

ICT 171

# CLOUD SERVER PROJECT DOCUMENTATION

AYAAN NAJIB

34990531

Global IP Address: 13.50.145.230

Domain: <https://financetracker.click/>

GitHub repository:

[https://github.com/Ayaan0311/ICT171\\_Assignment2/tree/9672c792d24264e264562b7e1839e450850975d0](https://github.com/Ayaan0311/ICT171_Assignment2/tree/9672c792d24264e264562b7e1839e450850975d0)

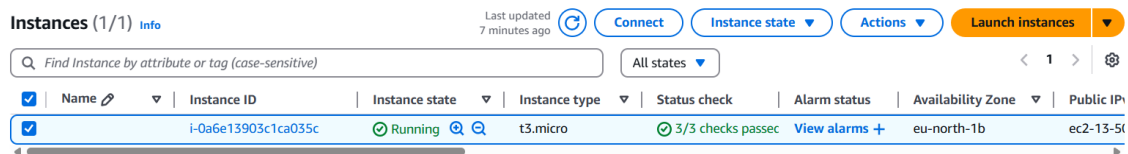
## TABLE OF CONTENT

<b>S. No</b>	<b>Title</b>	<b>Page No.</b>
<b>1</b>	<b>EC2 Instance setup</b>	<b>3</b>
<b>2</b>	<b>Web server installation</b>	<b>3-4</b>
<b>3</b>	<b>Database setup</b>	<b>4-6</b>
<b>4</b>	<b>Backend development</b>	<b>7-14</b>
<b>5</b>	<b>DNS Setup</b>	<b>15</b>
<b>6</b>	<b>HTTPS Setup and SSL/TSL</b>	<b>15</b>
<b>7</b>	<b>Website functioning</b>	<b>16-18</b>

# 1. EC2 Instance Setup

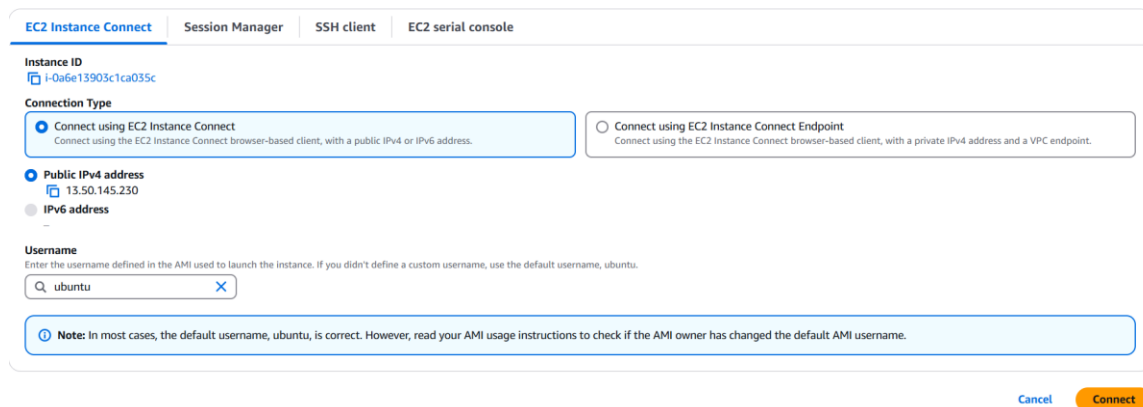
## Step 1: Launch EC2 Instance

- Chose Ubuntu Server 22.04 LTS on AWS EC2.
- Created and downloaded a new .pem key pair.
- Selected t2.micro (free tier) instance type.
- Allowed HTTP, HTTPS, and SSH in the security group settings.



