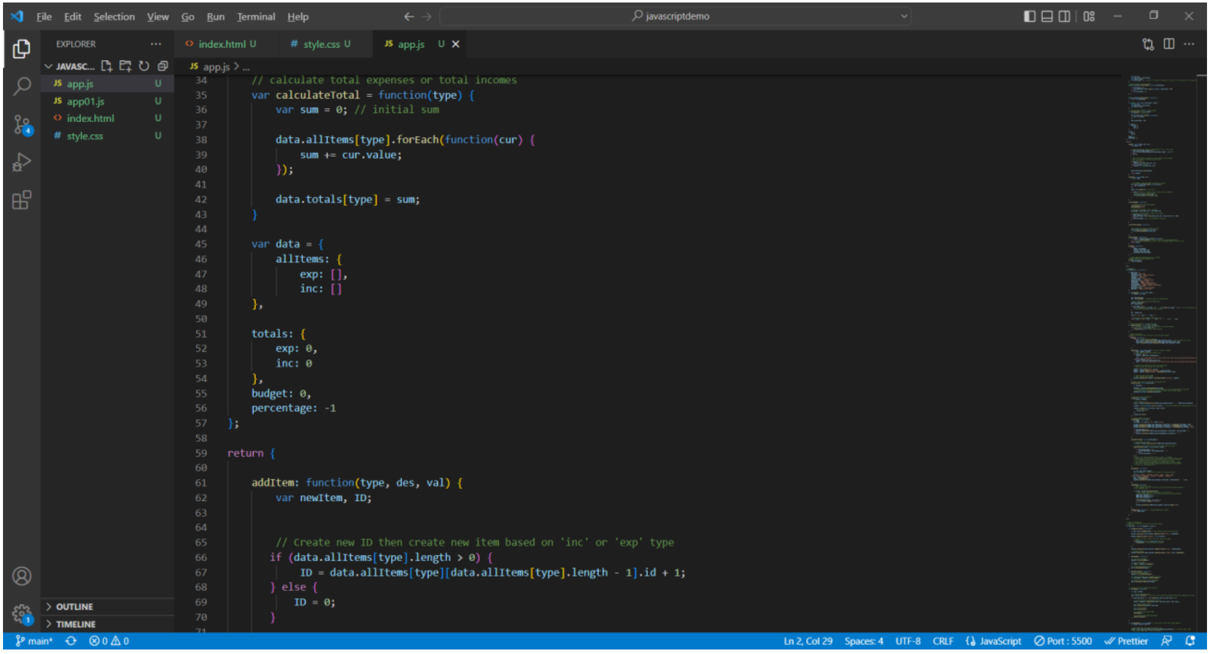


```css
.add__value { width: 100px;}

.add__btn {
    font-size: 35px;
    background: none;
    border: none;
    color: #28B9B5;
    cursor: pointer;
    display: inline-block;
    vertical-align: middle;
    line-height: 1.1;
    margin-left: 10px;
}

.add__btn:active { transform: translateY(2px); }

.add__type:focus,
.add__description:focus,
.add__value:focus {
    outline: none;
    border: 1px solid #28B9B5;
}

.add__btn:focus { outline: none; }

/***** LISTS *****/
.container {
    width: 1000px;
    margin: 60px auto;


}

.income {
    float: left;
    width: 475px;
    margin-right: 50px;
```

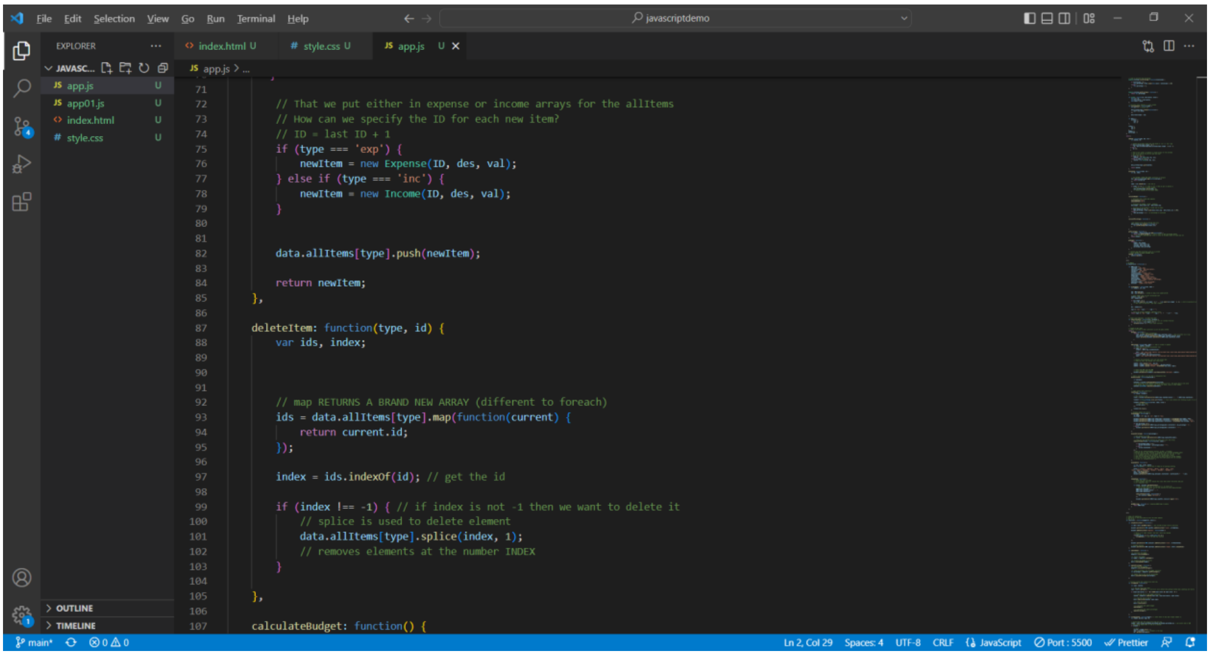
Here I added few Screenshots of app.js(javascript code) Due to exceeding file size limit . So I added the full code in another pdf.



