# JavaScript for Front-End – Retake Exam Flower Basket

You have been tasked to create a simple Front-End application, which looks like a **flower basket harvester**. You should create an interface that allows the user to **buy pots**, **plant flowers**, **harvest flowers** and **sell flowers**.

## Task

The task is split into **2 sub-tasks**:

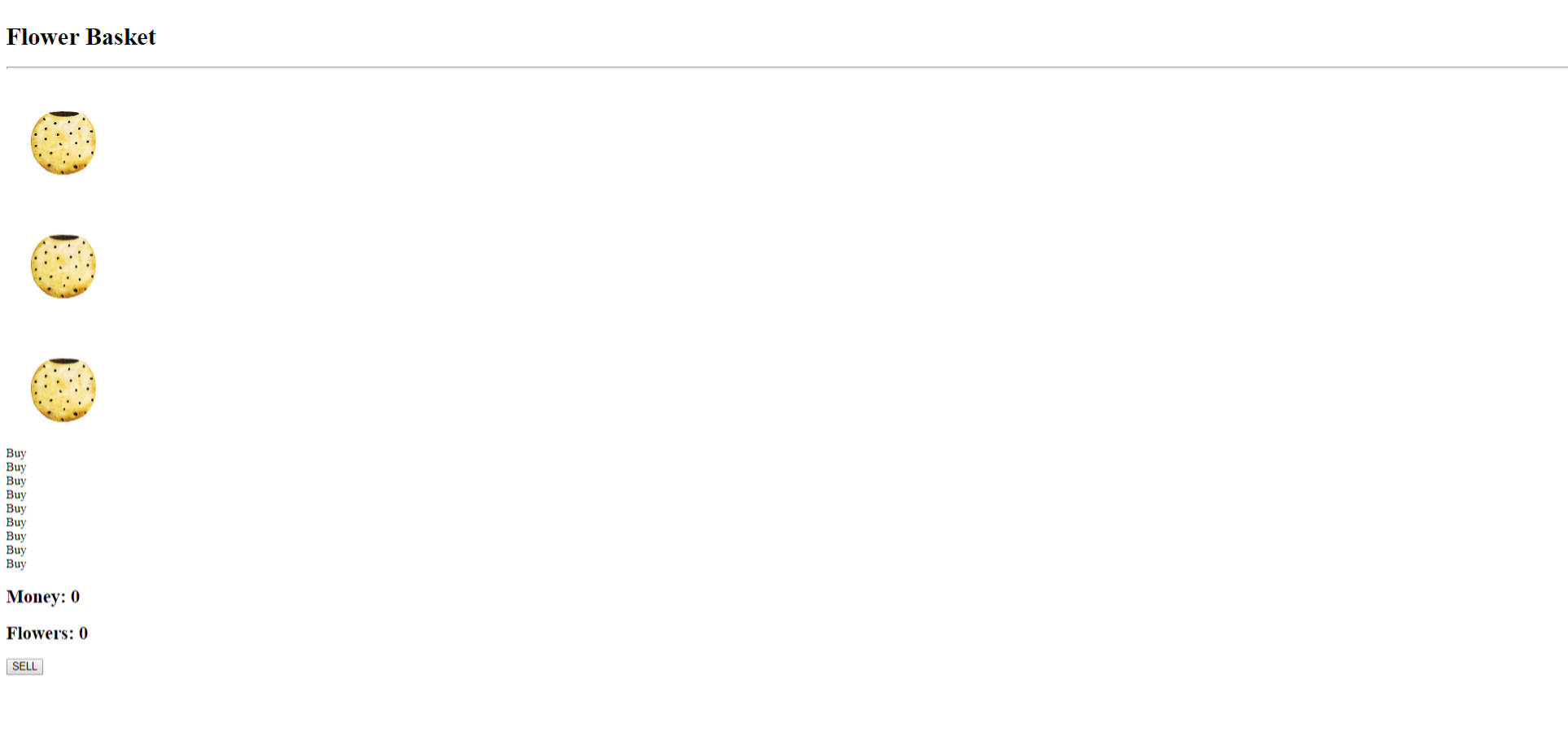
* A simple but fancy **web page design**.
* A simple but dynamic **web page functionality**.

The clients are understanding people, and they understand that you are quite novice as a developer, so they are willing to accept **ONLY** the **web page design** (without **functionality**) or **ONLY** the **web page functionality** (without **design**), if you are **unable** to **implement** one or the other.

