

Gästebuch - Umsetzungshilfe

Schritt-für-Schritt zur fertigen Anwendung

Link zur Portainerumgebung <https://bfw-docker.lan>

Anmeldung ist Identisch mit wlan

1. Projektstruktur anlegen

Erstelle folgende Ordnerstruktur:

```
gaestebuch/  
├── docker-compose.yml  
└── src/  
    ├── index.php  
    ├── db.php  
    └── style.css
```

2. Docker-Setup (docker-compose.yml)

Diese Datei beschreibt deine drei Container:

```
version: '3.8'
```

```
services:
```

```
  webserver:
```

```
    image: php:8.2-apache
```

```
    container_name: gaestebuch_web
```

```
    volumes:
```

```
      - ./src:/var/www/html
```

```
    ports:
```

```
      - "8080:80"
```

```
    depends_on:
```

```
      - db
```

```
    command: >
```

```
      bash -c "docker-php-ext-install mysqli && apache2-foreground"
```

```
  db:
```

```
    image: mariadb:latest
```

```
    container_name: gaestebuch_db
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: geheim123
```

```
      MYSQL_DATABASE: gaestebuch_db
```

```
    volumes:
```

```
      - db_data:/var/lib/mysql
```

```
  phpmyadmin:
```

```
    image: phpmyadmin:latest
```

```
    container_name: gaestebuch_pma
```

```
    ports:
```

```
      - "8081:80"
```

```
    environment:
```

```
      PMA_HOST: db
```

```
volumes:
  db_data:
```

Wichtige Erklärungen:

- Port 8080: Webseite erreichbar unter <http://localhost:8080>
- Port 8081: phpMyAdmin erreichbar unter <http://localhost:8081>
- PMA_HOST: db → Container-Name der Datenbank
- Volume db_data: Daten bleiben erhalten, auch nach Container-Neustart

3. Datenbank erstellen

Schritt 1: Container starten

```
docker-compose up -d
```

Schritt 2: phpMyAdmin öffnen

Gehe zu <http://localhost:8081>

Login: root / geheim123

Schritt 3: Tabelle erstellen

Wähle links die Datenbank 'gaestebuch_db', dann 'SQL' anklicken und folgendes ausführen:

```
CREATE TABLE eintraege (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  nachricht TEXT NOT NULL,
  erstellt_am TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

4. Datenbankverbindung (db.php)

Diese Datei verbindet sich zur Datenbank. EINMAL schreiben, überall nutzen!

```
<?php
$host = 'db'; // Container-Name!
$user = 'root';
$pass = 'geheim123';
$dbname = 'gaestebuch_db';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    die("Verbindung fehlgeschlagen: " . $e->getMessage());
}
?>
```

WICHTIG:

- Host ist 'db' (nicht localhost!) → weil das der Container-Name ist
- PDO nutzen (nicht mysqli) → modernerer Standard

5. Hauptseite (index.php) - Stufe für Stufe

Stufe 1: Einfaches HTML-Formular

Erstelle erst mal nur die Struktur:

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Gästebuch</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Gästebuch</h1>

    <form method="post">
        <input type="text" name="name" placeholder="Dein Name" required>
        <textarea name="nachricht" placeholder="Deine Nachricht"
required></textarea>
        <button type="submit" name="submit">Absenden</button>
    </form>
</body>
</html>
```

✓ Test:

Öffne <http://localhost:8080> - siehst du das Formular? Perfekt!

Stufe 2: Formular-Daten empfangen

Füge GANZ OBEN in index.php hinzu (vor <!DOCTYPE>):

```
<?php
if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $nachricht = $_POST['nachricht'];

    echo "<p>Danke, $name! Deine Nachricht: $nachricht</p>";
}
?>
```

✓ Test:

Fülle das Formular aus und sende ab. Wird deine Nachricht angezeigt? Super!