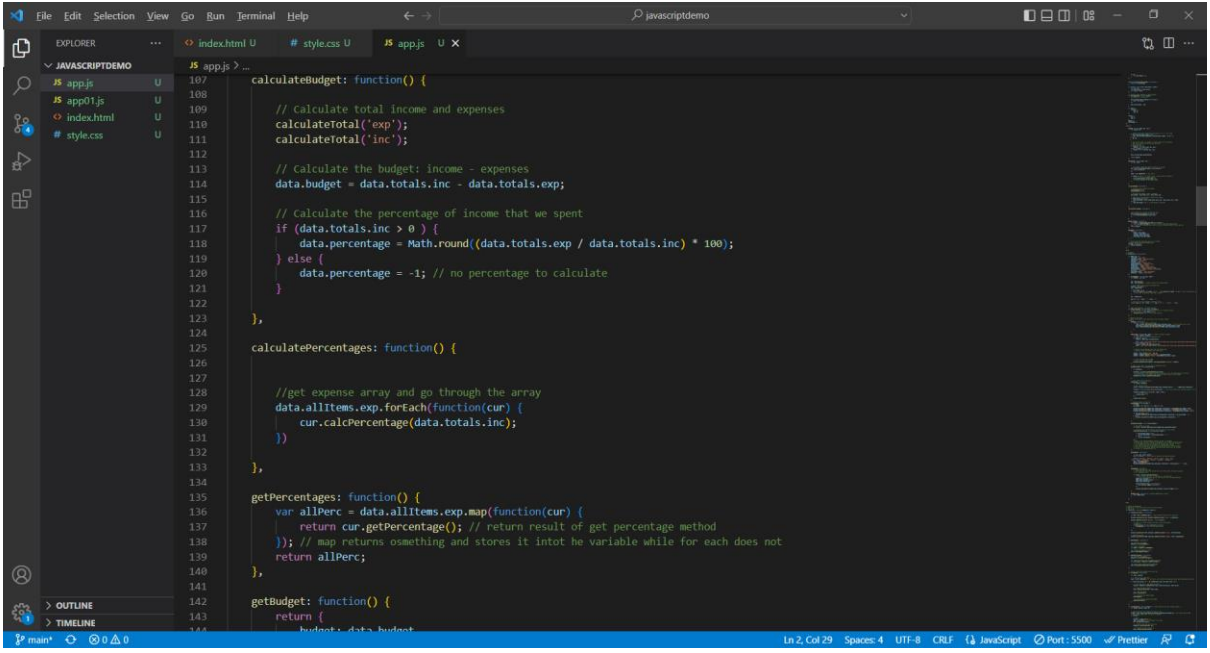
```javascript

// BUDGET CONTROLLER  MODULE
var budgetController = (function() {

    var Expense = function(id, description, value) {

        this.id = id;
        this.description = description;
        this.value = value;
        this.percentage = -1; // before it is defined, initialise to -1,set to -1 to show that it is non-existent
    };

    // used to calculate the percentage
    Expense.prototype.calcPercentage = function(totalIncome) {

        if (totalIncome > 0) {
            this.percentage = Math.round((this.value / totalIncome) * 100)
        } else {
            this.percentage = -1;
        }
    };


    Expense.prototype.getPercentage = function() {
        return this.percentage;
    }

    var Income = function(id, description, value) {
        this.id = id;
        this.description = description;
        this.value = value;
    };

    // calculate total expenses or total incomes
    var calculateTotal = function(type) {
        var sum = 0; // initial sum

```



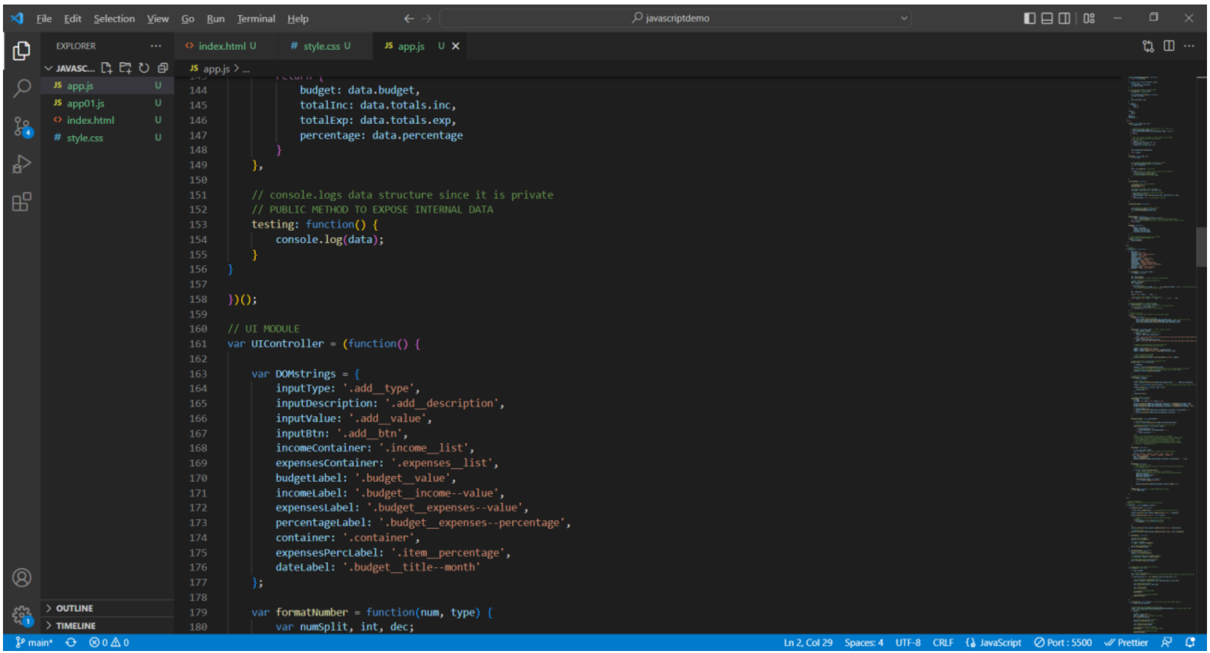
```javascript
    // calculate total expenses or total incomes
    var calculateTotal = function(type) {