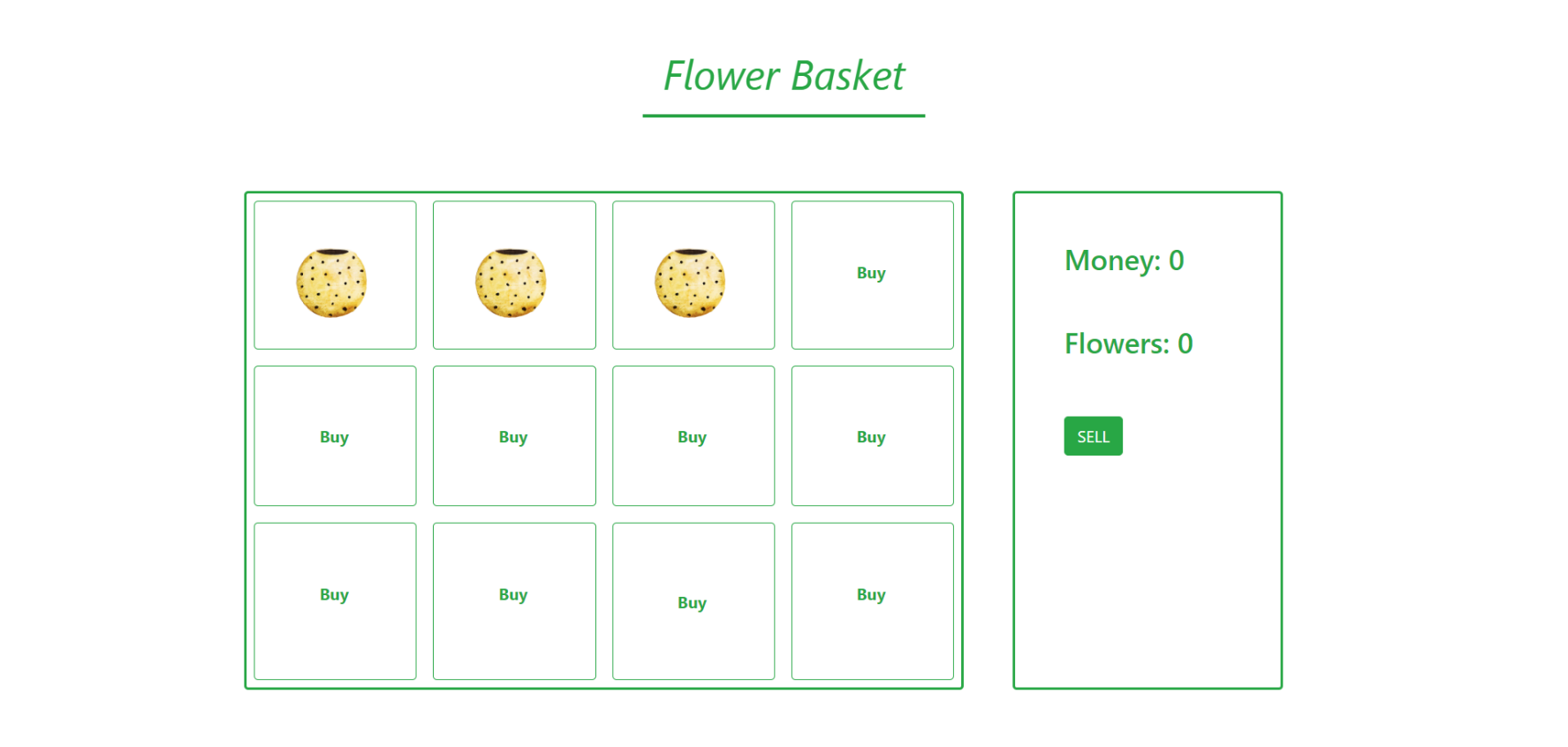
## Overview

### Web Page Design

The web page should look like this **WITHOUT design**:



The web page should look like this **WITH design**:



### Web Page Functionality

The application contains **4 main functionalities**:

* **Buying** pots
* **Planting** flowers
* **Harvesting** flowers
* **Selling** flowers

#### Buying Pots

The application should provide functionality to **buying** **pots** on which they can **plant flowers**. The **pots** **cost money**, which can be earned by **planting** flowers, **harvesting** them and **selling** them afterwards.

Buying **pots** costs **$5.00** money.

Initially the user starts with **3 pots** and **0 money**.