## Step 2: Connect to EC2

Connected using EC2 Instance Connect



# 2. Web Server Installation

## Step 3: Install Apache

```
sudo apt update
sudo apt install apache2 -y
```

```
ubuntu@ip-172-31-34-92:~$ sudo apt update
sudo apt install apache2 -y
```

## Step 4: Install PHP

```
sudo apt install php libapache2-mod-php php-mysql -y
```

```
ubuntu@ip-172-31-34-92:~$ sudo apt install php libapache2-mod-php php-mysql -y
```

### Step 5: Install MySQL

```
sudo apt install mysql-server -y  
sudo mysql_secure_installation
```

```
ubuntu@ip-172-31-34-92:~$ sudo apt install mysql-server -y  
sudo mysql_secure_installation
```

## 3. Database Setup

### Step 6: Create Database and User

```
CREATE DATABASE finance_tracker;  
CREATE USER 'finance_user'@'localhost' IDENTIFIED BY 'An171_2016';  
GRANT ALL PRIVILEGES ON finance_tracker.* TO 'finance_user'@'localhost';  
FLUSH PRIVILEGES;
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| finance_tracker |  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.05 sec)
```

## Step 7: Create Tables

```
mysql> show tables;
+-----+
| Tables_in_finance_tracker |
+-----+
| expenses                   |
| transactions                |
| users                       |
+-----+
```

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL
);
```

```
mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    | auto_increment |
| username   | varchar(50)   | NO   | UNI | NULL    |                 |
| password   | varchar(255)  | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
CREATE TABLE transactions (
  id INT AUTO_INCREMENT PRIMARY KEY,
  type ENUM('income', 'expense') NOT NULL,
  category VARCHAR(255) NOT NULL,
  amount DECIMAL(10, 2) NOT NULL,
  date DATE NOT NULL
);
```

```
mysql> desc transactions;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
type	enum('income','expense')	NO		NULL	
category	varchar(255)	NO		NULL	
amount	decimal(10,2)	NO		NULL	
date	date	NO		NULL	

5 rows in set (0.00 sec)

```
CREATE TABLE expenses (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  user_id INT,  
  amount DECIMAL(10,2) NOT NULL,  
  type ENUM('income','expense') NOT NULL,  
  category VARCHAR(100) NOT NULL,  
  description TEXT,  
  date DATE NOT NULL,  
  FOREIGN KEY (user_id) REFERENCES users(id)  
);
```

```
mysql> desc expenses;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
user_id	int	NO	MUL	NULL	
category	varchar(50)	NO		NULL	
amount	decimal(10,2)	NO		NULL	
date	date	NO		NULL	
description	text	YES		NULL	
type	enum('expense','income')	NO		NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

8 rows in set (0.00 sec)

## 4. Backend Development (PHP)

### db.php - Database Connection

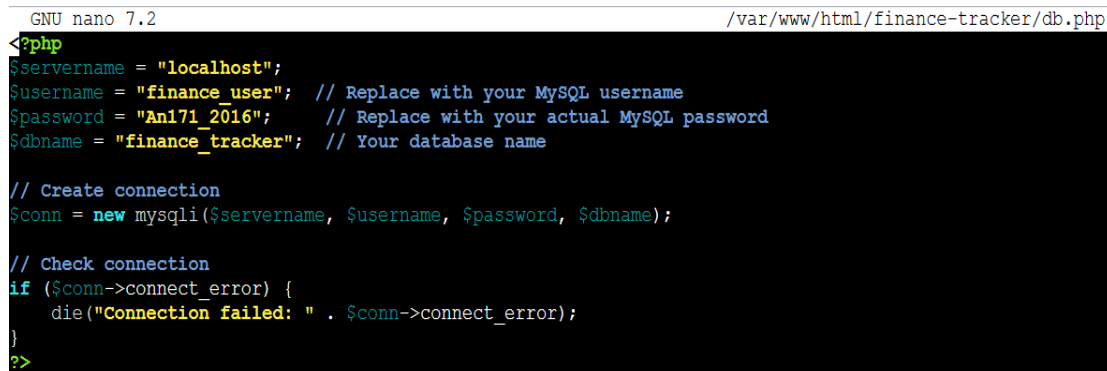
Contains the database connection settings.

Used by all other scripts to connect to the MySQL database. Centralizes the DB credentials for easy maintenance.

```
<?php
$host = 'localhost';
$db   = 'finance_tracker';
$user = 'finance_user';
$pass = 'An171_2016';

$conn = new mysqli($host, $user, $pass, $db);

if ($conn->connect_error) {
    die('Connection failed: ' . $conn->connect_error);
}
?>
```

A screenshot of a GNU nano 7.2 text editor window. The title bar shows the file path /var/www/html/finance-tracker/db.php. The editor contains PHP code for database connection, with comments indicating where to replace credentials. The code is as follows:

```
<?php
$servername = "localhost";
$username = "finance_user"; // Replace with your MySQL username
$password = "An171_2016"; // Replace with your actual MySQL password
$dbname = "finance_tracker"; // Your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

### register.php - User Registration

Contains the database connection settings.