        var sum = 0; // initial sum

        data.allItems[type].forEach(function(cur) {
            sum += cur.value;
        });

        data.totals[type] = sum;
    }

    var data = {
        allItems: {
            exp: [],
            inc: []
    },

    totals: {
        exp: 0,
        inc: 0
    },
    budget: 0,
    percentage: -1
};

return {

    addItem: function(type, des, val) {
        var newItem, ID;


        // Create new ID then create new item based on 'inc' or 'exp' type
       if (data.allItems[type].length > 0) {
            ID = data.allItems[type][data.allItems[type].length - 1].id + 1;
       } else {
           ID = 0;
       }
```



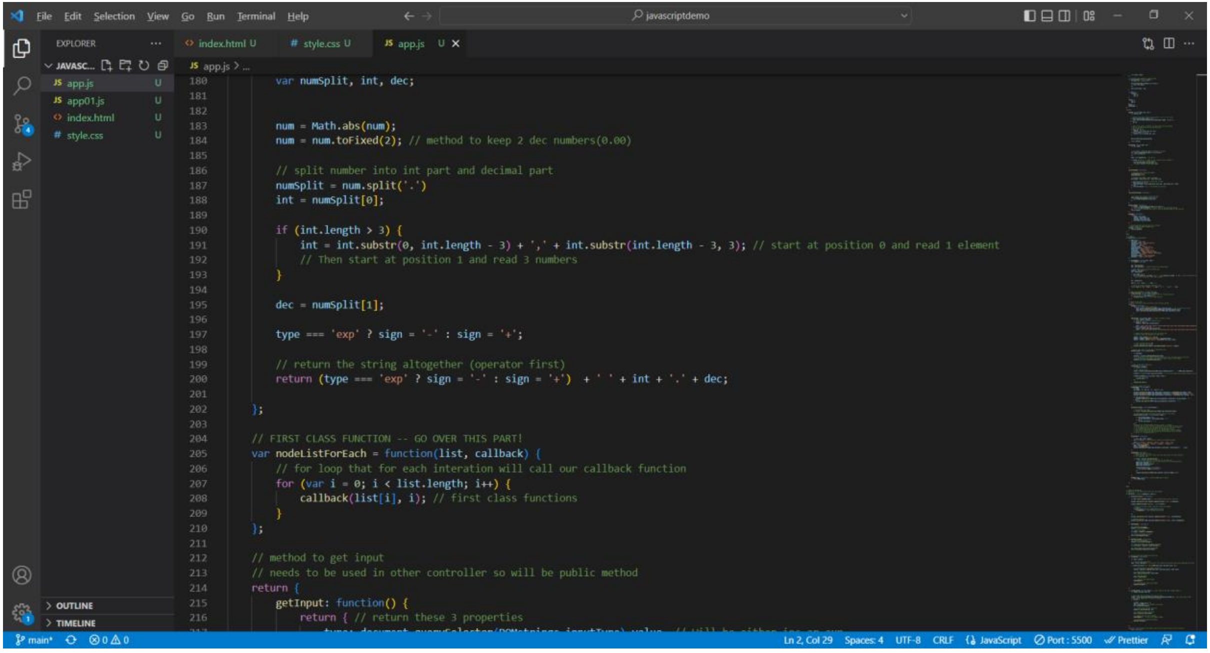
```
        // That we put either in expense or income arrays for the allItems
        // How can we specify the ID for each new item?
        // ID = last ID + 1
        if (type === 'exp') {
            newItem = new Expense(ID, des, val);
        } else if (type === 'inc') {
            newItem = new Income(ID, des, val);
        }


        data.allItems[type].push(newItem);

        return newItem;
    },

    deleteItem: function(type, id) {
        var ids, index;



