

# PlantFy

## Documentation

Team:

## TechSpark

---

12-Aug-2023

- 1.Nasreen (Student id: 1395177)
- 2.Fiza (Student id: 1395060)
- 3.Iqra(Student id: 1395059)
- 4.Anas Raza(Student id:1394096)

---

**Miss Kinza**

# Problem Statement

Whether it is a homely apartment, a sprawling mansion, or an office filled with rows of cubicles - plants tend to brighten and liven up any room. For seasoned gardeners or even beginners, visiting multiple nurseries to find the right plant to suit their requirements can get tedious and consume a lot of time.

An online plant nursery Web application is a solution for this and allows people to shop for plants from the comfort and convenience of their homes. Some plants can survive even with minimal care, while others demand much more attention. The online plant nursery Web application can also provide information to users on proper care for plants. Through such an application, users can view available plants with details such as the plant's cost, level of maintenance required, watering schedule, and so on. Users can also purchase plants and related gardening supplies or tools through the Web application.

# Requirements

## Functional Requirements

For Users:

- Home Page: The home page should typically showcase a selection of plants, featured categories, and so on. It should also provide an introduction to the online plant nursery.

- Account Registration: The registration option shall allow new users (henceforth, called as users) to create secure accounts. It will enable users to register themselves with PlantNest. At the time of registration, users must provide Name, Email ID, Contact Number, and Username, and then, configure their Password.

Appropriate error-checking must be done on the fields of the form to ensure correct data. For example, email id can be checked to see if it is of appropriate format. (Hint: Use client-side validation).

- Login: This option will allow successfully registered users to login to the PlantNest application and access various features of the application through menus or sidebars.

- Settings: Users will be able to manage their accounts by using sub-options such as Create, Update, and Delete Profile.

- Plants Catalog: This option will showcase all varieties of plants one can order online via PlantNest. Catalog will have various categories of plants as follows: 1. Flowering 2. Non-flowering 3. Indoor 4. Outdoor 5. Succulents 6. Medicinal and so on

Plant catalog should be displayed using an image gallery and can have information such as plant name, species, price, discount (if any), description, and so on. Each plant or tool should have Add to Cart and Add to Wishlist feature.

- Accessories: This option should include various accessories necessary for taking care of plants. Details of each accessory such as name, purpose, price, and image, and so on should be included. Accessories can include gardening supplies/tools such as soil, fertilizers, spades, and pesticides too.

# Functional Requirements

- Checkout: This should show the shopping cart contents and total bill based on plant/accessories price and quantity purchased. You do not have to implement payment functionality, just list the cart contents, and compute the bill.
- Cancel Order: Users can cancel an already placed order through this option.
- My Cart: Users can view their cart.
- Submit Reviews: Users can give reviews about plants and accessories they purchased.
- Contact Us: This option should display Email id, address, and contact number of the organization who is developing the application.
- Submit Feedback: Users can submit their feedback regarding the application using a feedback form.

Following are options available for Admin:

- Login: It will allow admin to login to the Web application and add plants, accessories, view orders, and so on.

- Dashboard: In this option, the admin can view the total registered users, total products (plants/accessories), status of orders, and stock.
- Availability: In this option, admin can check whether the particular plant from a specific category is available in stock or not.
- Sold Quantity: This option will allow the admin to keep records of the number of Plants that have been sold. This can be tracked from the cart checkouts.
- View Reviews: This option will allow admin to view reviews submitted by users.
- View Feedback: Admin can view feedback submitted by users regarding the application.

Following are options common to both types of users (Admin and Users):

- Search and Filter: Users/Admin can search for/filter specific plant/accessories based on some criteria.
- Sort: Users/Admin can sort data based on specific criteria.

# Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the application.

The application should be:

- **Safe to use:** The application should not result in any malicious downloads or unnecessary file downloads.

- **Accessible:** The application should have clear and legible fonts, user-interface elements, and navigation elements.

- **User-friendly:** The application should be easy to navigate with clear menus and other elements and easy to understand.

- **Operability:** The application should operate in a reliably efficient manner.

- **Performance:** The application should demonstrate high value of performance through speed and throughput. In simple terms, the application should be fast to load and page redirection should be smooth.

- **Security:** The application should implement adequate security measures such as authentication. For example, only registered users can access certain features.

- **Scalability:** The application architecture and infrastructure should be designed to handle increasing user traffic, data storage, and feature expansions.

- **Capacity:** The application should support large number of users.

- **Availability:** The application should be available 24/7 with minimum downtime.

- **Compatibility:** The application should be compatible with latest browsers.

