CS6314.002: PROJECT REPORT

PROJECT TITLE: FLORIST

TEAM NUMBER: TEAM-49

NAME FOR THE WEBSITE: INFINITY PETALS

TEAM MEMBERS (NET-IDs, Section No.):

1. SATYA SOMEPALLI (SXS190436, CS6314.002)
2. KRISHNA TEJA YELISETTI (KXY200016, CS6314.002)
3. SAKETH DASAVATHINI (SXD190016, CS6314.002)

PROJECT DESCRIPTION:

Infinity Petals is an online Florist website where a customer can buy a wide variety of flowers available in the portal. Firstly, a user has to register to place an order. There will be a registration link available in the main portal for new customers to register. Without an account, a user can just navigate through different products available on the website but cannot place an order. A user without logging in can filter the products based on category and can search for a product using the search box in the navbar menu. An admin will have a separate webpage to log in.

There are two types of logins available:

- ADMIN LOGIN:
  - Admin can add a new admin or delete an existing admin or admin can just navigate through all the available admins in the system.
  - Admin can add categories or edit categories of available products.

- Admin can add new products or edit available products.
- Admin will have access to all the queries (i.e., Contact-Us form in this project) raised by different customers.
- Admin will have access to all the placed orders and can change the status of the orders.

➔ CUSTOMER LOGIN:
- A customer can filter products based on available categories.
- A customer can search for a product using the search box on the website.
- A customer can add items to the cart.
- A customer can view his/her Profile and edit details.
- A customer can place an order from the cart or directly click on an item/product to place an order.
- A customer can contact the admin by navigating to the Contact-Us form in the navbar.
- A customer can view his/her placed orders.
- A customer can edit the shipping details while placing the order and can also write a special message while placing the order.
- A customer can navigate through different products available.

DATABASE DESIGN:

There will be totally six collections in our database which are as follows:

➔ admin_ids: This collection will store the Admin details. By default, we will add one document with a single admin at first. This single admin can then add multiple admins using the Admin portal. Different fields of admin_ids are :

```
var newAccount = new mongoose.Schema({
   username: {type: String, required: true, index: {unique: true}},
   email: {type: String, required: true, index: {unique: true}},
   password: {type: String, required: true},
   time_of_creation: {type: Date, default: Date.now},
 });
```

Note: A unique _id will be created for each document in the collection.

➔ user_ids: This collection will store all the user details. After successful registration of a user, the details will be stored in the user_ids collection. Different fields of user_ids are :

```
var newAccount = new mongoose.Schema({
   username: {type: String, required: true, index: {unique: true}},
   fName: {type:String, required:true},
   email: {type: String, required: true, index: {unique: true}},
   phone: {type: String, required:true},
   city: {type:String, required:true},
   address:{type:String, required:true},
   password: {type: String, required: true},
   time_of_creation: {type: Date, default: Date.now},
 });
```

Note: A unique _id will be created for each document in the collection.

➔ categories: The categories collection will have information about categories of different products added by the admin. Different fields of categories are :

```
var productCategory =new mongoose.Schema({
   name: String,
   details: String,
   isDeleted: {type:Boolean,default:false}
});
```

Note: A unique _id will be created for each document in the collection.

➔ Products: The products collection will have information about a wide variety of products added by the admin. Different fields of products are :

```
var myProductDetails =new mongoose.Schema({


   product: {type:String, required:true},
   details: String,
   price: {type:Number, required:true},
   images: [String],
   category: String,
   isDeleted: {type:Boolean,default:false}
});
```

Note: A unique _id will be created for each document in the collection.

➔ orders: The orders collection will have information about different orders placed by customers or users. Different fields of orders are :

```
var orders_placed =new mongoose.Schema({
   message: {type: String, required:true},
   amount_nos: [],
   status: {type:String, enum:['Initiated', 'In Progress', 'Delevered', 'Cancel'], default: 'Initiated'
},
   time_of_creation: {type: Date, default: Date.now},
   user: [{type:mongoose.Schema.Types.ObjectId, ref:'User_Ids'}],
   product: [{type:mongoose.Schema.Types.ObjectId, ref:'product'}]
});
```

Note: A unique _id will be created for each document in the collection.