        // map RETURNS A BRAND NEW ARRAY (different to foreach)
        ids = data.allItems[type].map(function(current) {
            return current.id;
        });

        index = ids.indexOf(id); // get the id

        if (index !== -1) { // if index is not -1 then we want to delete it
            // splice is used to delete element
            data.allItems[type].splice(index, 1);
            // removes elements at the number INDEX
        }

    },

    calculateBudget: function() {
```



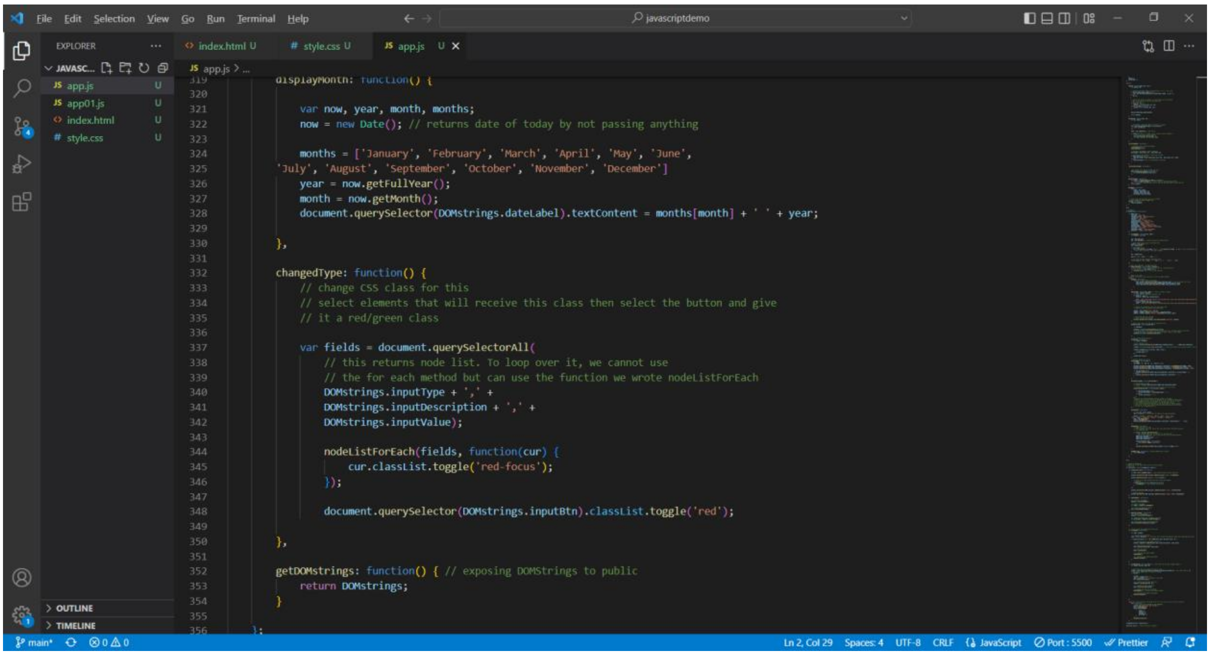
```
    calculateBudget: function() {

        // Calculate total income and expenses
        calculateTotal('exp');
        calculateTotal('inc');

        // Calculate the budget: income - expenses
        data.budget = data.totals.inc - data.totals.exp;

        // Calculate the percentage of income that we spent
        if (data.totals.inc > 0 ) {
            data.percentage = Math.round((data.totals.exp / data.totals.inc) * 100);
        } else {
            data.percentage = -1; // no percentage to calculate
        }

    },

    calculatePercentages: function() {


        //get expense array and go through the array
