

# Dokumentation: Kleinanzeigen-Website

## Projektübersicht:

Die Website dient dazu, Kleinanzeigen dynamisch aus einer MySQL-Datenbank anzuzeigen. Sie bietet Funktionen wie eine Suchleiste, Kategoriefilterung und die Möglichkeit, Anzeigen aufzugeben. Die Seite ist responsiv gestaltet und für Desktop- und mobile Geräte geeignet.

## Index.php

### PHP-Funktionalität:

1. Datenbankverbindung:
  - Verbindung zur MySQL-Datenbank bfw\_datenbank mittels MySQLi.
  - Überprüfung auf Verbindungsfehler, Abruf von Daten aus der Tabelle kleinanzeigen.
2. Anzeige-Logik:
  - Dynamische Anzeige der ersten vier Kleinanzeigen (inkl. Bild, Name, Preis).
  - Inhalte werden aus der Datenbank abgerufen und im Layout positioniert.
3. Such- und Filterfunktion:
  - Suchleiste: Benutzer können Stichworte eingeben, um relevante Anzeigen zu finden. Die Eingabe wird bereinigt und die Ergebnisse dynamisch angezeigt.
  - Kategoriefilter: Filtert nach Kategorien wie Bücher, Technik, Möbel usw., je nach gewählter Option.
4. Verwendete Datenbankfunktionen:
  - showsuche(\$eingabe): Zeigt Suchergebnisse basierend auf Benutzereingabe.
  - showall(): Zeigt alle verfügbaren Anzeigen.
  - Spezifische Methoden wie showBücher(), showTechnik() usw. für kategorisierte Anzeigen.

### HTML-Struktur:

1. Header-Bereich:
  - Logo: ebay\_logo.png.
  - Ein humorvoller Untertitel, um die Website zu beschreiben.
  - Ein Login-Button für Admin-Benutzer, verlinkt zu Loginadmin.php.
2. Banner:
  - Suchleiste: Eingabefeld für Benutzeranfragen mit "Finden"-Button.
  - Button für das Aufgeben neuer Anzeigen, verlinkt zu AnzeigeAufgeben.php.
3. Hauptinhalt:
  - Rubriken: Kategoriefilterung (z. B. Bücher, Technik).
  - Anzeigenbereich:
    - Anzeige von Top-Vorschlägen mit Bild, Name und Preis.
    - Dynamische Suchergebnisse basierend auf Benutzerinteraktion.
  - Ergebnisse werden in einer scrollbaren Ansicht dargestellt.
4. Footer:
  - Ein einfacher Impressum-Bereich mit Projektinformationen.

## CSS (Design):

- Responsive Design: Anpassung der Website für mobile Geräte durch flexible Layouts und prozentuale Größenangaben.
- Farbschema:
  - Grün (#87b817) und Blau (#0463d1) für Buttons.
  - Neutrale Farben (#d7d4d2, #f4e6be) für den Hintergrund.
- Flexbox-Layout:
  - Verwendung von display: flex für zentrierte und ansprechende Layouts.

## Verwendete Technologien:

1. Backend:
  - PHP: Serverseitige Logik, Verarbeitung von Benutzereingaben und Datenbankzugriff.
  - MySQL: Speicherung und Abfrage von Anzeigeninformationen.
2. Frontend:
  - HTML: Struktur der Website.
  - CSS: Styling und responsives Design.
3. Zusätzliche Dateien:
  - DatabaseKlasse.php: Enthält Methoden für Datenbankoperationen.
  - index.css: Externe CSS-Datei für das Layout und Styling.

## Zusammenfassung:

Die Website ist ein vollständiges Kleinanzeigen-Portal mit grundlegenden Funktionen wie Suchen, Anzeigen und Filtern. Die Kombination aus PHP, MySQL und modernem HTML/CSS gewährleistet eine dynamische und ansprechende Benutzeroberfläche.

# AnzeigeAufgaben.php

## Funktionalität:

Diese Seite ermöglicht es Nutzern, eine neue Anzeige aufzugeben, indem sie ein Formular ausfüllen. Die Anzeige wird anschließend in der MySQL-Datenbank gespeichert.

## HTML-Struktur:

1. **Kopfbereich:**
  - Enthält den Titel Anzeige aufgeben.
  - Verlinkung zu einer externen CSS-Datei (index.css) für das Layout.
  - Ein Logo (ebay\_logo.png), das auf die Übersichtsseite AlleAnzeigen.php verlinkt ist.
2. **Formular:**
  - **Kategorienauswahl:** Dropdown-Menü mit vordefinierten Optionen wie Bücher, Technik oder Lebensmittel.
  - **Name:** Eingabefeld für den Titel der Anzeige.
  - **Preis:** Eingabefeld für den Preis (numerisch).
  - **Beschreibung:** Textbereich für Details zur Anzeige.
  - **Absenden-Button:** Übermittelt die Formulareingaben an das Backend.
3. **Navigation:**
  - Link, um zur Startseite (index.php) zurückzukehren.