Stufe 3: In Datenbank speichern

Erweitere den PHP-Code oben:

```
<?php
require_once 'db.php';

if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $nachricht = $_POST['nachricht'];
```

```

// WICHTIG: Prepared Statement gegen SQL-Injection!
$sql = "INSERT INTO eintraege (name, nachricht) VALUES (?, ?)";
$stmt = $pdo->prepare($sql);
$stmt->execute([$name, $nachricht]);

echo "<p style='color: green;'>Eintrag gespeichert!</p>";
}
?>

```

✓ Test:

Sende einen Eintrag ab. Gehe zu phpMyAdmin → Tabelle 'eintraege' → siehst du den Eintrag?

Stufe 4: Alle Einträge anzeigen

Füge NACH dem Formular (aber VOR </body>) hinzu:

```

<h2>Bisherige Einträge</h2>

<?php
$sql = "SELECT * FROM eintraege ORDER BY erstellt_am DESC";
$stmt = $pdo->query($sql);

while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    // XSS-Schutz!
    $name_safe = htmlspecialchars($row['name']);
    $nachricht_safe = htmlspecialchars($row['nachricht']);
    $datum = $row['erstellt_am'];

    echo "<div class='eintrag'>";
    echo "<strong>$name_safe</strong> <small>($datum)</small><br>";
    echo "$nachricht_safe";
    echo "</div>";
}
?>

```

✓ Fertig!

Jetzt funktioniert dein Gästebuch komplett! Neue Einträge erscheinen sofort oben.

6. Styling (style.css)

Damit es nicht so trist aussieht:

```

body {
    font-family: Arial, sans-serif;
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
}

input, textarea {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    font-size: 16px;
}

```

```

}

button {
    background-color: #007bff;
    color: white;
    padding: 12px 30px;
    border: none;
    cursor: pointer;
    font-size: 16px;
}

.eintrag {
    background: #f8f9fa;
    padding: 15px;
    margin: 15px 0;
    border-left: 4px solid #007bff;
}

```

7. Häufige Fehler und Lösungen

Problem	Lösung
"Seite kann nicht angezeigt werden"	Prüfen ob Container laufen: <code>docker ps</code>
"Verbindung zur DB fehlgeschlagen"	Host muss 'db' heißen (nicht localhost!) Passwort und Datenbankname prüfen
"mysqli_connect() not found"	PHP-Extension fehlt. In docker-compose.yml ist das 'command:' wichtig!
Formular wird nicht abgeschickt	'name="submit"' beim Button prüfen 'method="post"' beim Form prüfen
Änderungen erscheinen nicht	Browser-Cache leeren (Strg + F5) Datei richtig im src/-Ordner gespeichert?

8. Sicherheits-Checkliste

Bevor du das Projekt abgibst:

- ☒ Prepared Statements verwendet? (Schutz vor SQL-Injection)
- ☒ htmlspecialchars() beim Anzeigen benutzt? (Schutz vor XSS)
- ☒ 'required' bei Formularfeldern gesetzt?
- ☒ Maximale Längen definiert? (name: 100 Zeichen, nachricht: 500)
- ☒ Fehlerbehandlung mit try-catch?

9. Nützliche Befehle

Befehl	Wofür?
<code>docker-compose up -d</code>	Container starten
<code>docker-compose down</code>	Container stoppen
<code>docker-compose restart</code>	Container neu starten
<code>docker ps</code>	Laufende Container anzeigen
<code>docker logs gaestebuch_web</code>	Fehler-Logs anzeigen

10. Erweiterungen (wenn Zeit ist)

Einfach:

- Einträge nach Datum sortieren lassen (ASC/DESC Toggle)
- Zeichenzähler bei Textarea (JavaScript)
- Bestätigungsmeldung verschönern

Mittel:

- Eintrag-Löschen (mit Passwort-Schutz!)
- Pagination (10 Einträge pro Seite)
- Suchfunktion nach Namen

Fortgeschritten:

- Admin-Login für Verwaltung
- E-Mail-Benachrichtigung bei neuem Eintrag
- Captcha gegen Spam

Viel Erfolg bei der Umsetzung! 🚀