        data.allItems.exp.forEach(function(cur) {
            cur.calcPercentage(data.totals.inc);
        })

    },

    getPercentages: function() {
        var allPerc = data.allItems.exp.map(function(cur) {
            return cur.getPercentage(); // return result of get percentage method
        }); // map returns osmething and stores it intot he variable while for each does not
        return allPerc;
    },

    getBudget: function() {
        return {
            budget: data.budget
```



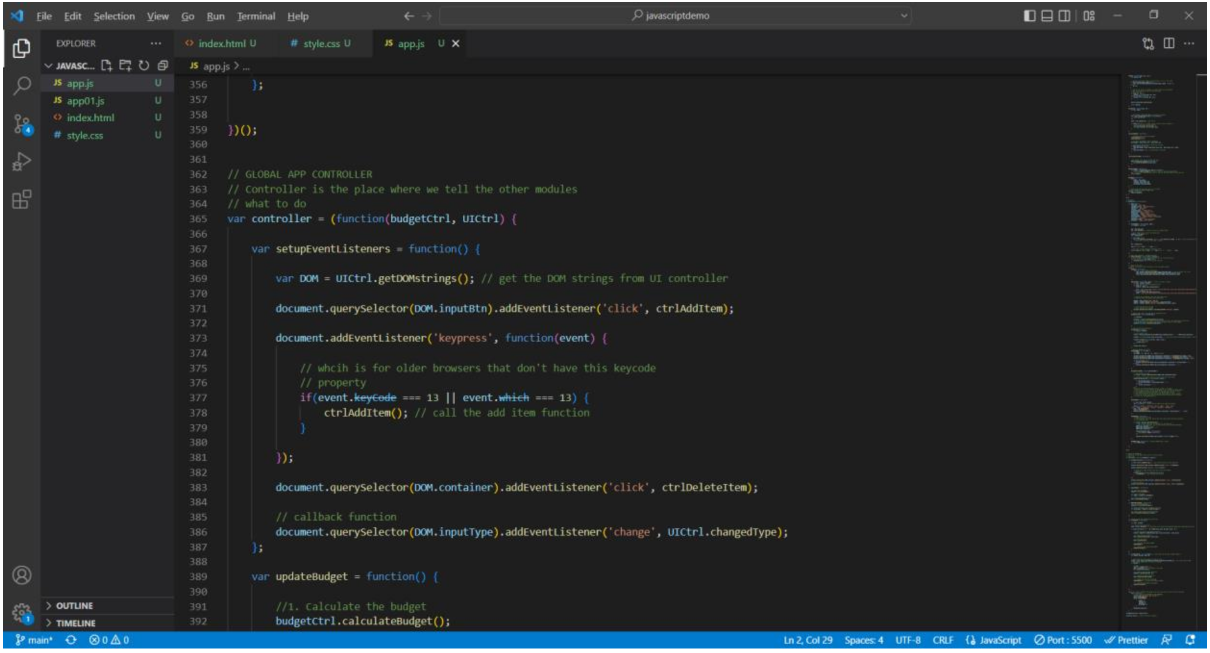
```
            budget: data.budget,
            totalInc: data.totals.inc,
            totalExp: data.totals.exp,
            percentage: data.percentage
        }
    },

    // console.logs data structure since it is private
    // PUBLIC METHOD TO EXPOSE INTERNAL DATA
    testing: function() {
        console.log(data);
    }
}

})();

// UI MODULE
var UIController = (function() {

    var DOMstrings = {
        inputType: '.add__type',
        inputDescription: '.add__description',
        inputValue: '.add__value',
        inputBtn: '.add__btn',
        incomeContainer: '.income__list',
        expensesContainer: '.expenses__list',
        budgetLabel: '.budget__value',