# Extra Feature

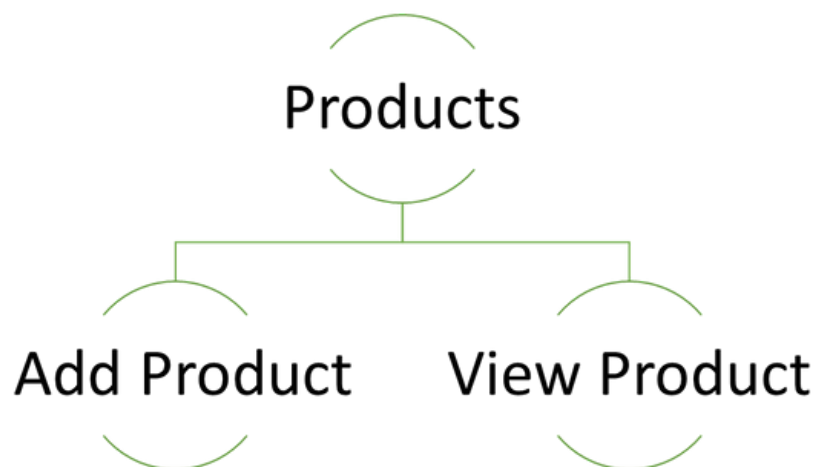
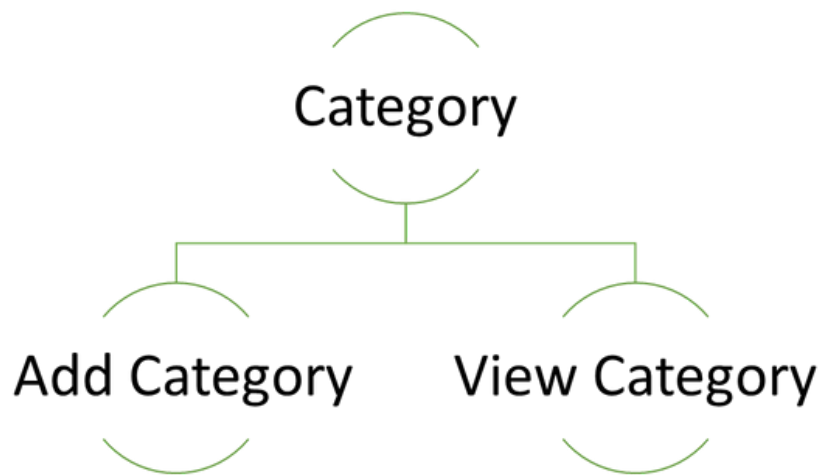
## My Profile:

My Profile section enables authenticated users to modify their profile details, including username, profile picture, and email address. It validates image uploads, interacts with the database using prepared SQL statements, and provides user feedback on successful updates.

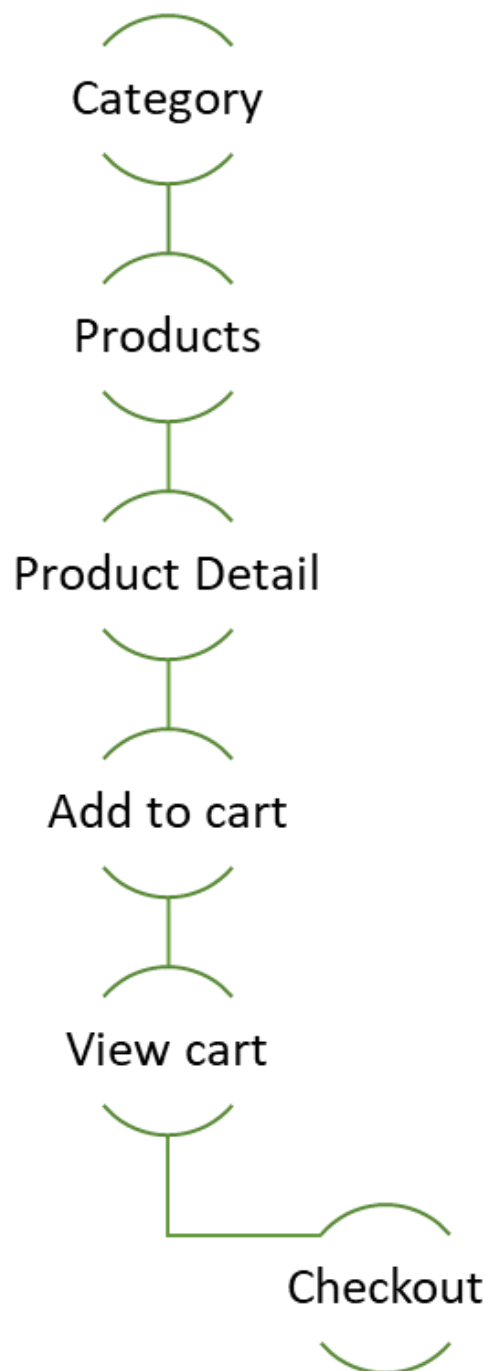
## Functions

At plantNest, we've simplified the process of executing database queries in PHP by introducing custom functions tailored to your needs.. These functions encapsulate the intricacies of query execution, abstracting away complex syntax and minimizing repetitive code.