Used by all other scripts to connect to the MySQL database. Centralizes the DB credentials for easy maintenance.

```
<!DOCTYPE html>
<html>
<head>
    <title>Register - Finance Tracker</title>
    <style>
        body { font-family: Arial; display: flex; align-items:
center; justify-content: center; height: 100vh; background-color:
#f4f4f4; }
        .container { background: #fff; padding: 30px; box-shadow:
0px 0px 10px #ccc; border-radius: 8px; text-align: center; width:
```

```

300px; }
    input[type=text], input[type=password] { width: 100%;
padding: 10px; margin: 10px 0; }
    input[type=submit] { padding: 10px 20px; background-
color: #007bff; color: #fff; border: none; cursor: pointer; }
    </style>
</head>
<body>
    <div class="container">
        <h2>Create an Account</h2>
        <form method="POST" action="register.php">
            <input type="text" name="username"
placeholder="Username" required><br>
            <input type="password" name="password"
placeholder="Password" required><br>
            <input type="submit" name="register"
value="Register"><br>
        </form>
        <p>Already have an account? <a href="login.php">Login
here</a></p>
    </div>
</body>
</html>

```

```

GNU nano 7.2 /var/www/html/finance-tracker/register.php
<?php
session_start();
include 'db.php';

$error = '';
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['username']) && isset($_POST['password'])) {
        $username = trim($_POST['username']);
        $password = password_hash(trim($_POST['password']), PASSWORD_DEFAULT);

        $stmt = $conn->prepare("INSERT INTO users (username, password) VALUES (?, ?);");
        if ($stmt) {
            $stmt->bind_param("ss", $username, $password);
            if ($stmt->execute()) {
                header("Location: login.php");
                exit();
            } else {
                $error = "Username already exists or database error.";
            }
            $stmt->close();
        } else {
            $error = "Failed to prepare statement.";
        }
    } else {

```

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^Q Location	M-U Undo	M-R Set Mark	M-I To Bracket	M-Q Previous
^X Exit	^R Read File	^N Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo	M-C Copy	^C Where Was	M-W Next



## login.php - User Login

Handles user login functionality.

Checks the username and password against the database. If credentials are correct, it starts a session and redirects the user to the main dashboard.

```
<?php
session_start();
if (isset($_SESSION['user_id'])) {
    header("Location: index.php");
    exit;
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Login - Finance Tracker</title>
    <style>
        body { font-family: Arial; display: flex; align-items:
center; justify-content: center; height: 100vh; background-color:
#f4f4f4; }
        .container { background: #fff; padding: 30px; box-shadow:
0px 0px 10px #ccc; border-radius: 8px; text-align: center; width:
300px; }
        input[type=text], input[type=password] { width: 100%;
padding: 10px; margin: 10px 0; }
        input[type=submit] { padding: 10px 20px; background-
color: #28a745; color: #fff; border: none; cursor: pointer; }
    </style>
</head>
<body>
    <div class="container">
        <h2>Welcome to the Finance Tracker</h2>
        <p>Register or login to get started.</p>
        <form method="POST" action="login.php">
            <input type="text" name="username"
placeholder="Username" required><br>
            <input type="password" name="password"
placeholder="Password" required><br>
            <input type="submit" name="login" value="Login"><br>
        </form>
        <p>Don't have an account? <a href="register.php">Register
here</a></p>
    </div>
</body>
</html>
```

```

GNU nano 7.2 /var/www/html/finance-tracker/login.php
<?php
session_start();
require 'db.php';

// Redirect to main page if already logged in
if (isset($_SESSION['user_id'])) {
    header("Location: index.php");
    exit();
}

$message = '';
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $stmt = $conn->prepare("SELECT id, password FROM users WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows == 1) {
        $stmt->bind_result($user_id, $hashed_password);
        $stmt->fetch();
    }
}

```

## index.php - Finance Tracker Main Page

Main dashboard for the finance tracker.

