

1. Index.php

This file is the homepage or landing page for the Investus platform, designed to:

- Welcome users with rotating promotional messages encouraging investment.
- Provide easy navigation to login or register as an Admin, Business, or User.
- Alert users when their session has expired due to inactivity via a styled popup message.
- Enhance user experience with interactive menus and timed message transitions.

In short, it acts as a user-friendly entry point that manages session-related messages and directs users to the appropriate login or registration pages.

Key code lines

Session timeout popup and message

```
<?php if (isset($_GET['session_timeout']) && $_GET['session_timeout'] == 1): ?>
```

```
<div class="popup-message" id="session-expired-popup">...</div>
```

```
<script>
```

```
    setTimeout(() => {
```

```
        document.getElementById('session-expired-popup').style.display = 'none';
```

```
    }, 5000);
```

```
</script>
```

```
<?php endif; ?>
```

```
<?php
```

```
session_start();
```

```
if (isset($_GET['message']) && $_GET['message'] === 'session_expired') {
```

```
    echo '<div class="session-expired-message">Your session has expired due to inactivity.  
Please log in again.</div>';
```

```
}
```

```
?>
```

rotating hero message:

```
const messages = [...];
```

```
setInterval(rotateMessages, 7000);
```

Navigation

```
<nav>
```

```
    <div class="nav-item" onclick="toggleSubmenu(event, 'login-menu')">...</div>
```

```
<div class="nav-item" onclick="toggleSubmenu(event, 'register-menu')">...</div>

</nav>
```

2. config.php

This file initializes the database connection using MySQL and ensures any connection error is caught and displayed. It is included in other PHP files like verify.php, login.php, and more.

Key code:

Defines DB

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "";
```

```
$dbname = "investus";
```

creates new connection

```
$conn = new mysqli(...);
```

Checks if connection failed

```
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
```

```
}
```

3. Login.php, Alogin.php, Blogin.php

These files are the login system for the Investus platform. It authenticates users by checking submitted credentials against stored data and starts a session if successful. It also provides a visually appealing login form for users to interact with.

Key code lines

Database connection

```
include 'config.php';
```

```
session_start();
```

Gets submitted info and check if username exists

```
$username = $_POST['username'];
```

```
$password = $_POST['password'];
```

```
$sql = "SELECT * FROM users WHERE username='$username'";
```

```
$result = $conn->query($sql);
```

password verification

```
if (password_verify($password, $row['password'])) {
```

```
    $_SESSION['username'] = $username;
```

```
    header("Location: Uindex.php");
```

```
} else {
```

```
    echo "Invalid password.";
```

```
}
```

4. register.php, Bregister.php

These files power the **Investus user registration system**. It validates input, prevents duplicate emails, securely stores passwords, and sends a confirmation email with a 5-digit code to verify the user's identity before account activation or in the case of business registry, it will send a application received email which fate will be upon the admins.

Key code

Connects to the database and includes PHPMailer for sending emails.

```
include 'config.php';
```

```
require 'vendor/autoload.php';
```

```
use PHPMailer\PHPMailer\PHPMailer;
```

checks for duplicate emails

```
$check = $conn->prepare("SELECT id FROM users WHERE email = ?");
```

Checks password strength

```
if (!preg_match('/[A-Z]/', $password)) { ... }
```

5. passwordrec.php and Bpasswordrec.php

This file handles a change of password for users and business users if a password is forgotten

Key code:

Check if the username and email match in the database

```
$sql = "SELECT * FROM users WHERE username='$username' AND email='$email'";  
$result = $conn->query($sql);
```

User exists, show the password change form

```
$_SESSION['username'] = $username;  
$_SESSION['email'] = $email;  
echo '<div class="login-container">
```

Verify the new password and confirmation match

```
if ($new_password === $confirm_password) {  
    $username = $_SESSION['username'];  
    $hashed_password = password_hash($new_password, PASSWORD_DEFAULT);
```

Update the password in the database

```
$stmt = $conn->prepare("UPDATE users SET password=? WHERE username=?");  
$stmt->bind_param("ss", $hashed_password, $username);
```

6. logout.php

this file acts as a session terminator and redirects back to the index.php

key code:

```
<?php  
session_start();  
session_unset();  
session_destroy();  
header("Location: index.php");  
exit();  
?>
```

7. verify.php

This file confirms a newly registered user's email address by verifying a 5-digit code they received via email. Once verified, the code is cleared from the database, and the user is redirected to login.

key code:

Fetches the stored verification code

```
$stmt = $conn->prepare("SELECT verification_code FROM users WHERE email = ?");
```

Compares user entered code with stored code

```
if ($stored_code == $input_code) {
```

8. Uindex.php, Bindex and Aindex

These files are the protected homepage for logged-in Investus users. It welcomes them, offers navigation to key areas, and ensures session security through automatic timeout.

Navigation Menu includes:

- Browse
- Investors
- Create Content
- Creators Hub
- Logout

Key code:

Session timeout

```
$timeout_duration = 600;
```

(From this point, Every page will contain this code(excluding handlers))

9. browse.php, Bbrowse.php, Abrowse.php

This page is designed to serve as the main feed or home page for a social media platform called Investus. It brings together the core features you'd expect in a social networking or content-sharing site, tailored for both regular users and business users.

Key code:

Fetch posts

```
function fetchPosts($username) {  
  
    ...  
  
}
```

Fetch stars of the day

```
$stars_stmt = $conn->prepare("  
  
    SELECT username, profile_picture FROM users WHERE id IN (SELECT user_id FROM stars)  
  
    UNION  
  
    SELECT username, profile_picture FROM business_users WHERE id IN (SELECT business_id  
FROM stars)  
");
```

Page layout

```
<div class="sidebar">...</div>  
  
<div class="main-content">...</div>  
  
<div class="sidebar">...</div>
```

Display each post

```
<?php foreach ($posts as $post): ?>  
  
<div class="post">  
  
    ...  
  
</div>  
  
<?php endforeach; ?>
```

Search for users

```
document.getElementById('search-input').addEventListener('input', function () {  
  
    ...  
  
});
```

10. search_users.php

This file powers the live search bar across the Investus platform. It queries both user types for matching usernames and returns clickable suggestions in real-time to help users quickly find and access other profiles.

Key code:

Retrives search query

```
$query = isset($_GET['q']) ? $conn->real_escape_string($_GET['q']) : '';
```

Search the user and business_users tables

```
$stmt = $conn->prepare("SELECT username FROM users WHERE username LIKE ? LIMIT 5");
```

Builds suggestions

```
echo '<div class="suggestion-item"><a href="view_profile.php?username=' .  
urlencode($suggestion) . "'>' . htmlspecialchars($suggestion) . '</a></div>';
```

11. create_content.php and Bcreate_content.php

This page enables authenticated users to create multimedia content (text, image, or video posts), ensuring secure sessions, clean UI, and a user-friendly publishing experience.

Key code:

inserts new post into table

```
INSERT INTO contents (username, text, file_url, file_type)
```

handles content submission

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
```

Check if file is an image or video

```
$check = getimagesize($_FILES["file"]["tmp_name"]);
```

```
if ($check !== false || in_array($imageFileType, ['mp4', 'avi', 'mov'])) {
```

```
    $uploadOk = 1;
```

```
} else {
```

```
    $uploadOk = 0;
```

```
}
```

12. profile.php and Bprofile.php

This file displays a logged-in user's profile on Investus. It allows them to update their profile picture, view their follower count, and manage their posts. The page also supports light/dark mode switching and session timeout handling.

Key code:

Profile picture upload

```
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_FILES['profile_picture'])) {  
    $target_dir = "uploads/";  
    $target_file = $target_dir . basename($_FILES["profile_picture"]["name"]);  
    move_uploaded_file($_FILES["profile_picture"]["tmp_name"], $target_file);  
    $stmt = $conn->prepare("UPDATE users SET profile_picture = ? WHERE username = ?");  
    $stmt->bind_param("ss", $target_file, $username);
```

Fetch profile picture

```
$stmt = $conn->prepare("SELECT profile_picture FROM users WHERE username = ?");
```

Fetch user posts

```
$stmt = $conn->prepare("SELECT id, text, file_url, file_type FROM contents WHERE username = ?");
```

Fetch follower count

```
$stmt = $conn->prepare("SELECT COUNT(*) FROM followers WHERE following = ?");
```

13. view_profile.php

this page is separate from the profile.php as it's what the search_users will redirect to and contains the function to follow users and view their content.

