

CarShow - GoDaddy-compatible PHP + MySQL

This document contains a ready-to-deploy PHP + MySQL car show website tailored for GoDaddy **shared hosting** (cPanel). Features:

- User event submissions (pending admin approval)
- Admin dashboard for approvals (login required)
- Event filtering by ZIP code, state, or date
- Sponsors page (multiple sponsors)
- Photo gallery with user uploads (pending admin approval)
- Mobile-friendly, automotive-themed design
- External daily scraper (Node.js) to run on Render/Railway that posts scraped events to site API

Project structure

```
/carshow/
├─ public/
│   ├── index.php
│   ├── events.php
│   ├── add_event.php
│   ├── gallery.php
│   ├── sponsors.php
│   ├── upload_handler.php
│   ├── api_receive.php
│   └─ assets/
│       ├── css/style.css
│       └─ uploads/    (writable)
├─ admin/
│   ├── login.php
│   ├── dashboard.php
│   ├── approve_event.php
│   ├── approve_photo.php
│   └─ logout.php
├─ includes/
│   ├── db.php
│   └─ functions.php
├─ sql/
│   └─ schema.sql
└─ scraper/
    └─ scraper_example.js
```

1) SQL schema (sql/schema.sql)

```
CREATE DATABASE IF NOT EXISTS carshow_db DEFAULT CHARACTER SET utf8mb4;  
USE carshow_db;
```

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

```
CREATE TABLE events (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(255) NOT NULL,  
  description TEXT,  
  date DATE,  
  start_time TIME NULL,  
  end_time TIME NULL,  
  address VARCHAR(255),  
  city VARCHAR(100),  
  state VARCHAR(50),  
  zip VARCHAR(20),  
  website VARCHAR(255),  
  contact_email VARCHAR(100),  
  source VARCHAR(255),  
  status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE sponsors (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  website VARCHAR(255),  
  logo VARCHAR(255),  
  description TEXT  
);
```

```
CREATE TABLE photos (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  event_id INT NULL,  
  uploader_email VARCHAR(100),  
  filename VARCHAR(255),  
  caption VARCHAR(255),  
  status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (event_id) REFERENCES events(id) ON DELETE SET NULL
```

```
);
```

```
-- Insert a default admin (password: changeme) - change after install
INSERT INTO admins (username, password_hash) VALUES ('admin',
'$2y$10$Zb1GxF1u7Qw8Fh0/somefakesaltstringhash');
```

Important: Replace the hashed password with a real bcrypt hash after deployment. I'll include instructions below for generating a hash.

2) Database connection (includes/db.php)

```
<?php
// includes/db.php
$DB_HOST = 'localhost';
$DB_NAME = 'carshow_db';
$DB_USER = 'dbuser';
$DB_PASS = 'dbpass';

try {
    $pdo = new PDO("mysql:host=$DB_HOST;dbname=$DB_NAME;charset=utf8mb4",
$DB_USER, $DB_PASS, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ]);
} catch (PDOException $e) {
    die('DB Connection failed: ' . $e->getMessage());
}
```

3) Common functions (includes/functions.php)

```
<?php
// includes/functions.php
require_once __DIR__ . '/db.php';

function esc($s) { return htmlspecialchars($s ?? '', ENT_QUOTES); }

function fetchApprovedEvents($filters = []) {
    global $pdo;
    $sql = "SELECT * FROM events WHERE status = 'approved'";
    $params = [];

    if (!empty($filters['state'])) {
```

```

        $sql .= ' AND state = :state'; $params[':state'] = $filters['state'];
    }
    if (!empty($filters['zip'])) {
        $sql .= ' AND zip = :zip'; $params[':zip'] = $filters['zip'];
    }
    if (!empty($filters['date'])) {
        $sql .= ' AND date = :date'; $params[':date'] = $filters['date'];
    }
    $sql .= ' ORDER BY date ASC';
    $stmt = $pdo->prepare($sql);
    $stmt->execute($params);
    return $stmt->fetchAll();
}

```

4) Public pages

public/index.php

A simple homepage that links to events, add event, gallery, sponsors.

```

<?php
require_once __DIR__ . '/../includes/functions.php';
$events = fetchApprovedEvents();
?>
<!doctype html>
<html>
<head>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Car Show Events</title>
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="site">
    <header class="site-header">
        <h1>CarShow</h1>
        <nav>
            <a href="events.php">Events</a>
            <a href="add_event.php">Add Event</a>
            <a href="gallery.php">Gallery</a>
            <a href="sponsors.php">Sponsors</a>
        </nav>
    </header>
    <main>
        <section class="hero">
            <h2>Find car shows near you</h2>