Displays a form to input income or expenses, and lists all previous entries with the option to delete them. Shows data in AED and is user-specific based on session.

```

<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}
include 'db.php';

$user_id = $_SESSION['user_id'];

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $type = $_POST['type'];
    $amount = $_POST['amount'];
    $category = $_POST['category'];
    $description = $_POST['description'];
    $date = $_POST['date'];

    $stmt = $conn->prepare("INSERT INTO expenses (user_id, type,
amount, category, description, date) VALUES (?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("isdsss", $user_id, $type, $amount, $category,
    $description, $date);
    $stmt->execute();
    $stmt->close();
}

```

```

if (isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    $stmt = $conn->prepare("DELETE FROM expenses WHERE id = ? AND
user_id = ?");
    $stmt->bind_param("ii", $delete_id, $user_id);
    $stmt->execute();
    $stmt->close();
}

$result = $conn->prepare("SELECT id, type, amount, category,
description, date FROM expenses WHERE user_id = ? ORDER BY date DESC");
$result->bind_param("i", $user_id);
$result->execute();
$result->store_result();
$result->bind_result($id, $type, $amount, $category, $description,
$date);
?>

```

```

<!DOCTYPE html>
<html>
<head>
    <title>Finance Tracker</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #f4f4f4;
            padding: 20px;
            text-align: center;
        }
        .container {
            max-width: 700px;
            margin: auto;
            background: white;
            padding: 30px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
        }
        th, td {
            padding: 12px;
            border: 1px solid #ccc;
        }
    </style>

```

```

th {
    background: #f2f2f2;
}
form {
    margin-bottom: 20px;
}
input, select {
    padding: 10px;
    width: calc(100% - 22px);
    margin-bottom: 10px;
}

.logout-btn {
    float: right;
    background: #e74c3c;
    color: white;
    padding: 8px 16px;
    text-decoration: none;
    border-radius: 4px;
}

.logout-btn:hover {
    background: #c0392b;
}

</style>
</head>
<body>
    <div class="container">
        <a class="logout-btn" href="logout.php">Logout</a>
        <h2>Welcome to your Finance Tracker</h2>
        <form method="post">
            <select name="type" required>
                <option value="">Select Type</option>
                <option value="Income">Income</option>
                <option value="Expense">Expense</option>
            </select>
            <input type="number" name="amount" step="0.01"
placeholder="Amount in AED" required>
            <input type="text" name="category" placeholder="Category"
required>
            <input type="text" name="description"
placeholder="Description" required>
            <input type="date" name="date" required>
            <button type="submit">Add Entry</button>
        </form>
    </div>

```

```

        <tr>
            <th>Type</th>

            <th>Amount (AED)</th>
            <th>Category</th>
            <th>Description</th>
            <th>Date</th>
            <th>Action</th>
        </tr>
        <?php while ($result->fetch()): ?>
            <tr>
                <td><?php echo htmlspecialchars($type); ?></td>
                <td><?php echo number_format($amount, 2); ?>
AED</td>

                <td><?php echo htmlspecialchars($category); ?></td>
                <td><?php echo htmlspecialchars($description);
?></td>

                <td><?php echo htmlspecialchars($date); ?></td>
                <td><a href="?delete=<?php echo $id; ?>"
onclick="return confirm('Are you sure you want to delete this
entry?');">Delete</a></td>
            </tr>
        <?php endwhile; ?>
    </table>
</div>
</body>
</html>

```

```

GNU nano 7.2 /var/www/html/finance-tracker/index.php
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

include 'db.php';

// Handle delete request
if (isset($_POST['delete_entry'])) {
    $delete_id = $_POST['delete_id'];
    $stmt = $conn->prepare("DELETE FROM expenses WHERE id = ? AND user_id = ?");
    $stmt->bind_param("ii", $delete_id, $_SESSION['user_id']);
    $stmt->execute();
    header("Location: index.php");
    exit();
}

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['submit'])) {
    $type = $_POST['type'];
    $category = $_POST['category'];
    $description = $_POST['description'];
}
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-] To Bracket M-Q Previous
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-G Copy      ^Q Where Was  M-W Next

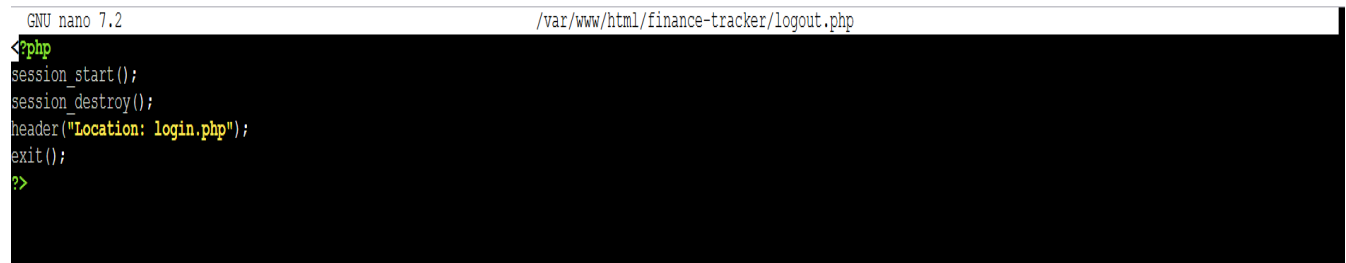
```