Key code:

follow/unfollow logic

```
function toggleFollow() {  
    let username = "<?php echo $viewed_user; ?>";  
    let formData = new FormData();  
    formData.append("following", username);  
  
    fetch("follow.php", {
```



```

        method: "POST",
        body: formData
    })
    .then(response => response.json())
    .then(data => {
        if (data.status === "followed") {
            document.getElementById("followBtn").innerText = "Unfollow";
        }
    })
}

```

14. delete_post.php and Bdelete_posts

this file is a handler for the deletion of posts by users and business users

key code:

```

// Delete the post
$stmt = $conn->prepare("DELETE FROM contents WHERE id = ? AND username = ?");
$stmt->bind_param("is", $post_id, $username);

```

15. follow.php

handler for the following function

key code:

Check if user exists in either 'users' or 'business_users'

```

$user_exists = false;
$tables = ['users', 'business_users'];
foreach ($tables as $table) {
    $stmt = $conn->prepare("SELECT username FROM $table WHERE username = ?");
    $stmt->bind_param("s", $following);
}

```

Check if already following

```

$stmt = $conn->prepare("SELECT id FROM followers WHERE follower = ? AND following = ?");
$stmt->bind_param("ss", $follower, $following);

```

Unfollow the user

```
$stmt->close();  
  
$stmt = $conn->prepare("DELETE FROM followers WHERE follower = ? AND following = ?");  
  
$stmt->bind_param("ss", $follower, $following);
```

Follow the user

```
$stmt->close();  
  
$stmt = $conn->prepare("INSERT INTO followers (follower, following) VALUES (?, ?)");  
  
$stmt->bind_param("ss", $follower, $following);
```

16. library.php, Blibrary.php, Alibrary.php

This page is an Investment Library where users can browse various types of investment-related content such as articles, videos, and images. The content is fetched from a database, and it is displayed in a user-friendly layout. On the admin side of this page, they are allowed to upload content for users and businesses to see.

Key code:

Fetching content

```
$articles = $conn->query("SELECT * FROM library_articles ORDER BY created_at DESC");  
$videos = $conn->query("SELECT * FROM library_videos ORDER BY created_at DESC");  
$images = $conn->query("SELECT * FROM library_images ORDER BY created_at DESC");
```

Displaying content

```
<div class="article">  
  
    <h3><?php echo htmlspecialchars($article['title']); ?></h3>  
  
    <p><?php echo nl2br(htmlspecialchars($article['content'])); ?></p>  
  
</div>  
  
<iframe src="<?php echo htmlspecialchars($video['video_url']) . '?autoplay=0'; ?>"  
frameborder="0" allowfullscreen></iframe>  
  
')">
```

17. creators_hub.php and Bcreators_hub.php

This page allows registered users to share their ideas by posting text along with an image or video. It also provides an option for users to toggle between light and dark modes. The posts are displayed in a feed, along with the username of the poster and their profile picture. The page features a form for uploading new ideas, and all posts are stored in a database. On the business side, this page will not allow them to post, but they get a special I'm interested button that will send an email of interest to the receiver and sender.

Key code:

Fetch all posts along with profile pictures

```
$posts = $conn->query("
```

```
    SELECT chp.*, u.profile_picture
```

```
    FROM creators_hub_posts chp
```

```
    JOIN users u ON chp.username = u.username
```

```
// Insert into database
```

```
    $stmt = $conn->prepare("INSERT INTO creators_hub_posts (username, idea_text, file_url, file_type) VALUES (?, ?, ?, ?)");
```

```
    $stmt->bind_param("ssss", $username, $idea_text, $file_url, $file_type);
```

Handle uploads

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
```

```
    $idea_text = isset($_POST['idea_text']) ? trim($_POST['idea_text']) : '';
```

```
    $file_url = '';
```

```
    $file_type = '';
```

```
    if (!empty($_FILES['file']['name'])) {
```

```
        $fileName = $_FILES['file']['name'];
```

Handle "I'm Interested" action

```
if (isset($_POST['interested']) && isset($_POST['id'])) {
```

```
    $post_id = $_POST['id'];
```

```
    $business_user = $_SESSION['username'];
```

Fetch the business user's email from the database