# Admin Panel Hierarchy

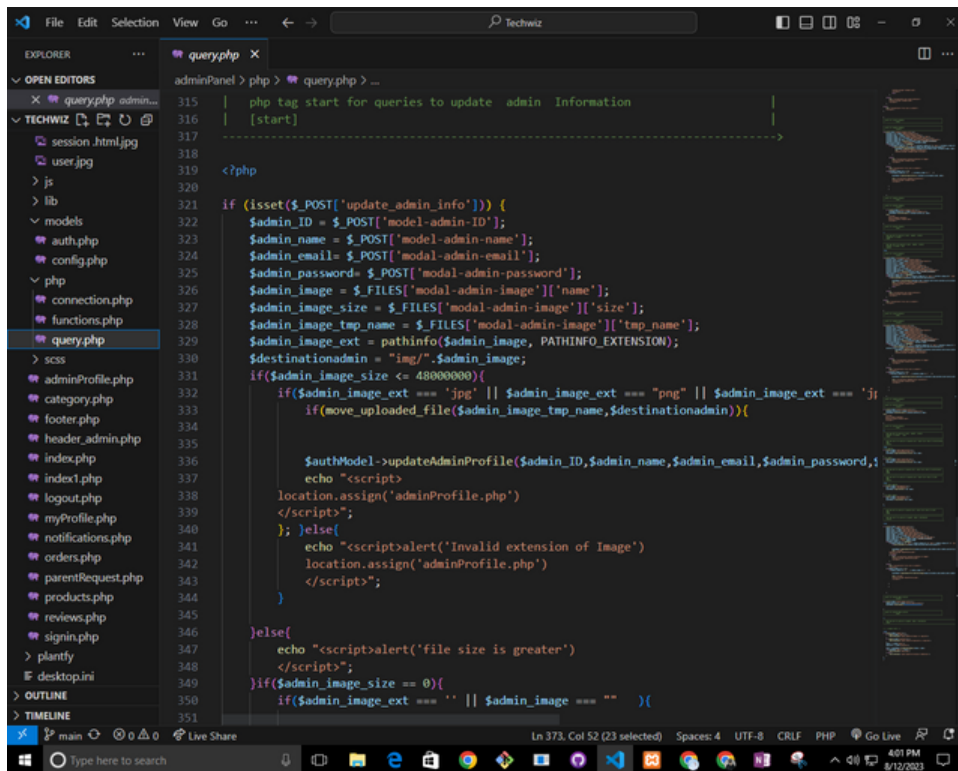


# Website Interface Hierarchy

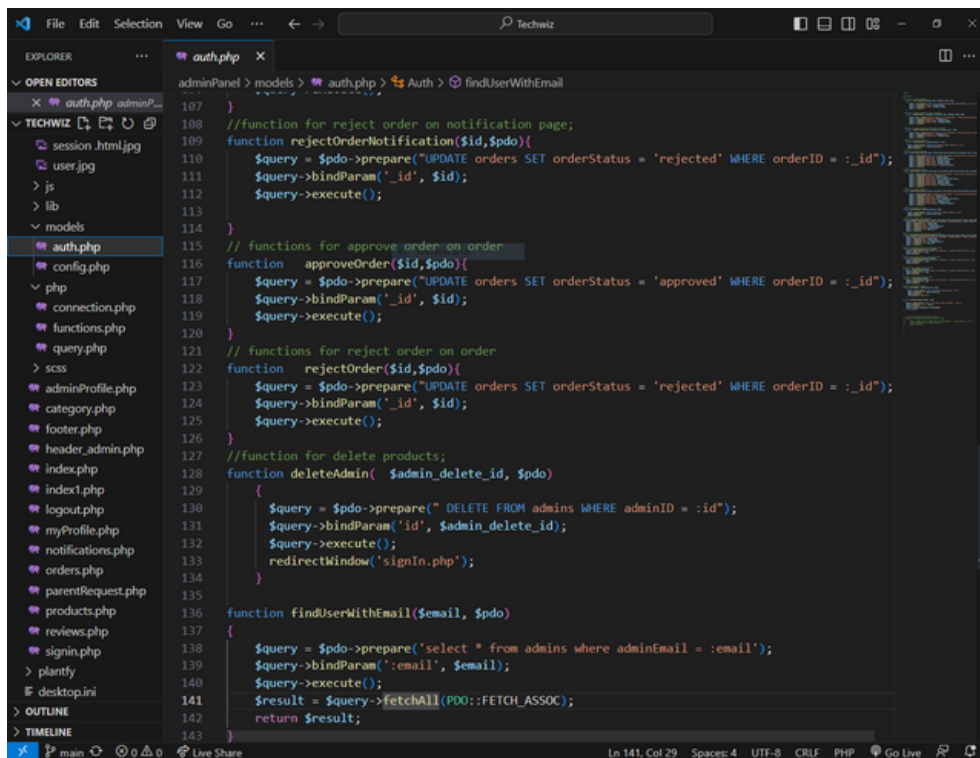




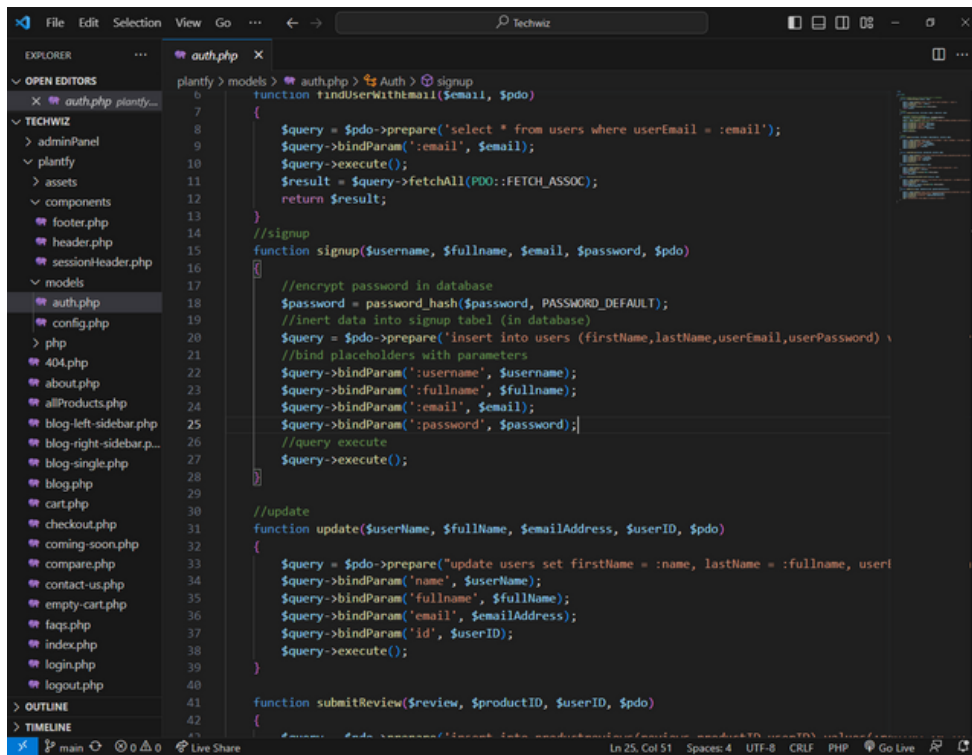
# Screenshots



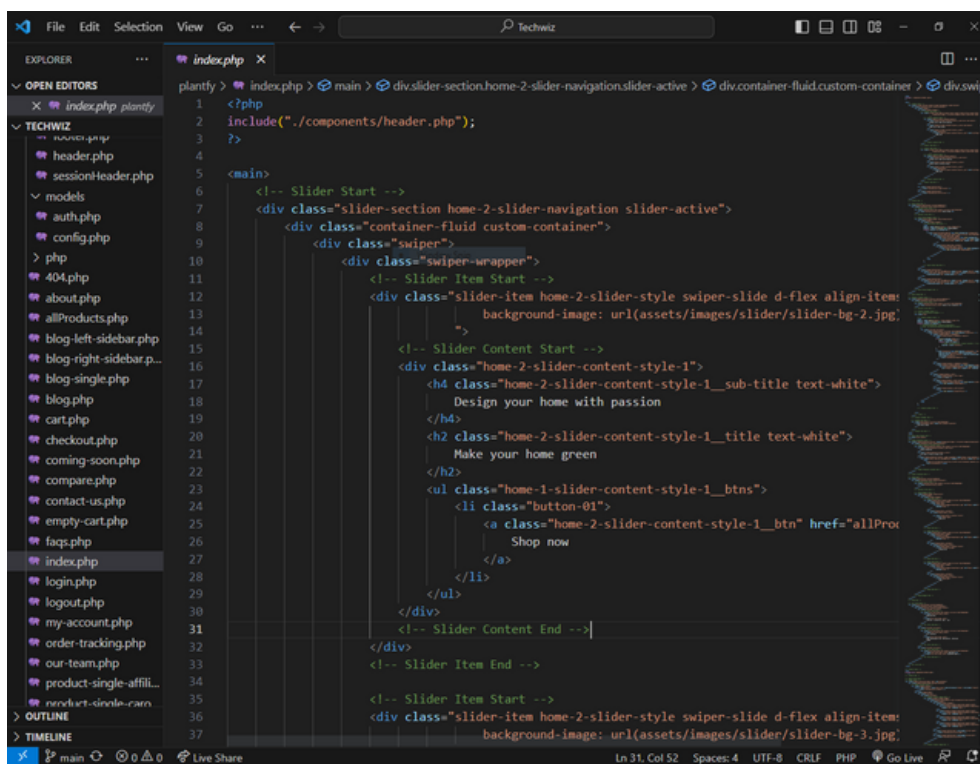
```
adminPanel > php > query.php > --
315 | php tag start for queries to update admin Information
316 | [start]
317 | ----->
318 |
319 | <?php
320 |
321 | if (isset($_POST['update_admin_info'])) {
322 |     $admin_ID = $_POST['model-admin-ID'];
323 |     $admin_name = $_POST['model-admin-name'];
324 |     $admin_email = $_POST['model-admin-email'];
325 |     $admin_password = $_POST['model-admin-password'];
326 |     $admin_image = $_FILES['modal-admin-image']['name'];
327 |     $admin_image_size = $_FILES['modal-admin-image']['size'];
328 |     $admin_image_tmp_name = $_FILES['modal-admin-image']['tmp_name'];
329 |     $admin_image_ext = pathinfo($admin_image, PATHINFO_EXTENSION);
330 |     $destinationadmin = "img/".$admin_image;
331 |     if($admin_image_size <= 48000000){
332 |         if($admin_image_ext === 'jpg' || $admin_image_ext === 'png' || $admin_image_ext === 'j')
333 |             if(move_uploaded_file($admin_image_tmp_name,$destinationadmin)){
334 |
335 |                 $authModel->updateAdminProfile($admin_ID,$admin_name,$admin_email,$admin_password);
336 |                 echo "<script>
337 |                 location.assign('adminProfile.php')
338 |                 </script>";
339 |             }
340 |             ; }else{
341 |                 echo "<script>alert('Invalid extension of Image')
342 |                 location.assign('adminProfile.php')
343 |                 </script>";
344 |             }
345 |         }else{
346 |             echo "<script>alert('file size is greater')
347 |             </script>";
348 |         }
349 |     }if($admin_image_size == 0){
350 |         if($admin_image_ext === '' || $admin_image === '') {
351 |
```