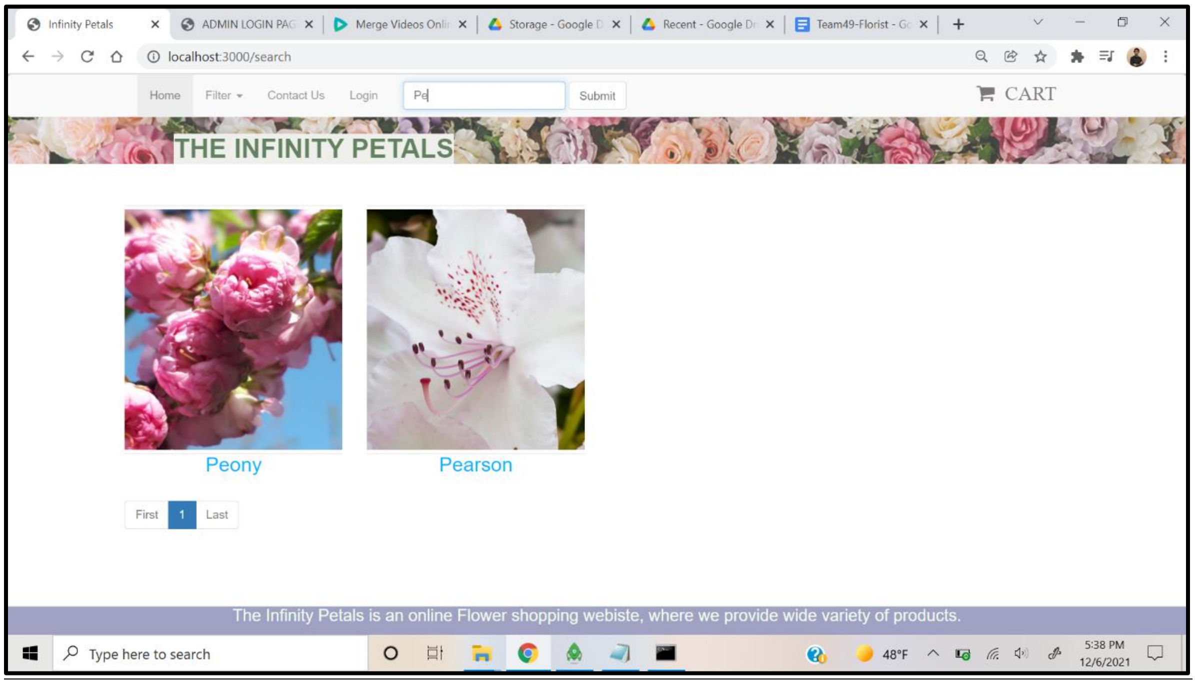
➔ contacts: The contacts collection will have information of all the queries raised by users using the Contact-Us form in the navbar menu. Different fields of contacts are :

```
var usercontactformDetails =new mongoose.Schema({
   name: {type: String, required:true},
   email: {type: String, required:true},
   subject:{type: String, required:true},
   message: {type: String, required:true},
   mobile: {type: String, required:true},
   counter: {type: Number,default:0},
   time_of_creation: {type: Date, default: Date.now},
});
```

Note: A unique _id will be created for each document in the collection.