```
$stmt = $conn->prepare("SELECT email FROM business_users WHERE username = ?");
```

```
$stmt->bind_param("s", $username);
```

18. interest.php

handler for the im interested function in the above.

Key code:

Retrieve the post ID from POST data

```
$post_id = $_POST['post_id'];
```

Fetch the post details

```
$stmt = $conn->prepare("SELECT chp.username AS creator_username, chp.idea_text, u.email  
AS creator_email FROM creators_hub_posts chp JOIN users u ON chp.username = u.username  
WHERE chp.id = ?");
```

```
$stmt->bind_param("i", $post_id);
```

Fetch the business user's email from the business_users table

```
$business_user_email = "";
```

```
$business_user_query = $conn->prepare("SELECT email FROM business_users WHERE  
username = ?");
```

```
$business_user_query->bind_param("s", $logged_in_user);
```

19. market.php

The file fetches live stock market data via an external API, processes it, and displays it in a graphical format

Key code:

Fetching Stock Data

```
function fetchStockData($params) {
```

```
    $url = "https://www.searchapi.io/api/v1/search?" . http_build_query($params);
```

```
    $curl = curl_init();
```

```
    curl_setopt_array($curl, [
```

```

    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_TIMEOUT => 60,
    CURLOPT_HTTPHEADER => ["Accept: application/json"]
  ]);
  $response = curl_exec($curl);
  curl_close($curl);
  return json_decode($response, true);
}

```

Rendering Stock Data in a Chart

```

const ctx = document.getElementById('stockChart').getContext('2d');
new Chart(ctx, {
  type: 'line',
  data: {
    labels: labels,
    datasets: [
      { label: 'Nikkei 225', data: nikkeiPrices, borderColor: 'gold', borderWidth: 2, fill: false },
      { label: 'Tesla', data: teslaPrices, borderColor: 'red', borderWidth: 2, fill: false },
    ]
  },
  options: { responsive: true, maintainAspectRatio: false }
});

```

20. More.php

This file is a store for additional functions that may affect the website graphics

Key code:

```

<script>

  document.addEventListener('DOMContentLoaded', () => {
    if (localStorage.getItem('dark-mode') === 'enabled') {
      document.body.classList.add('dark-mode');
    }
  });

```

```

    }

    window.toggleDarkMode = function() {
        document.body.classList.toggle('dark-mode');
        if (document.body.classList.contains('dark-mode')) {
            localStorage.setItem('dark-mode', 'enabled');
        } else {
            localStorage.removeItem('dark-mode');
        }
    };
};
});
</script>

```

21. Investors.php and Ainvestors.php

This PHP file checks for session inactivity and redirects to the login page if the session has expired or if the user is not logged in. It displays a list of investor opportunities across different industries, with each category linking to a separate page.

Key code:

Layout

```

<div class="key-industries">

    <h2 style="text-align:center;">Key Industries</h2>

    <div class="industries-grid">

        <a href="industries/aerospace_defense.php" class="industry">Aerospace &
        Defense</a>

        ...

        <a href="industries/tech.php" class="industry">Technology</a>

    </div>

</div>

```

22. Aeospace_defence.php to tech.php and Naerospace_defence.php to Ntech.php

This file collection handles the display of investment opportunities on a secure page where only authenticated users can access the content. It checks for session timeouts and ensures the

user is logged in before proceeding. It fetches and displays investment details from a database, showing relevant information such as descriptions, investment type, and images. On the business side of the website, they are confined to the N versions, so they do not gain access to the invest button.

Key code:

Data Fetching

```
$sql = "SELECT * FROM admin_investments WHERE company_type = 'Aerospace & Defense'
ORDER BY id DESC";
```

```
$result = $conn->query($sql);
```

```
$conn->close();
```

Predefined structure

```
<h1>Aerospace & Defense Investment Opportunities</h1>
```

```
<?php if ($result->num_rows > 0): ?>
```

```
    <?php while ($investment = $result->fetch_assoc()): ?>
```

```
        <div class="container">
```

```
            
```

```
            <div class="investment-info">
```

```
                <h2><?php echo htmlspecialchars($investment['header_text']); ?></h2>
```

```
                <p><?php echo nl2br(htmlspecialchars($investment['description'])); ?></p>
```

