Create users table

Got it 👍
Based on your PHP code, your **`users`** table should have the following columns:

- **`id`** → Primary key (auto increment)
- **`name`** → VARCHAR for storing name
- **`email`** → VARCHAR (unique)
- **`password`** → VARCHAR (hashed password)
- **`profile_photo`** → VARCHAR (path of uploaded image)
- **`created_at`** → TIMESTAMP (when registered)

Here's the SQL for your database table:

```sql
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(150) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    profile_photo VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

⚡ Notes:

- **`VARCHAR(255)`** for password because **`password_hash()`** can generate long strings.
- **`email`** is marked **`UNIQUE`** to prevent duplicate registrations.
- **`created_at`** will automatically store registration time.

Do you also want me to make a **login table structure** (for sessions/tokens) or just keep everything in **`users`**?

Great 👍
Your current code logs in a user by storing their info in **`$_SESSION`**. That works fine, but if you want to **track logins and logouts in the database** (audit, activity tracking, multi-device sessions, etc.), you can create a **`user_sessions`** table.

Here's a clean schema for that:

```sql
CREATE TABLE user_sessions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    session_token VARCHAR(255) NOT NULL UNIQUE,
    ip_address VARCHAR(45) DEFAULT NULL,
    user_agent TEXT DEFAULT NULL,
    login_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    logout_time TIMESTAMP NULL,
    status ENUM('active','logged_out') DEFAULT 'active',
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Explanation:

- **user_id** → links to `users.id`.
- **session_token** → you can generate a random token (for cookies or persistent login).
- **ip_address** → to track where the user logged in from.
- **user_agent** → browser/device info (optional but useful).
- **login_time** → automatically records login timestamp.
- **logout_time** → when user logs out, update this.
- **status** → whether the session is active or logged out.

⚡ Usage idea in PHP:

- When login is successful → insert a row in `user_sessions`.
- When logout → update `logout_time` + set `status = 'logged_out'`.

👉 Do you want me to **modify your PHP login/logout code** so it automatically writes into this `user_sessions` table?

Perfect 👍
For your code, the `notes` table needs to store:

- Note ID (primary key)
- User ID (foreign key → `users.id`)
- Title
- Subject
- Description
- File path (where uploaded file is saved)
- Created timestamp

Here's the SQL schema:

```sql
CREATE TABLE notes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    title VARCHAR(150) NOT NULL,
    subject VARCHAR(100) NOT NULL,
    description TEXT,
    file_path VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Explanation:

- `id` → unique identifier for each note.
- `user_id` → links note to the user who uploaded it.
- `title` and `subject` → descriptive fields.
- `description` → extra details about the note.
- `file_path` → stores relative path of uploaded file.
- `created_at` → timestamp when the note was uploaded.
- `ON DELETE CASCADE` → if a user is deleted, their notes also get deleted automatically.

👉 Do you also want me to add a **“last_updated” column** so you can track when a note is edited later?

Got it ✅

Your `dashboard.php` code is already fetching notes, but I see a **small issue**:
In your `notes` table schema earlier, you had `created_at` as the timestamp column.
But in your query, you're ordering with `upload_date`:

```
php

ORDER BY upload_date DESC
```

👉 That will throw an SQL error because `upload_date` doesn't exist.
It should be `created_at`.

Here's the **fixed query section**:

```
php

if (!empty($search)) {
    $search = $conn->real_escape_string($search);
    $sql = "SELECT notes.*, users.name AS uploader_name FROM notes
            JOIN users ON notes.user_id = users.id
            WHERE notes.title LIKE '%$search%'
               OR notes.subject LIKE '%$search%'
               OR notes.description LIKE '%$search%'
            ORDER BY notes.created_at DESC";
} else {
    $sql = "SELECT notes.*, users.name AS uploader_name FROM notes
            JOIN users ON notes.user_id = users.id
            ORDER BY notes.created_at DESC";
}
```

✅ Now, your dashboard will:

- Show all notes (latest first).
- Support searching by title, subject, or description.
- Display uploader's name.
- Show likes + allow liking.

Do you also want me to add a **“My Notes Only” filter button** (so users can quickly see just the notes they uploaded)?

Perfect 👍
I'll give you the **clean, full working `dashboard.php` code** with the fixes applied (using `created_at` instead of `upload_date`) and a proper structure.

Here's the complete code:

```php
<?php
include 'includes/auth.php';
include 'includes/db.php';
include 'includes/header.php';

$user_name = $_SESSION['user_name'];
$user_id = $_SESSION['user_id'];
?>

<!-- Outer Wrapper -->
<div class="min-h-screen bg-gradient-to-br from-white via-blue-50 to-purple-100">

 <!-- Sticky Top Section (Profile + Search) -->
 <div class="sticky top-0 z-40 bg-white shadow-md border-b">
  <div class="max-w-6xl mx-auto p-4 flex flex-col sm:flex-row items-center justify-between gap-4">
   <div class="flex items-center gap-4">
    <img src="<?php echo htmlspecialchars($_SESSION['profile_photo']); ?>"
         class="w-16 h-16 rounded-full object-cover shadow-md" alt="Profile">
    <div>
     <h2 class="text-xl sm:text-2xl font-bold text-gray-800">
       Welcome, <?php echo htmlspecialchars($user_name); ?>!
     </h2>
     <a href="upload.php"
        class="inline-block mt-1 bg-blue-600 text-white text-sm px-4 py-2 rounded hover:bg-blue-
700 transition">
      Upload New Note
     </a>
    </div>
   </div>