        incomeLabel: '.budget__income--value',
        expensesLabel: '.budget__expenses--value',
        percentageLabel: '.budget__expenses--percentage',
        container: '.container',
        expensesPercLabel: '.item__percentage',
        dateLabel: '.budget__title--month'
    };

    var formatNumber = function(num, type) {
        var numSplit, int, dec;
```



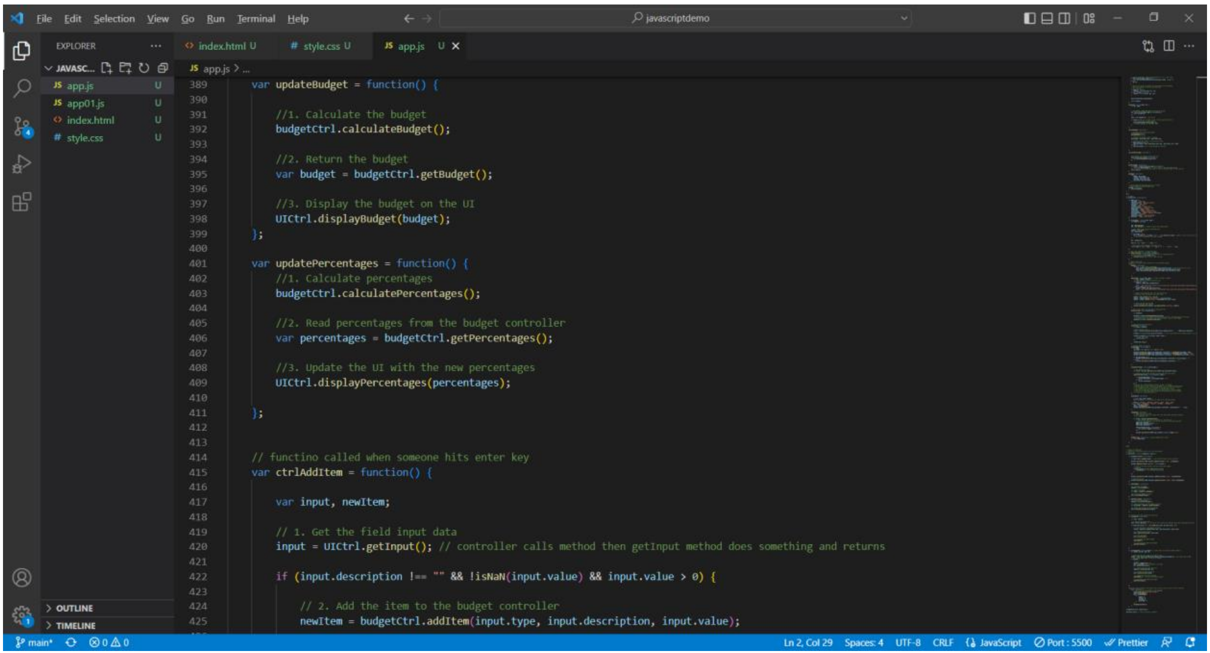
```
        var numSplit, int, dec;


        num = Math.abs(num);
        num = num.toFixed(2); // method to keep 2 dec numbers(0.00)

        // split number into int part and decimal part
        numSplit = num.split('.')
        int = numSplit[0];

        if (int.length > 3) {
            int = int.substr(0, int.length - 3) + ',' + int.substr(int.length - 3, 3); // start at position 0 and read 1 element
            // Then start at position 1 and read 3 numbers
        }

        dec = numSplit[1];

        type === 'exp' ? sign = '-' : sign = '+';

        // return the string altogether (operator first)
        return (type === 'exp' ? sign = '-' : sign = '+')  + ' ' + int + '.' + dec;

    };

    // FIRST CLASS FUNCTION -- GO OVER THIS PART!