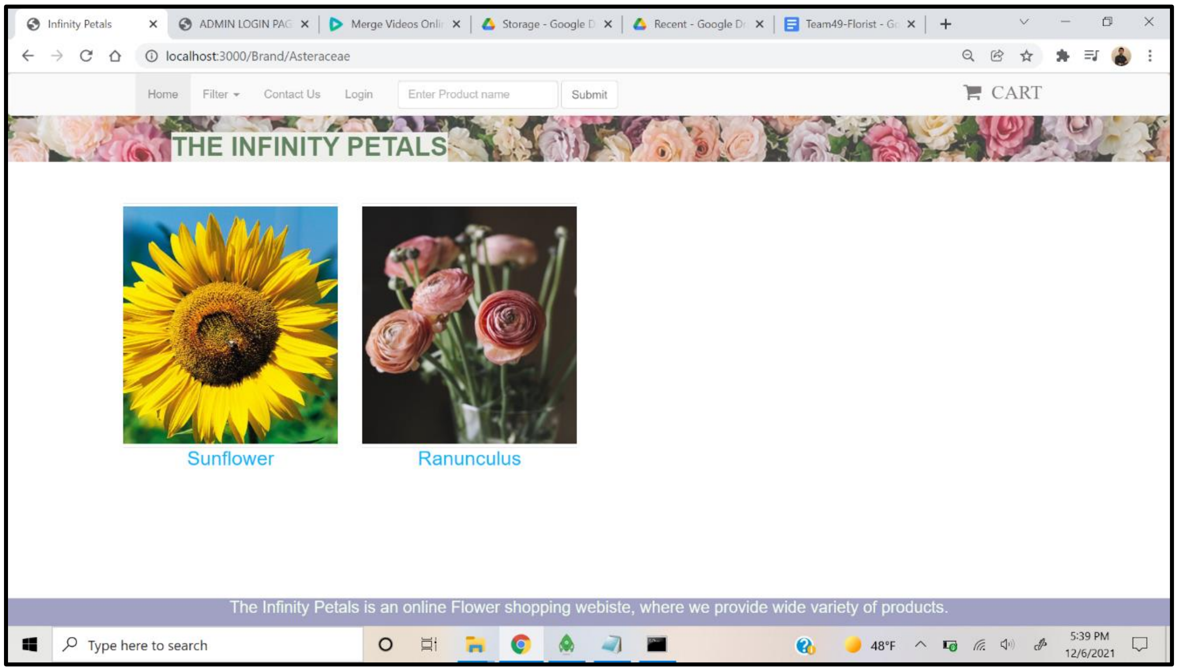
SCREENSHOTS OF MAIN FUNCTIONALITIES:

Following is a screenshot of product search functionality:



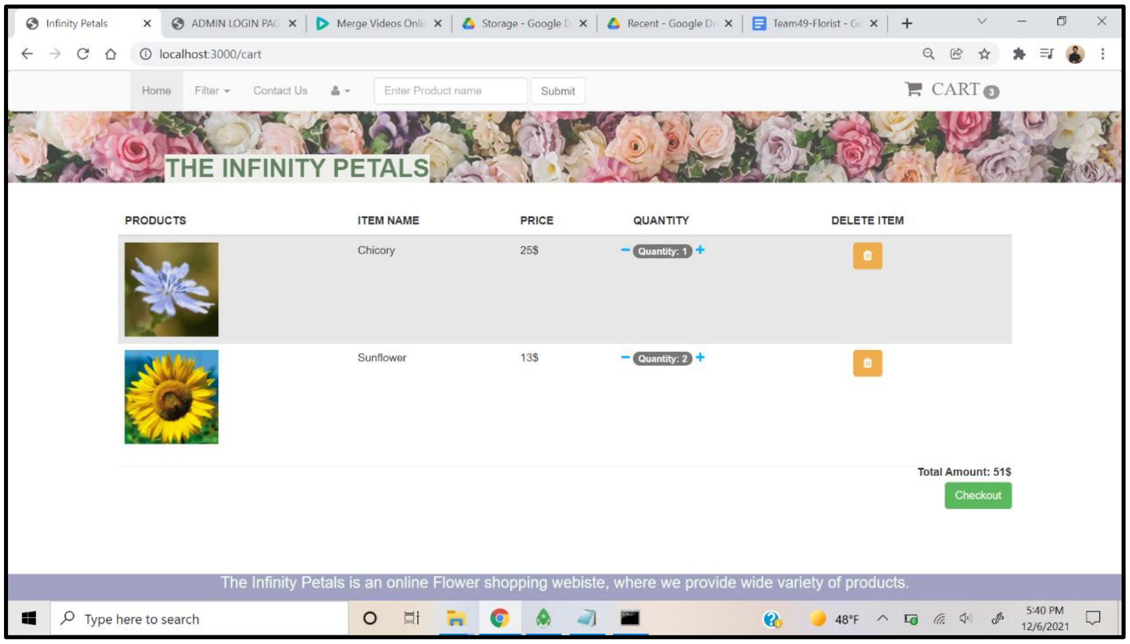
The Infinity Petals is an online Flower shopping webiste, where we provide wide variety of products.