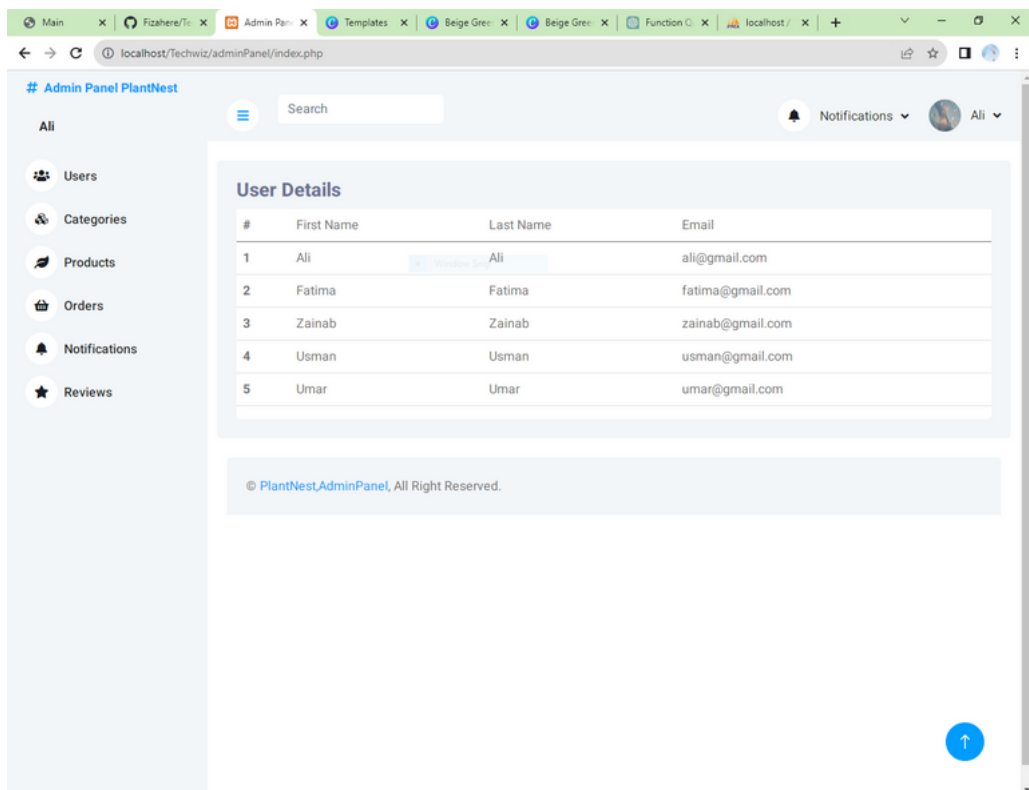
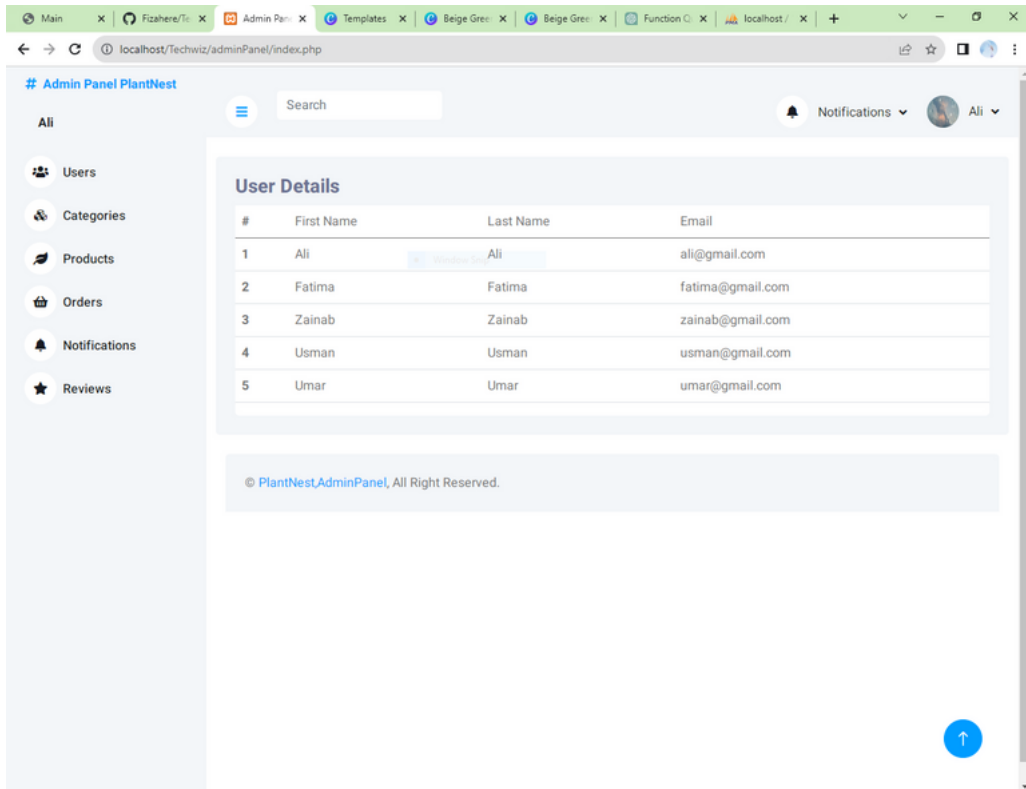
```
adminPanel > models > auth.php > Auth > findUserWithEmail
107 | }
108 | //function for reject order on notification page;
109 | function rejectOrderNotification($id,$pdo){
110 |     $query = $pdo->prepare("UPDATE orders SET orderStatus = 'rejected' WHERE orderID = :_id");
111 |     $query->bindParam('_id', $id);
112 |     $query->execute();
113 | }
114 |
115 | // functions for approve order on order
116 | function approveOrder($id,$pdo){
117 |     $query = $pdo->prepare("UPDATE orders SET orderStatus = 'approved' WHERE orderID = :_id");
118 |     $query->bindParam('_id', $id);
119 |     $query->execute();
120 | }
121 |
122 | // functions for reject order on order
123 | function rejectOrder($id,$pdo){
124 |     $query = $pdo->prepare("UPDATE orders SET orderStatus = 'rejected' WHERE orderID = :_id");
125 |     $query->bindParam('_id', $id);
126 |     $query->execute();
127 | }
128 | //function for delete products;
129 | function deleteAdmin( $admin_delete_id, $pdo)
130 | {
131 |     $query = $pdo->prepare(" DELETE FROM admins WHERE adminID = :id");
132 |     $query->bindParam('id', $admin_delete_id);
133 |     $query->execute();
134 |     redirectWindow('signIn.php');
135 | }
136 |
137 | function findUserWithEmail($email, $pdo)
138 | {
139 |     $query = $pdo->prepare('select * from admins where adminEmail = :email');
140 |     $query->bindParam(':email', $email);
141 |     $query->execute();
142 |     $result = $query->fetchAll(PDO::FETCH_ASSOC);
143 |     return $result;
144 | }
```