## logout.php - Logging Out

Logs the user out of their session.

Destroys the session and redirects the user back to the login page.

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



A screenshot of a terminal window with a black background and light green text. The window title bar shows "GNU nano 7.2" on the left and "/var/www/html/finance-tracker/logout.php" on the right. The code displayed in the terminal is the same as the code block above, with syntax highlighting: "<?php" is green, "session\_start();" is white, "session\_destroy();" is white, "header(\"Location: login.php\");" is white, "exit();" is white, and "?>" is green.

```
GNU nano 7.2 /var/www/html/finance-tracker/logout.php
<?php
session_start();
session_destroy();
header("Location: login.php");
exit();
?>
```

## 5. DNS Setup

For my domain, I purchased it directly from AWS using their route 53 service.

After purchasing the domain, I linked it to the public IP address of my instance by creating a new record in the hosted zone.

Create record [info](#)

Quick create record [Switch to wizard](#)

▼ Record 1 [Delete](#)

Record name [info](#)  Keep blank to create a record for the root domain. financetracker.click

Record type [info](#)

Alias ☒

Value [info](#)

Enter multiple values on separate lines.

TTL (seconds) [info](#)  [1m](#) [1h](#) [1d](#) Recommended values: 60 to 172800 (two days)

Routing policy [info](#)

## 6. HTTPS setup and SSL/TSL

To set up HTTPS for my website I first ensured that HTTPS (port 443) was enabled for my instance under the inbound security rules.

Inbound rules (3) [Manage tags](#) [Edit inbound rules](#)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0505f1c976767522a	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0a7ef66173ad6e503	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-0ae96eaf590c6341f	IPv4	HTTPS	TCP	443

And then, to secure my website with SSL, I used certbot.