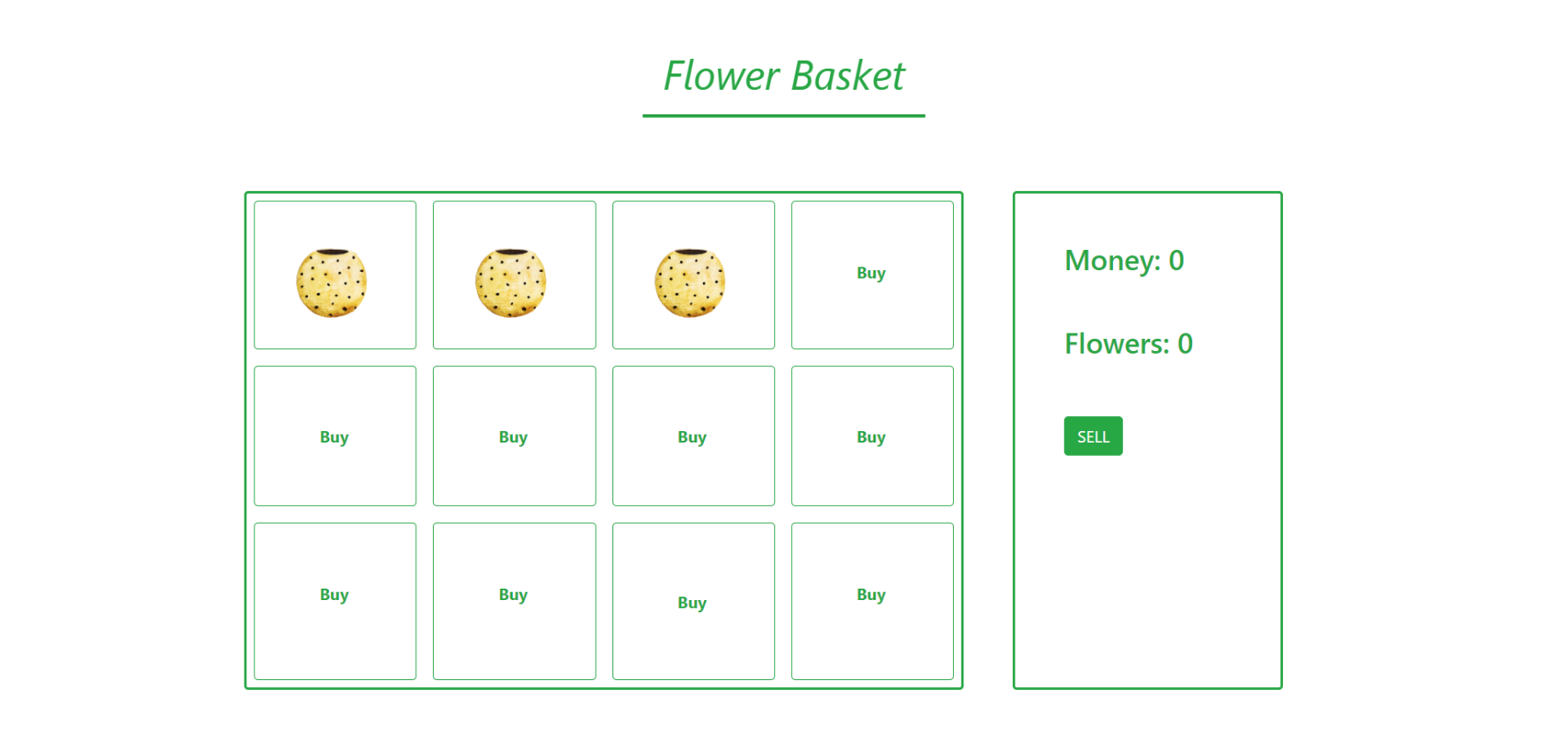
#### Planting Flowers Functionality

The application should provide functionality to **plant flowers** on **available pots**.

An **available pot** is a pot which is **bought** and has **NO flower planted** on it. As we stated above, by default you have **3 such pots** in the beginning.

When you **click** on a pot, it should change the image – **“waiting”** to image – **“in-progress”**.   
**Note**: This simulates a **planted plant behaviour**.

Once the plant has been **planted** a **timer** with **5 seconds** should initiate. After **5 second** the image should be changed **automatically** to the image – **“done”**.   
**Note**: This simulates a **grown plant behaviour**.



#### Harvesting Plants Functionality

The application should provide functionality to **harvest** the **flowers**.

The **“done”** image should have **hover effect**. When one **flower** is ready for **harvest** (the **5 seconds** have **passed**) on **hover** you should show the image – **”harvest”**. If you **hover out**, the image – “**done**” should be **shown again**.

If you **click** the image – “**harvest**”, the **flowers** **count** should be **incremented** with **1** and the image should return to image – **“waiting**”, on that place.

**Note**: This simulates **harvesting plant** behaviour.

#### Selling Plants Functionality

The application should provide functionality to **sell flowers**.

When you **harvest flowers**, you **save** them in the **flowers count**.

**Every flower** costs **$2.50**. When you **click** the [SELL] button you should **sell all flowers**, **returning** the **flower count** to **0** and **increasing** your **money** by **$2.50** for each **flower**.

**Note**: You will be given a **GIF** representing the complete functionality of the **web application**.

## Constraints

The clients are not that much into Front-End technologies, however:

* They heard **Bootstrap** is a good web page styling framework, and they have requested that you **use it**.
* They heard JavaScript makes pages quite dynamic, and **jQuery** is a good library, so you must **use it**.

That being said, you have several constraints to follow, during the implementation of the web application:

* You must use **ONLY** **Bootstrap** for the **web page design**.
* You must use **ONLY** native **JavaScript** & **jQuery** for the **web page functionality**.

## Hints

You can check how to use **setTimeout, mouseover** and **mouseout functions** inJavaScript.