    var nodeListForEach = function(list, callback) {
        // for loop that for each interation will call our callback function
        for (var i = 0; i < list.length; i++) {
            callback(list[i], i); // first class functions
        }
    };

    // method to get input
    // needs to be used in other controller so will be public method
    return {
        getInput: function() {
            return { // return these 3 properties
```



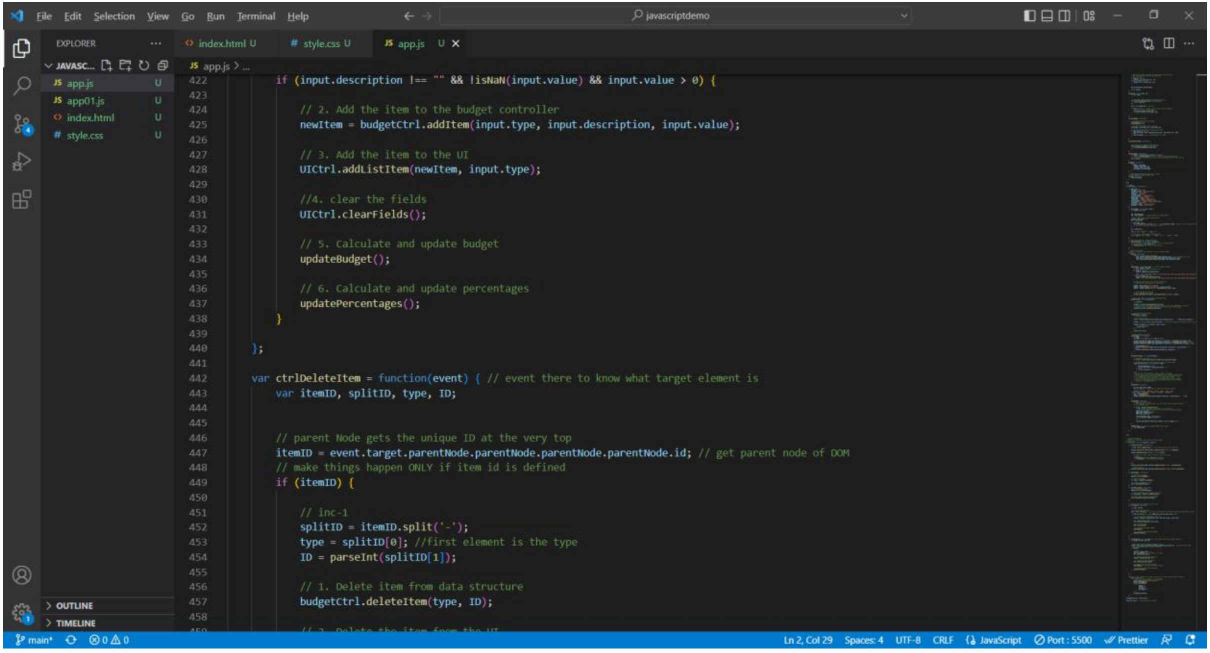
```
        displayMonth: function() {

            var now, year, month, months;
            now = new Date(); // returns date of today by not passing anything

            months = ['January', 'February', 'March', 'April', 'May', 'June',
        'July', 'August', 'September', 'October', 'November', 'December']
            year = now.getFullYear();
            month = now.getMonth();
            document.querySelector(DOMstrings.dateLabel).textContent = months[month] + ' ' + year;

        },

        changedType: function() {
            // change CSS class for this
            // select elements that will receive this class then select the button and give
            // it a red/green class

            var fields = document.querySelectorAll(
                // this returns node list. To loop over it, we cannot use
                // the for each method but can use the function we wrote nodeListForEach
                DOMstrings.inputType + ',' +
                DOMstrings.inputDescription + ',' +
                DOMstrings.inputValue);

                nodeListForEach(fields, function(cur) {
                    cur.classList.toggle('red-focus');
                });

                document.querySelector(DOMstrings.inputBtn).classList.toggle('red');

        },

        getDOMstrings: function() { // exposing DOMStrings to public
            return DOMstrings;
        }

    };
```



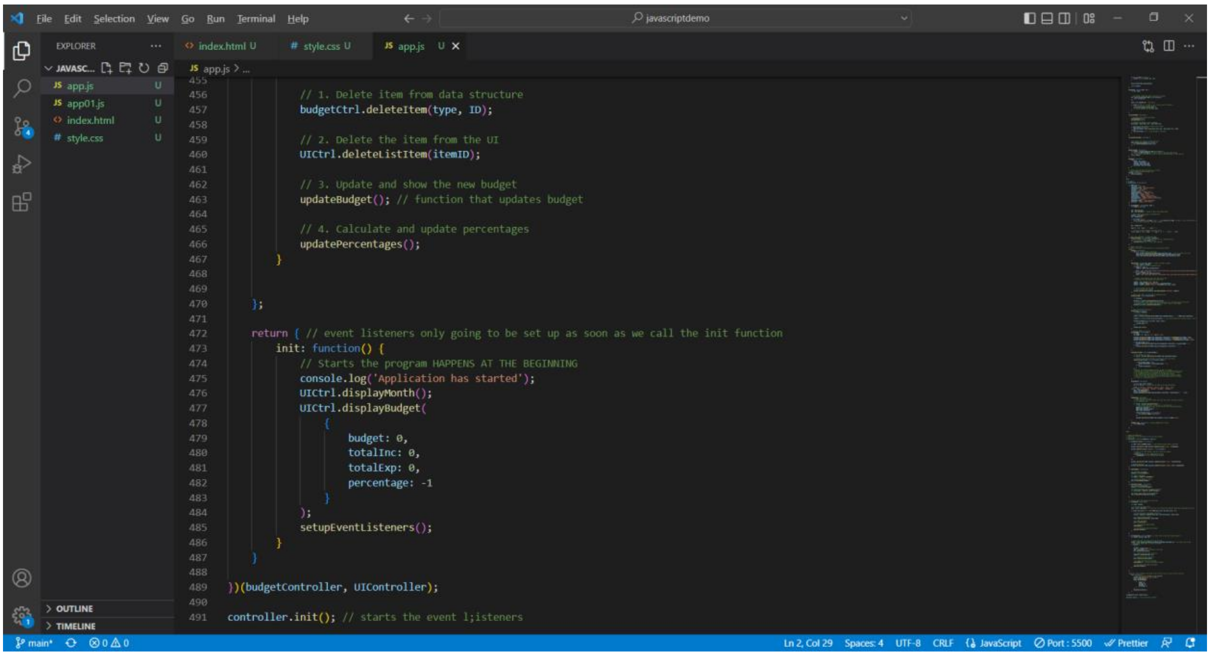
```
    };


})();


// GLOBAL APP CONTROLLER
// Controller is the place where we tell the other modules
// what to do
var controller = (function(budgetCtrl, UICtrl) {

    var setupEventListeners = function() {

        var DOM = UICtrl.getDOMstrings(); // get the DOM strings from UI controller

        document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

        document.addEventListener('keypress', function(event) {

            // whcih is for older browsers that don't have this keycode
            // property
            if(event.~~keyCode~~ === 13 || event.~~which~~ === 13) {
                ctrlAddItem(); // call the add item function
            }

        });

        document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

        // callback function
        document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
    };

    var updateBudget = function() {

        //1. Calculate the budget
        budgetCtrl.calculateBudget();
```



```
    var updateBudget = function() {

        //1. Calculate the budget
        budgetCtrl.calculateBudget();

        //2. Return the budget
        var budget = budgetCtrl.getBudget();

        //3. Display the budget on the UI
        UICtrl.displayBudget(budget);