```
                <p><strong>Investment Type:</strong> <?php echo
ucfirst(htmlspecialchars($investment['investment_type'])); ?></p>
```

```
            </div>
```

```
            <a href="investment-process.php?investment_type=<?php echo
urlencode($investment['investment_type']); ?>" class="invest-button">Invest Now</a>
```

```
        </div>
```

```
    <?php endwhile; ?>
```

```
<?php else: ?>
```

```
    <p style="text-align:center;">No Aerospace & Defense investments available. Please check
back later.</p>
```

```
<?php endif; ?>
```

23. Investment-process.php

This file is a redirector/ handler which redirects based on the given investment type stated in each container.

Key code:

Retrieve the investment type from the URL parameter

```
$investment_type = isset($_GET['investment_type']) ? $_GET['investment_type'] : '';
```

Validate the investment type

```
$valid_investment_types = [  
    'stocks' => 'forms/stock.php',  
    'bonds' => 'forms/bond.php',  
    'real_estate' => 'forms/real_estate.php',  
    'mutual' => 'forms/mutual.php',  
    'etf' => 'forms/etf.php',  
    'commodities' => 'forms/commodities.php',  
    'crypto' => 'forms/crypto.php'
```

Check if the investment type is valid and redirect

```
if (array_key_exists(strtolower($investment_type), $valid_investment_types)) {
```

Redirect to the respective form

```
    header("Location: " . $valid_investment_types[strtolower($investment_type)]);  
    exit();
```

23. bond.php, commodities.php, crypto.php, etf.php, mutual.php, real_estate.php, stock.php

This collection PHP scripts handles form submission for the various forms and performs multiple actions, including session management, form validation, file creation, database insertion, and email notifications.

Key code:

Form Submission Handling

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $investor_name = $_POST['investor_name'] ?? "";  
    // Collect all form data...  
    if (empty($investor_name) || empty($investor_email) || empty($investor_phone) /* other  
checks */) {  
        die("All required fields must be filled.");  
    }  
    // Create .txt file...  
}
```

File Creation

```
$details = "Stock Investment Request\n\n";  
$details .= "Investor Name: $investor_name\n";  
// Add other details...  
$filename = "Stock_Investment_Request_" . time() . ".txt";  
$filepath = $dir . $filename;  
file_put_contents($filepath, $details);
```

Database Insert

```
$stmt = $conn->prepare("INSERT INTO stock_requests (company_name, file_path, created_at)  
VALUES (?, ?, NOW())");  
$stmt->bind_param("ss", $company_name, $filepath);  
$stmt->execute();
```

Email Notifications

```
$mail = new PHPMailer(true);  
Prepare and send email to business user...  
$mail->addAddress($business_email, $company_name);  
$mail->Subject = 'New Stock Investment Request';  
// Send the email to the business user...
```

```
$mail->send();
```

Send email to the investor

```
$mail->clearAddresses();
```

```
$mail->addAddress($investor_email, $investor_name);
```

```
$mail->Subject = 'Stock Investment Request Submitted';
```

```
$mail->send();
```

24. Submit_stock.php

This file is a handler for form submission.

Key code:

```
$stmt = $conn->prepare("INSERT INTO investments (business_user_id, company_name, industry, investment_type, description, logo_url) VALUES (?, ?, ?, ?, ?, ?)");
```

```
$stmt->bind_param("isssss", $business_user_id, $company_name, $industry, $investment_type, $description, $logo_url);
```

25. Get_ratings.php

This file is a handler for the display of specific ratings under each posts.

Key code:

Get average rating for the post

```
$stmt = $conn->prepare("SELECT COALESCE(AVG(rating), 0) as avg_rating FROM ratings WHERE post_id = ?");
```

```
$stmt->bind_param("i", $post_id);
```

```
$stmt->execute();
```

```
$stmt->bind_result($avg_rating);
```

```
$stmt->fetch();
```

```
$stmt->close();
```

```
$user_rating = null;
```

```
$has_rated = false;
```

Business Exclusive webpages

1. Stock.php

This file gives businesses a way to fill out a form as a request to administrators to advertise their investment options available to the users of the website.