Following is a screenshot of category filter functionality:



The Infinity Petals is an online Flower shopping webiste, where we provide wide variety of products.

Following are screenshots of Cart functionality:

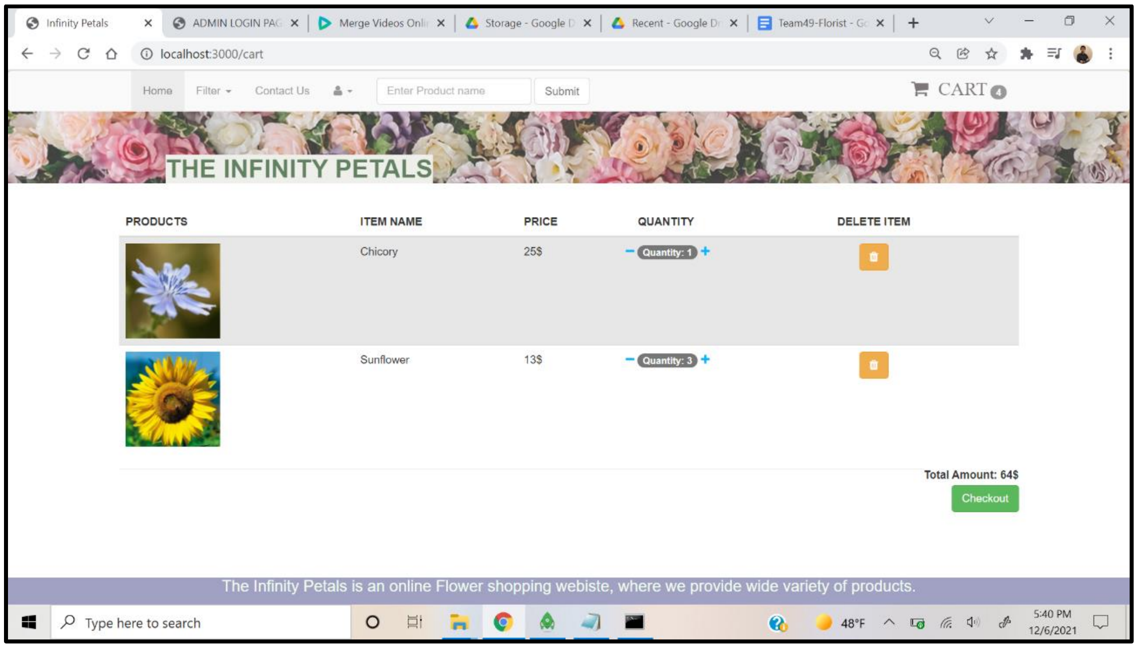


PRODUCTS	ITEM NAME	PRICE	QUANTITY	DELETE ITEM
	Chicory	25$	Quantity: 1	
	Sunflower	13$	Quantity: 2	

Total Amount: 51$

Checkout

The Infinity Petals is an online Flower shopping webiste, where we provide wide variety of products.