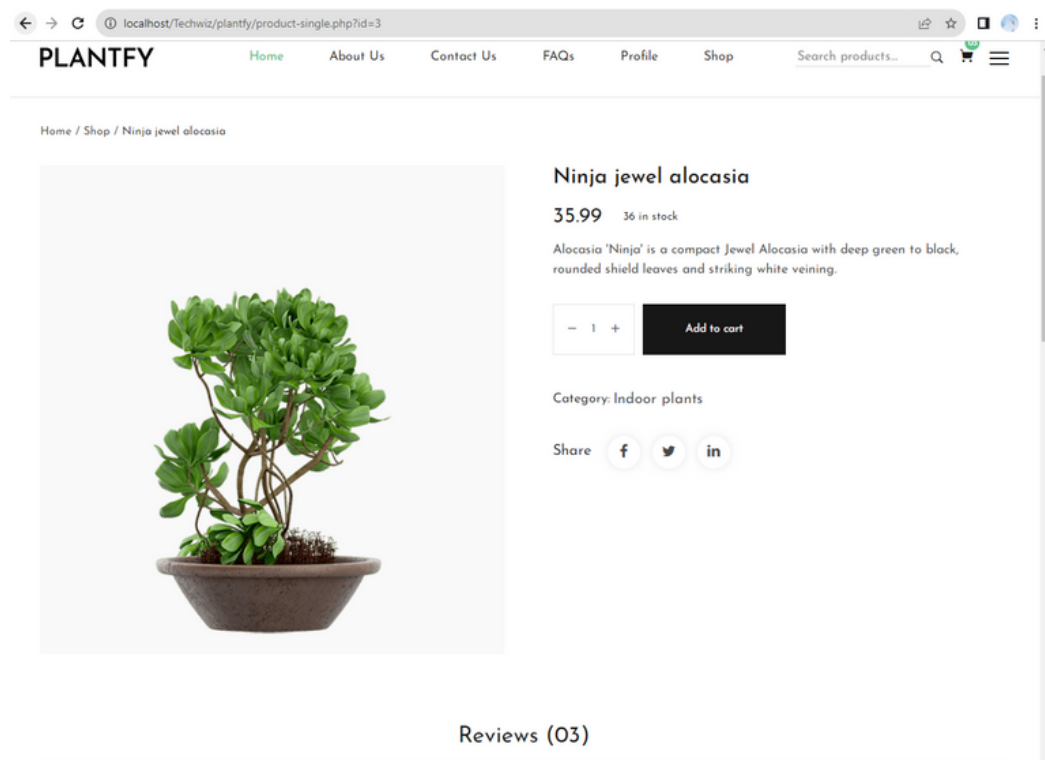
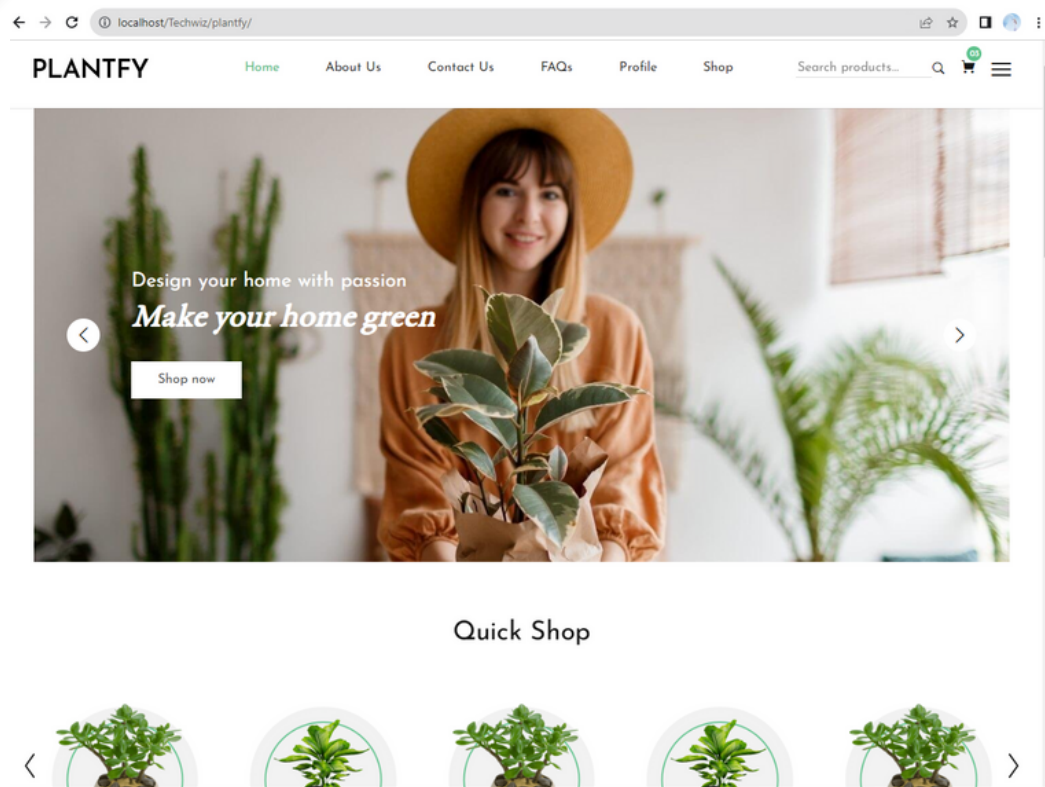


```
plantify > models > auth.php > Auth > signup
1 function findUserWithEmail($email, $pdo)
2 {
3     $query = $pdo->prepare('select * from users where userEmail = :email');
4     $query->bindParam(':email', $email);
5     $query->execute();
6     $result = $query->fetchAll(PDO::FETCH_ASSOC);
7     return $result;
8 }
9 //signup
10 function signup($username, $fullname, $email, $password, $pdo)
11 {
12     //encrypt password in database
13     $password = password_hash($password, PASSWORD_DEFAULT);
14     //insert data into signup tabel (in database)
15     $query = $pdo->prepare('insert into users (firstName,lastName,userEmail,userPassword)');
16     //bind placeholders with parameters
17     $query->bindParam(':username', $username);
18     $query->bindParam(':fullname', $fullname);
19     $query->bindParam(':email', $email);
20     $query->bindParam(':password', $password);
21     //query execute
22     $query->execute();
23 }
24 //update
25 function update($userName, $fullName, $emailAddress, $userID, $pdo)
26 {
27     $query = $pdo->prepare('update users set firstName = :name, lastName = :fullname, user');
28     $query->bindParam('name', $userName);
29     $query->bindParam('fullname', $fullName);
30     $query->bindParam('email', $emailAddress);
31     $query->bindParam('id', $userID);
32     $query->execute();
33 }
34 function submitReview($review, $productID, $userID, $pdo)
35 {
36 }
```



```
plantify > index.php > main > div.slider-section.home-2-slider-navigation.slider-active > div.container-fluid.custom-container > div.wiper
1 <?php
2 include("../components/header.php");
3 ?>
4
5 <main>
6     <!-- Slider Start -->