Key code:

Insert into requests table

```
$stmt = $conn->prepare("INSERT INTO requests (username, company_name, industry, description, investment_type, logo_url, email, address, phone, status)
```

```
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, 'Pending')");
```

```
$stmt->bind_param("ssssssss", $_SESSION['username'], $company_name, $industry, $description, $investment_type, $logo_url, $email, $address, $phone);
```

2. Notifications.php

This file will provide business users with updates based on their send forms to the administrators. These messages will appear once the administrators select done on requests.

Key code:

Fetch notifications from done_requests table

```
$sql = "SELECT * FROM done_requests WHERE username = ? ORDER BY completed_at DESC";
```

```
$stmt = $conn->prepare($sql);
```

```
$stmt->bind_param("s", $username);
```

3. Investment_options.php

This file will allow business users to choose the investment options they are willing to offer. Those options will be shown on the page one updated and will appear in the respective investment options forms based on what's chosen. For eg, a change to stocks will make the company appear in the stock form and nothing else.

Key code:

Delete existing selections for this user

```
$conn->query("DELETE FROM investment_options WHERE business_username = '$business_username'");
```

Insert new selections for the username

```
$stmt = $conn->prepare("INSERT INTO investment_options (business_username, investment_type) VALUES (?, ?)");
```

```
foreach ($selected_options as $option) {
```

```
    $stmt->bind_param("ss", $business_username, $option);
```

```
    $stmt->execute();
```

Fetch current investment options for display

```
$selected_options_display = [];
```

```
$result = $conn->query("SELECT investment_type FROM investment_options WHERE business_username = '$business_username'");
```

```
while ($row = $result->fetch_assoc()) {
```

```
    $selected_options_display[] = $row['investment_type'];
```

```
}
```

4. Brequests.php

This file will display to the business users the current investment request sent via the forms under the respective investment type section.

Key code:

Handle File Download

```
if (isset($_GET['file'])) {
```

```
    $file_name = basename($_GET['file']);
```

```
    $file_path = __DIR__ . "/downloads/" . $file_name;
```

Handle Request Deletion

```
if (isset($_GET['delete'])) {  
    $request_id = intval($_GET['delete']);  
    $type = $_GET['type'] ?? 'stock';  
  
    $table = $type === 'bond' ? 'bond_requests' : ($type === 'real_estate' ?  
'real_estate_requests' : ($type === 'commodity' ? 'commodity_requests' : 'stock_requests'));
```

Delete record

```
    $delete_query = $conn->prepare("DELETE FROM $table WHERE id = ?");  
    $delete_query->bind_param("i", $request_id);  
    if ($delete_query->execute()) {
```

Fetch Requests

```
$stock_stmt = $conn->prepare("SELECT id, file_path, created_at FROM stock_requests  
WHERE company_name = ?");  
$stock_stmt->bind_param("s", $business_username);  
$stock_stmt->execute();  
$stock_result = $stock_stmt->get_result();
```

Admin exclusive webpages

1. Arequests.php

This file will represent a workstation for admins where they receive requests from the business users and have access to a form in which they can input the information so it can appear in the respective industry.

Key code:

Fetch all requests

```
$sql = "SELECT id, company_name, industry, status FROM requests ORDER BY id DESC";  
$result = $conn->query($sql);
```

Handle form submission

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $company_name = trim($_POST['company_name']);  
    $description = trim($_POST['description']);  
    $header_text = trim($_POST['header_text']);  
    $investment_type = trim($_POST['investment_type']);  
    $company_type = trim($_POST['company_type']);
```

Submenu popup

```
<div id="popup" class="popup">  
    <h3>Request Details</h3>  
    <p id="request-info"></p>  
    <button class="close-btn"  
onclick="document.getElementById('popup').style.display='none'">Close</button>  
    <button class="go-btn" onclick="window.location.href='Ainvestors.php'">Go To</button>  
    <button class="done-btn" onclick="markAsDone()">Done</button>  
</div>
```

2. get_requests.php

this file is a handler of the fetch of requests to the Arequests page.

Key code: `$request_id = intval($_GET['id']);`

```

$sql = "SELECT * FROM requests WHERE id = $request_id";