## PHP-Logik:

### 1. Formularverarbeitung:

- Nach dem Absenden (isset(\$\_POST['submit'])) werden die eingegebenen Daten gesammelt und validiert:
  - Kategorie (\$\_POST['kategorie'])
  - Name (\$\_POST['name'])
  - Preis (\$\_POST['preis'])
  - Beschreibung (\$\_POST['beschreibung'])

### 2. Standardwerte:

- Bild: Standardpfad /images/nophoto.jpg, falls kein Bild hochgeladen wird.
- Datum: Automatisch generiert mit date('Y-m-d').
- Bestätigung: Standardwert 0, da die Anzeige zunächst auf Freigabe wartet.

### 3. Datenbank-Integration:

- Eine Instanz der Klasse Database wird erstellt.
- Die Funktion insertProdukt wird aufgerufen, um die Anzeige in die MySQL-Datenbank zu speichern.

php

Code kopieren

```
$db->insertProdukt(NULL, $kategorie, $name, $preis, $datum, $bild,  
$beschreibung, $bestätigung);
```

### 4. Rückmeldung an den Nutzer:

- Nach erfolgreicher Speicherung: Hinweis, dass die Anzeige auf Bestätigung wartet.

## Verwendete Technologien:

### • Frontend:

- HTML für das Formular.
- CSS für das Styling.

### • Backend:

- PHP zur Verarbeitung der Formulareingaben und Speicherung in der Datenbank.
- Datenbank-Integration über die Methode insertProdukt aus der Klasse Database.

## Besonderheiten:

- **Standardbild:** Anzeigen ohne hochgeladenes Bild verwenden ein Platzhalterbild.
- **Benutzerführung:**
  - Direkte Rückmeldung nach der Eingabe.
  - Links zur Navigation zurück zur Startseite oder Übersicht der Anzeigen.

## Zusammenfassung:

Die Seite bietet eine benutzerfreundliche Möglichkeit, Anzeigen aufzugeben. Mit einem klar strukturierten Formular, standardisierten Eingaben und Datenbankanbindung ist sie für die Erweiterung und Integration in ein größeres Anzeigenportal vorbereitet.

# Adminpage.php

## Funktionalität:

Die Seite dient der Verwaltung von Anzeigen durch den Administrator. Anzeigen, die auf Freigabe warten, können hier bestätigt oder bearbeitet werden.

## HTML-Struktur:

### 1. Kopfbereich:

- Der <head>-Bereich definiert die Seiteneigenschaften:
  - Zeichensatz: UTF-8.
  - Responsives Design mit viewport.
  - Verlinkung zu einer externen CSS-Datei index.css.

### 2. Seiteninhalt:

- **Logo:** Ein Bild mit dem Logo von eBay Kleinanzeigen wird angezeigt.
- **Titel:** Eine humorvolle Überschrift beschreibt die Seite.
- **Logout-Funktion:** Ein Button, der die aktuelle Session beendet und den Nutzer ausloggt.
- **Banner:** Enthält statische Texte, die den Bereich kennzeichnen.
- **Anzeigeübersicht:**
  - Eine kurze Anleitung, wie Anzeigen per Doppelklick bestätigt werden können.
  - Ein Platzhalter (<div class="suche\_kasten">) für dynamisch geladene Anzeigen, die vom Admin bearbeitet werden können.

## PHP-Logik:

### 1. Logout-Funktion:

- Beim Klick auf den *Ausloggen*-Button wird die PHP-Session beendet:  
php  
Code kopieren  
session\_start();  
session\_destroy();
- Der Nutzer wird nach dem Logout auf die Startseite index.php umgeleitet.

### 2. Anzeigeanzeige:

- Die Klasse Database wird eingebunden:  
php  
Code kopieren  
include "DatabaseKlasse.php";
- Die Methode showalladmin() wird aufgerufen, um alle Anzeigen, die auf Freigabe warten, abzurufen und anzuzeigen.

## Verwendete Technologien:

### • Frontend:

- HTML: Struktur der Seite.
- CSS: Styling über index.css.

### • Backend:

- PHP: Verarbeitung von Formularen und Anzeige der Daten.
- MySQL (über DatabaseKlasse): Abruf von Anzeigen aus der Datenbank.

## Funktionalität der Methode showalladmin:

Die Methode showalladmin der Klasse Database lädt alle Anzeigen, die sich im Status "nicht bestätigt" befinden. Der Administrator kann durch Aktionen wie einen Doppelklick diese Anzeigen freigeben. Die spezifische Logik (z. B. Aktualisierung des Status) wird in der DatabaseKlasse implementiert.

**Zusammenfassung:**

Diese Seite stellt die Administrationsoberfläche für Anzeigen dar, die vom Nutzer erstellt wurden. Mit einfachen Mitteln wie der Anzeige wartender Anzeigen und der Möglichkeit, diese zu bestätigen, erleichtert sie die Verwaltung und Moderation der Plattform. Die Logout-Funktion sorgt für eine sichere Trennung zwischen Nutzern und Admins.