7     <div class="slider-section home-2-slider-navigation slider-active">
8         <div class="container-fluid custom-container">
9             <div class="swiper">
10                 <div class="swiper-wrapper">
11                     <!-- Slider Item Start -->
12                     <div class="slider-item home-2-slider-style swiper-slide d-flex align-items-center">
13                         background-image: url(assets/images/slider/slider-bg-2.jpg);
14
15                         <!-- Slider Content Start -->
16                         <div class="home-2-slider-content-style-1">
17                             <h4 class="home-2-slider-content-style-1_sub-title text-white">
18                                 Design your home with passion
19                             </h4>
20                             <h2 class="home-2-slider-content-style-1_title text-white">
21                                 Make your home green
22                             </h2>
23                             <ul class="home-1-slider-content-style-1_btns">
24                                 <li class="button-01">
25                                     <a class="home-2-slider-content-style-1_btn" href="allProds">
26                                         Shop now
27                                     </a>
28                                 </li>
29                             </ul>
30                         </div>
31                     <!-- Slider Content End -->
32                 </div>
33             <!-- Slider Item End -->
34
35             <!-- Slider Item Start -->
36             <div class="slider-item home-2-slider-style swiper-slide d-flex align-items-center">
37                 background-image: url(assets/images/slider/slider-bg-3.jpg);
```