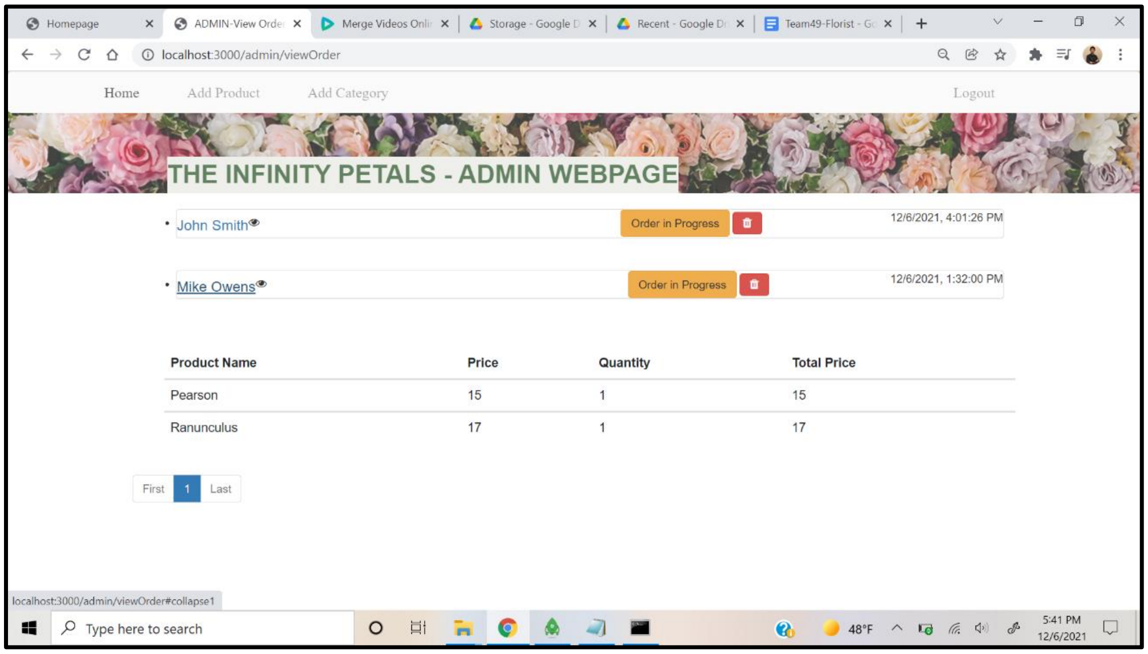
PRODUCTS	ITEM NAME	PRICE	QUANTITY	DELETE ITEM
	Chicory	25$	Quantity: 1	
	Sunflower	13$	Quantity: 3	

Total Amount: 64$

Checkout

The Infinity Petals is an online Flower shopping webiste, where we provide wide variety of products.

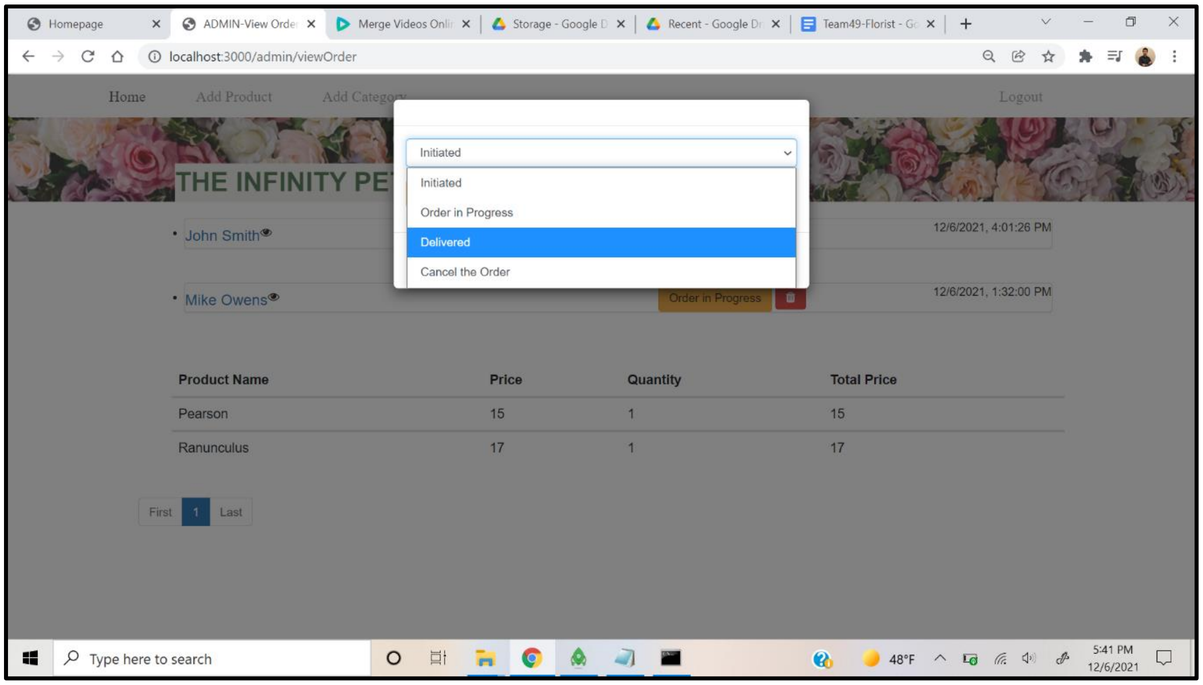
Following is a screenshot of purchase history of users functionality for admin:



- John Smith — Order in Progress — 12/6/2021, 4:01:26 PM
- Mike Owens — Order in Progress — 12/6/2021, 1:32:00 PM

Product Name	Price	Quantity	Total Price
Pearson	15	1	15
Ranunculus	17	1	17

Following is a screenshot of status update of orders functionality for admin:



Product Name	Price	Quantity	Total Price
Pearson	15	1	15
Ranunculus	17	1	17

WORK DIVISION AMONG TEAM MEMBERS:

FILE LOCATIONS	SXS190436	KXY200016	SXD190016
model\product.js		Worked	
model\usercontactform.js			Worked
model\userDBModel.js	Worked		
model\adminDBModel.js	Worked		
model\buy-product.js			Worked
model\cart.js	Worked	Worked	Worked
model\category.js		Worked	
public\stylesheets		Worked	Worked
public\uploads	Worked	Worked	Worked
public\images	Worked	Worked	Worked
public\javascript	Worked		
routes\create-my-category.js		Worked	
routes\create-my-product.js		Worked	
routes\index.js	Worked	Worked	Worked
routes\my-admin-find.js			Worked

routes\my-admin-index.js			Worked
routes\my-admin-signin.js	Worked		
routes\my-category-read.js		Worked	
routes\my-category-update.js		Worked	
routes\my-product-read.js			Worked
routes\my-product-update.js	Worked	Worked	Worked
routes\my-user-search.js	Worked		
routes\usercontactform.js	Worked	Worked	Worked
routes\userLogin.js			Worked
routes\users.js	Worked	Worked	Worked
routes\UserSignup.js	Worked		Worked
routes\view-admin-account.js	Worked		
routes\view-contact.js	Worked	Worked	Worked
routes\view-order.js		Worked	
routes\adminDBModel.js	Worked		
routes\buy-product.js	Worked	Worked	Worked
views\admin\create-my-category.ejs	Worked		

views\admin\create-my-product.ejs		Worked	
views\admin\edit_category.ejs			Worked
views\admin\edit_product.ejs			Worked
views\admin\login-admin.ejs	Worked		
views\admin\my-category-read.ejs	Worked		
views\admin\my-product-read.ejs	Worked		
views\admin\signup-admin.ejs			Worked
views\admin\view-admin-account.ejs		Worked	
views\admin\view-contact.ejs		Worked	
views\admin\view-order.ejs		Worked	
views\admin\adminDashboard.ejs	Worked	Worked	Worked
views\admin\adminHeader.ejs			Worked
views\client\cart.ejs	Worked		
views\client\cart_main.ejs	Worked		
views\client\checkout.ejs	Worked	Worked	Worked
views\client\completedOreder.ejs			Worked

views\client\contact_us.ejs		Worked	
views\client\footer.ejs	Worked	Worked	Worked
views\client\header.ejs		Worked	
views\client\latestProduct.ejs			Worked
views\client\productDetails.ejs			Worked
views\client\userLogin.ejs	Worked		
views\client\userOrder.ejs			Worked
views\client\userProfile.ejs	Worked		
views\client\userSignup.ejs		Worked	
views\client\view-category.ejs	Worked		
views\client\view-product.ejs		Worked	
views\client\buy-product.ejs			Worked
views\error.ejs			Worked
views\index.ejs	Worked	Worked	Worked
app.js	Worked	Worked	Worked

PROJECT- VIDEO- LINK:

https://drive.google.com/file/d/1qFGbVJ4778ZllHmZfMlK9MKo3ggJrcHL/view?usp=sharing

PROJECT-VIDEO-DRIVE-FOLDER:

https://drive.google.com/drive/folders/1yecld6Eot868oLPdErfTK041sva5jcJ2?usp=sharing