$result = $conn->query($sql);

if ($row = $result->fetch_assoc()) {

    echo "<strong>Company:</strong> " . $row['company_name'] . "<br>";

    echo "<strong>Email:</strong> " . $row['email'] . "<br>";

    echo "<strong>Address:</strong> " . $row['address'] . "<br>";

    echo "<strong>Phone:</strong> " . $row['phone'] . "<br>";

```

3. Mark_done.php

This file is a handler for the done button function in the Arequests page. It facilitates the transfer of the information to the done.php and to the notifications.php.

Key code:

Insert into done_requests

```

$insertQuery = "INSERT INTO done_requests (username, company_name, industry,
description, investment_type, logo_url, email, address, phone, status, submitted_at)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, 'Completed', ?)";

$insertStmt = $conn->prepare($insertQuery);

$insertStmt->bind_param("sssssssss",

    $request['username'],

    $request['company_name'],

    $request['industry'],

    $request['description'],

    $request['investment_type'],

    $request['logo_url'],

    $request['email'],

    $request['address'],

    $request['phone'],

    $request['submitted_at']

);

$insertStmt->execute();

```

Remove from requests

```
$deleteQuery = "DELETE FROM requests WHERE id = ?";

$deleteStmt = $conn->prepare($deleteQuery);

$deleteStmt->bind_param("i", $requestId);

$deleteStmt->execute();

echo json_encode(["success" => true]);

} else {

    echo json_encode(["success" => false, "message" => "Request not found"]);

}

} else {

    echo json_encode(["success" => false, "message" => "Invalid request"]);

}
```

4. Done.php

This file will offer administrators a log of completed requests and its db store also servers as a fetch destination for the business users notifications.

Key code:

Fetch completed requests

```
$sql = "SELECT * FROM done_requests ORDER BY completed_at DESC";

$result = $conn->query($sql);
```

5. Admin_stars.php

This file allows admins to control who appears on the stars of the day section in the browse section. It displays all users and business users in the database and allows admins to pick via checkbox.

Key code:

Fetch all users and business users

```
$users_stmt = $conn->prepare("SELECT id, username FROM users");
```



```

$users_stmt->execute();

$users_stmt->bind_result($user_id, $user_username);

$users = [];

while ($users_stmt->fetch()) {

    $users[] = [

        'id' => $user_id,

        'username' => $user_username,

        'type' => 'user'

    ];

}

$users_stmt->close();

```

```

$business_stmt = $conn->prepare("SELECT id, username FROM business_users");

$business_stmt->execute();

$business_stmt->bind_result($business_id, $business_username);

$business_users = [];

while ($business_stmt->fetch()) {

    $business_users[] = [

        'id' => $business_id,

        'username' => $business_username,

        'type' => 'business'

    ];

}

$business_stmt->close();

```

Update selected stars

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $selected_users = $_POST['stars'] ?? [];

```

Insert new stars

```

    foreach ($selected_users as $selected) {

```

```
list($id, $type) = explode("_", $selected);
if ($type === "user") {
    $stmt = $conn->prepare("INSERT INTO stars (user_id, business_id) VALUES (?, NULL)");
} else {
    $stmt = $conn->prepare("INSERT INTO stars (user_id, business_id) VALUES (NULL, ?)");
}
```

6. Confirm_business_user.php

This file is the most important for admins as they have the final verdict on the registration of business users accounts. Admins must look out for fake accounts and accept real ones.

This file works by using a 0, 1 and 2 system where 0 is pending, 1 is accepted and 2 is rejected. The outcomes for 1 and 2 will send out emails to the recipient .

Key code:

Handle approval

```
if (isset($_GET['approve'])) {
    $id = intval($_GET['approve']);
    $stmt = $conn->prepare("UPDATE business_users SET status = 1 WHERE id = ?");
    $stmt->bind_param("i", $id);
```

Handle rejection

```
if (isset($_GET['reject'])) {
    $id = intval($_GET['reject']);
    Set status to 2 (rejected)
    $stmt = $conn->prepare("UPDATE business_users SET status = 2 WHERE id = ?");
    $stmt->bind_param("i", $id);
    if ($stmt->execute()) {
```

additional files

1. composer package

used to handle the sending of emails

2. script.js

store of javascripts for functions but most to all functions are within the php files

3. styles.css

store of css styles for visual improvement but most to all css are within the php files

Not as prevalent code

1. check_follow_status.php and fetch_comments.php

these files are not 100% completed as they are outside of the scope but adds to the astatic of the webpage