    };

    var updatePercentages = function() {
        //1. Calculate percentages
        budgetCtrl.calculatePercentages();

        //2. Read percentages from the budget controller
        var percentages = budgetCtrl.getPercentages();

        //3. Update the UI with the new percentages
        UICtrl.displayPercentages(percentages);

    };


    // functino called when someone hits enter key
    var ctrlAddItem = function() {

        var input, newItem;

        // 1. Get the field input data
        input = UICtrl.getInput(); // controller calls method then getInput method does something and returns

        if (input.description !== "" && !isNaN(input.value) && input.value > 0) {

            // 2. Add the item to the budget controller
            newItem = budgetCtrl.addItem(input.type, input.description, input.value);
```



```
        if (input.description !== "" && !isNaN(input.value) && input.value > 0) {

            // 2. Add the item to the budget controller
            newItem = budgetCtrl.addItem(input.type, input.description, input.value);

            // 3. Add the item to the UI
            UICtrl.addListItem(newItem, input.type);

            //4. clear the fields
            UICtrl.clearFields();

            // 5. Calculate and update budget
            updateBudget();

            // 6. Calculate and update percentages
            updatePercentages();
        }

    };

    var ctrlDeleteItem = function(event) { // event there to know what target element is
        var itemID, splitID, type, ID;


        // parent Node gets the unique ID at the very top
        itemID = event.target.parentNode.parentNode.parentNode.parentNode.id; // get parent node of DOM
        // make things happen ONLY if item id is defined
        if (itemID) {

            // inc-1
            splitID = itemID.split('-');
            type = splitID[0]; //first element is the type
            ID = parseInt(splitID[1]);

            // 1. Delete item from data structure
            budgetCtrl.deleteItem(type, ID);

            // 2. Delete the item from the UI
```



```javascript

            // 1. Delete item from data structure
            budgetCtrl.deleteItem(type, ID);

            // 2. Delete the item from the UI
            UICtrl.deleteListItem(itemID);

            // 3. Update and show the new budget
            updateBudget(); // function that updates budget

            // 4. Calculate and update percentages
            updatePercentages();
        }


    };

    return { // event listeners only going to be set up as soon as we call the init function
        init: function() {
            // Starts the program HAPPENS AT THE BEGINNING
            console.log('Application has started');
            UICtrl.displayMonth();
            UICtrl.displayBudget(
                {
                    budget: 0,
                    totalInc: 0,
                    totalExp: 0,
                    percentage: -1
                }
            );
            setupEventListeners();
        }
    }

})(budgetController, UIController);

controller.init(); // starts the event l;isteners
```

Use the full code in another pdf attached.