### Installing certbot:

```
sudo apt install certbot python3-certbot-apache -y
```

### Run certbot:

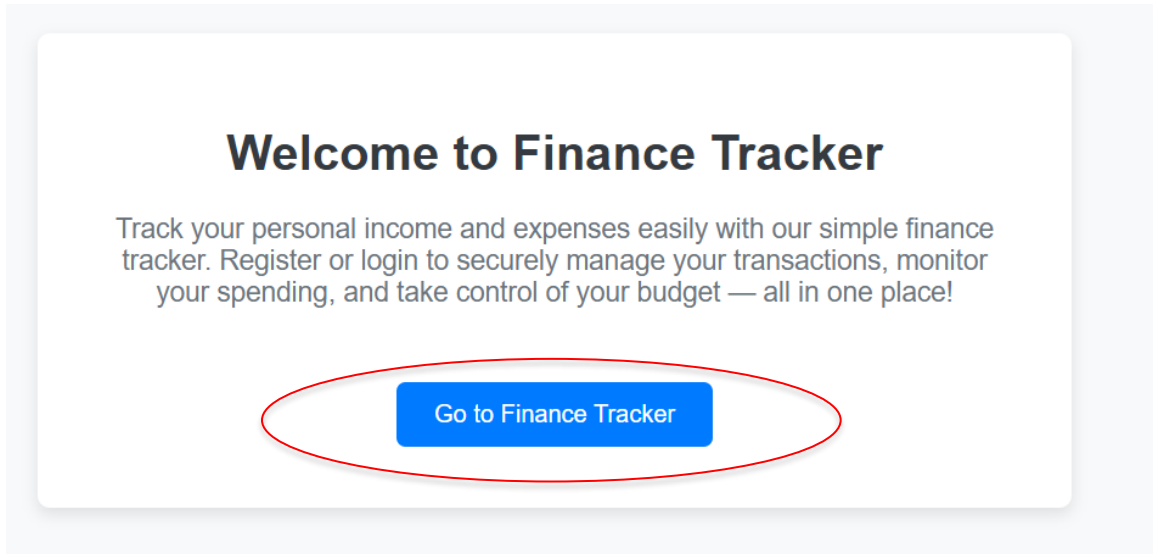
```
sudo certbot --apache
```

followed the prompts and provided my email and domain name to register.

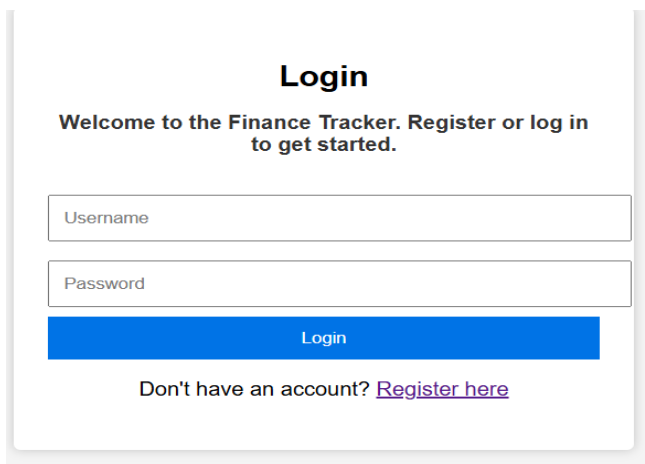
```
ubuntu@ip-172-31-34-92:~$ echo | openssl s_client -connect localhost:443 -servername financetracker.click
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = E5
verify return:1
depth=0 CN = financetracker.click
verify return:1
---
Certificate chain
 0 s:CN = financetracker.click
  i:C = US, O = Let's Encrypt, CN = E5
  a:PKKEY: id-ecPublicKey, 256 (bit); sigalg: ecdsa-with-SHA384
  v:NotBefore: Apr  6 07:42:56 2025 GMT; NotAfter: Jul  5 07:42:55 2025 GMT
 1 s:C = US, O = Let's Encrypt, CN = E5
  i:C = US, O = Internet Security Research Group, CN = ISRG Root X1
  a:PKKEY: id-ecPublicKey, 384 (bit); sigalg: RSA-SHA256
  v:NotBefore: Mar 13 00:00:00 2024 GMT; NotAfter: Mar 12 23:59:59 2027 GMT
```

## 7. Website Functioning:

When you first open the website, this welcome message is displayed. Click on the “Go to Finance Tracker” button to proceed to the finance tracker.



In order to proceed to the finance tracker, you must first login with your username and password.

A screenshot of a login form. The form is white with a light gray border. At the top, the word "Login" is centered in a bold black font. Below it, a line of text reads: "Welcome to the Finance Tracker. Register or log in to get started." There are two input fields: the first is labeled "Username" and the second is labeled "Password". Below these fields is a blue button with the text "Login" in white. At the bottom of the form, there is a line of text: "Don't have an account? [Register here](#)".

If you haven't already created an account, click on “register here” to create your account.



## Create Account

**Register**

[Already have an account? Login](#)

After creating your account and logging in, you can now start recording your transactions.

**Finance Tracker**Logout

Type

Select type

Category

Description

Amount (AED)

Date

dd/mm/yyyy

**Add Entry**

**Your Entries**

Type	Category	Description	Date	Amount (AED)	Action
------	----------	-------------	------	--------------	--------

For the type, you can select either income or expense depending on what you want to record. You can then select the category, add a description, enter the total amount, and select the date of the transaction. Finally, you click “Add Entry” to add the transaction on to the table.

Once you add an entry, it will look like this:

### Your Entries

Type	Category	Description	Date	Amount (AED)	Action
expense	Food	Shawarma	2025-04-07	20.00 ا.د	Delete

If you want to remove an entry, you can click on the “delete” button under action.

Finally, once you are done using the finance tracker, you can log out if you wish to, by clicking the “log out” button on the top right corner of the page.