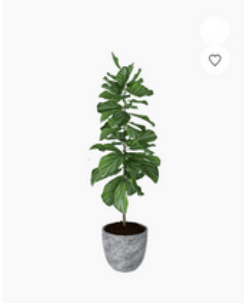
plantnest@mail.com(867)195-6696Los Angeles

LoginHelp Center

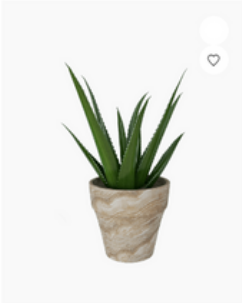
PLANTFYHomeAbout UsContact UsFAQsProfileShop

♥🛒☰

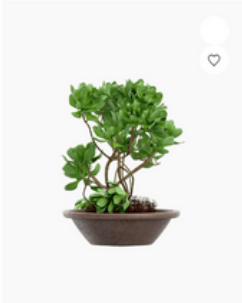
Plants



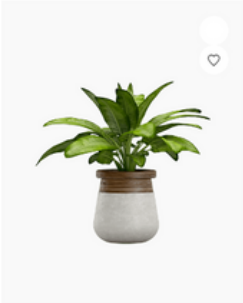
Plant  
Bambino dwarf  
\$ 39.99




Plant  
Golden pothos  
\$ 39.99





Plant  
Ninja jewel alocasia  
\$ 35.99

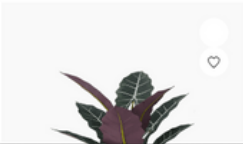


Plant  
Croton  
\$ 25.99









plantnest@mail.com(867)195-6696Los Angeles

LoginHelp Center

PLANTFYHomeAbout UsContact UsFAQsProfileShop

♥🛒☰

My Account

Home > My Account

ProfileLogout

First Nname \*

iman

Last Name \*

Iman

Email Address \*

iman@gmail.com

Save Change