   <!-- Search Bar -->
   <form method="GET" action="dashboard.php" class="w-full sm:w-auto flex items-center gap-2">
    <input type="text" name="search" placeholder="Search notes..."
           value="<?php echo htmlspecialchars($_GET['search'] ?? ''); ?>"
           class="w-full sm:w-80 px-4 py-2 border border-gray-300 rounded-lg
                  focus:outline-none focus:ring-2 focus:ring-blue-500">
    <button type="submit"
            class="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700 transition">
     Search
    </button>
   </form>
  </div>
 </div>

 <!-- Main Content -->
 <div class="max-w-6xl mx-auto px-4 py-6">
  <div class="bg-white shadow-md rounded-lg overflow-x-auto">
   <table class="min-w-full text-sm text-left text-gray-700">
    <thead class="bg-gray-100 text-gray-800 font-semibold">
     <tr>
      <th class="px-4 py-4">Title</th>
      <th class="px-4 py-4">Subject</th>
      <th class="px-4 py-4">Description</th>
      <th class="px-4 py-4">Uploaded By</th>
      <th class="px-4 py-4">Actions</th>
     </tr>
    </thead>
    <tbody class="divide-y divide-gray-200">
     <?php
        $search = $_GET['search'] ?? '';

        if (!empty($search)) {
            $search = $conn->real_escape_string($search);
            $sql = "SELECT notes.*, users.name AS uploader_name
                    FROM notes
                    JOIN users ON notes.user_id = users.id
                    WHERE notes.title LIKE '%$search%'
                       OR notes.subject LIKE '%$search%'
                       OR notes.description LIKE '%$search%'
                    ORDER BY notes.created_at DESC";
        } else {
            $sql = "SELECT notes.*, users.name AS uploader_name
                    FROM notes
                    JOIN users ON notes.user_id = users.id
                    ORDER BY notes.created_at DESC";
        }

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

        if ($result && $result->num_rows > 0):
            while ($row = $result->fetch_assoc()):
                $note_id = $row['id'];

                // Likes count
                $like_sql = "SELECT COUNT(*) AS total_likes FROM likes WHERE note_id = $note_id";
                $like_result = $conn->query($like_sql);
                $likes = $like_result->fetch_assoc()['total_likes'];

                // Check if current user already liked
                $check_sql = "SELECT * FROM likes WHERE user_id = $user_id AND note_id =
$note_id";
                $check_res = $conn->query($check_sql);
     ?>
     <tr class="hover:bg-gray-50">
      <td class="px-4 py-4 font-medium"><?php echo htmlspecialchars($row['title']); ?></td>
      <td class="px-4 py-4"><?php echo htmlspecialchars($row['subject']); ?></td>
      <td class="px-4 py-4 max-w-xs break-words">
        <?php echo nl2br(htmlspecialchars($row['description'])); ?>
      </td>
      <td class="px-4 py-4"><?php echo htmlspecialchars($row['uploader_name']); ?></td>
      <td class="px-4 py-4">
       <div class="flex flex-col gap-2 w-36">
        <a href="<?php echo htmlspecialchars($row['file_path']); ?>" target="_blank"
           class="bg-blue-600 hover:bg-blue-700 text-white text-sm px-4 py-2 rounded text-center">
         View
        </a>
        <a href="<?php echo htmlspecialchars($row['file_path']); ?>" download
           class="bg-green-600 hover:bg-green-700 text-white text-sm px-4 py-2 rounded text-
center">
         Download
        </a>
        <div class="text-yellow-600 text-sm text-center font-semibold">
         Likes: <?php echo $likes; ?>
        </div>
        <?php if ($check_res->num_rows == 0): ?>
        <a href="php/like_note.php?note_id=<?php echo $note_id; ?>"
           class="bg-pink-600 hover:bg-pink-700 text-white text-sm px-4 py-2 rounded text-center">
         Like
        </a>
        <?php else: ?>
        <div class="text-green-600 text-sm font-medium text-center">
         You liked this
        </div>
        <?php endif; ?>
       </div>
      </td>
     </tr>
     <?php endwhile; ?>
     <?php else: ?>
     <tr>
      <td colspan="5" class="text-center px-4 py-6 text-gray-500">
        No notes found.
      </td>
     </tr>
     <?php endif; ?>
    </tbody>
   </table>
  </div>
 </div>

</div>
```

```
<?php include 'includes/footer.php'; ?>
```

✅ Fixes applied:

- Changed `upload_date` → `created_at`.
- Escaped user input (`htmlspecialchars`, `$conn->real_escape_string`).
- Added safety for file paths and output.
- Structured code for readability.

👉 Do you also want me to add a **filter button ("My Notes")** in the dashboard so users can quickly see only their uploaded notes?

Nice 👍
For your like system, you'll need a `likes` table that links **users** → **notes**.

Here's the SQL schema:

```sql
CREATE TABLE likes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    note_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY unique_like (user_id, note_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (note_id) REFERENCES notes(id) ON DELETE CASCADE
);
```

Explanation:

- `id` → unique like entry.
- `user_id` → the user who liked.
- `note_id` → the note that was liked.
- `created_at` → when the like happened.
- `UNIQUE KEY (user_id, note_id)` → ensures a user can only like a note **once**.
- `ON DELETE CASCADE` → if a user or note is deleted, their likes are also removed.

👉 Do you also want me to extend this so users can **unlike a note** (toggle like/unlike), instead of blocking them after one like?