app.js

```
/*Developers-Details:*/
/*Lidan Danino 207599473
 *Niv Netanel  208540302
 *Carmel Bar   207895103*/

// Define an instance of the StorageActions class to handle storage actions (reading and writing to local storage)
const storage = new StorageActions();

// Select UI elements from the HTML document to be used in the JavaScript code
const form = document.querySelector('#costs-form');
const taskList = document.querySelector('.collection');
const clearBtn = document.querySelector('.clear-costs');
const taskInput = document.querySelector('#cost');
const sumInput = document.querySelector('#sum');
const categoryInput = document.querySelector('#categories');
const reportInput = document.querySelector('#report');
const monthAndYearInput = document.querySelector('#monthAndYear');
const months = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12'];
const ul = document.querySelector('.modal-content .modal-ul');
const modalClose = document.querySelector('.modal-action');
const modalTitle = document.querySelector('.modal-h4');

// Load all the event listeners required for the JavaScript code to function correctly
loadEventListeners();

// Function that sets up all the event listeners needed for the JavaScript code to work
function loadEventListeners() {
    // Event listener that fires when the HTML document has finished loading
    document.addEventListener('DOMContentLoaded', async function () {
        await getCosts();
    });
    // Event listener for the form submission
    form.addEventListener('submit', async function () {
        await addCost();
    });

    // Event listener for the clear costs button
    clearBtn.addEventListener('click', async function () {
        await clearCosts();
    });
    reportInput.addEventListener('click', async function () {
        await getReport();
    });
    modalClose.addEventListener('click', cleanModal);

}

function cleanModal() {
    ul.innerHTML = '';
}
// getting the report of all missions
async function getReport() {
    let year = monthAndYearInput.value.substring(0, 4);
    let month = monthAndYearInput.value.substring(5, 7);
    let show = [];
    await storage.getCostsFromLs().then(items => {
        items.forEach(function (cost) {
            if (month === cost.month && year === cost.year.toString()) {
                show.push(cost);
            }
        });
        modalTitle.innerHTML = `Report of ${month}-${year}:`;
        if (show.length === 0) {
            ul.appendChild(document.createTextNode('You have not entered any expenses on this date'));
        } else {
            show.forEach(function (task) {
                const li = document.createElement('li');
                // Add class
                li.className = 'collection-item';
                // Create text node and append to li
                const span = document.createElement('span');
                span.style.fontWeight = 'bold';
                span.appendChild(document.createTextNode(task.mission));
                li.appendChild(span);
                li.appendChild(document.createTextNode(` ,category: ${task.category}`));
                li.appendChild(document.createTextNode(` ,sum: ${task.sum}`));
                ul.appendChild(li);
            });
        }
    })

}

async function getCosts() {
    await storage.getCostsFromLs().then(items => {
        items.forEach(task => AppendToList(task.mission));
    })
}

function AppendToList(cost) {
    // Create li element
    const li = document.createElement('li');
    // Add class
    li.className = 'collection-item';
    // Create text node and append to li
    li.appendChild(document.createTextNode(cost));
    // Append li to ul
    taskList.appendChild(li);
}

async function addCost(e) {
    if (taskInput.value === '') {
        alert('Add a cost');
    }
    // Create li element
    const li = document.createElement('li');
    // Add class
    li.className = 'collection-item';
    // Create text node and append to li
    li.appendChild(document.createTextNode(taskInput.value));
    // Append li to ul
    taskList.appendChild(li);

    //Add to local storage
    await storage.storeCostInLocalStorage(taskInput.value, sumInput.value, categoryInput.value);

    // Clear input
    taskInput.value = '';
    sumInput.value = '';
}

//Clear Tasks
async function clearCosts() {
    while (taskList.firstChild) {
        taskList.removeChild(taskList.firstChild);
    }
    //Clear ls
    await storage.clearCosts()
}
```