```

```

        <p>Search by ZIP, state, or date. Submit events and photos for admin
        approval.</p>
    </section>
    <section class="featured">
        <h3>Upcoming Events</h3>
        <?php if (count($events)==0) echo '<p>No approved events yet.</p>';?>
        <ul class="events-list">
            <?php foreach(array_slice($events,0,6) as $e): ?>
                <li>
                    <strong><?=esc($e['title'])?></strong>
                    <div><?=esc($e['date'])?> - <?=esc($e['city'])?>, <?
                    =esc($e['state'])?></div>
                </li>
            <?php endforeach; ?>
        </ul>
    </section>
</main>
<footer class="site-footer">&copy; <?=date('Y')?> CarShow</footer>
</body>
</html>

```

public/events.php (filtering UI + list)

```

<?php
require_once __DIR__ . '/../includes/functions.php';
$filters = [];
if (!empty($_GET['state'])) $filters['state'] = $_GET['state'];
if (!empty($_GET['zip'])) $filters['zip'] = $_GET['zip'];
if (!empty($_GET['date'])) $filters['date'] = $_GET['date'];
$events = fetchApprovedEvents($filters);
?>
<!doctype html>
<html>
<head><meta name="viewport" content="width=device-width,initial-
scale=1"><title>Events</title>
<link rel="stylesheet" href="assets/css/style.css"></head>
<body>
    <header class="site-header"><a href="index.php">CarShow</a></header>
    <main>
        <h2>Events</h2>
        <form method="get" class="filters">
            <input name="zip" placeholder="ZIP" value="<?=esc($_GET['zip'])?>">
            <input name="state" placeholder="State (e.g. GA)" value="<?
            =esc($_GET['state'])?>">
            <input name="date" type="date" value="<?=esc($_GET['date'])?>">
            <button type="submit">Filter</button>

```

```

</form>
<ul class="events-list">
  <?php if (count($events)==0) echo '<li>No events match.</li>'?>
  <?php foreach($events as $e): ?>
    <li>
      <h3><?=esc($e['title'])?></h3>
      <div><?=esc($e['date'])?> – <?=esc($e['city'])?>, <?=esc($e['state'])?
> <?=esc($e['zip'])?></div>
      <p><?=nl2br(esc($e['description']))?></p>
    </li>
  <?php endforeach; ?>
</ul>
</main>
</body>
</html>

```

public/add_event.php (user submission)

```

<?php
require_once __DIR__ . '/../includes/db.php';
if ($_SERVER['REQUEST_METHOD']=='POST') {
  $stmt = $pdo->prepare("INSERT INTO events
(title,description,date,address,city,state,zip,website,contact_email,source,status)
VALUES
(:t,:d,:date,:addr,:city,:state,:zip,:web,:email,'user_submission','pending')");
  $stmt->execute([
    ':t'=>$_POST['title'], ':d'=>$_POST['description'], ':date'=>$_POST['date'],
    ':addr'=>$_POST['address'], ':city'=>$_POST['city'],
    ':state'=>$_POST['state'], ':zip'=>$_POST['zip'],
    ':web'=>$_POST['website'], ':email'=>$_POST['contact_email']
  ]);
  $msg = 'Thanks – your event is pending admin approval.';
}
?>
<!doctype html>
<html><head><meta name="viewport" content="width=device-width,initial-
scale=1"><link rel="stylesheet" href="assets/css/style.css"></head>
<body>
  <h2>Submit an Event</h2>
  <?php if (!empty($msg)) echo '<p class="notice">'.esc($msg).'</p>';?>
  <form method="post">
    <label>Title<input name="title" required></label>
    <label>Date<input name="date" type="date"></label>
    <label>Address<input name="address"></label>
    <label>City<input name="city"></label>
    <label>State<input name="state"></label>

```

```

<label>ZIP<input name="zip"></label>
<label>Website<input name="website"></label>
<label>Contact Email<input name="contact_email" type="email"></label>
<label>Description<textarea name="description"></textarea></label>
<button type="submit">Submit</button>
</form>
</body></html>

```

`public/gallery.php` and `public/upload_handler.php`

The gallery page shows **approved** photos. Uploads go to `upload_handler.php` and are saved as `pending` in DB until admin approves.

`public/gallery.php`:

```

<?php
require_once __DIR__.'../includes/db.php';
$photos = $pdo->query("SELECT * FROM photos WHERE status='approved' ORDER BY
created_at DESC")->fetchAll();
?>
<!doctype html>
<html><head><meta name="viewport" content="width=device-width,initial-
scale=1"><link rel="stylesheet" href="assets/css/style.css"></head>
<body>
  <h2>Photo Gallery</h2>
  <form action="upload_handler.php" method="post" enctype="multipart/form-data">
    <label>Your Email<input name="email" type="email" required></label>
    <label>Event ID (optional)<input name="event_id"></label>
    <label>Photo<input name="photo" type="file" accept="image/*" required></
label>
    <button type="submit">Upload (Pending approval)</button>
  </form>
  <div class="gallery-grid">
    <?php foreach($photos as $p): ?>
      <figure>
        "
        <figcaption><?=esc($p['caption'])?></figcaption>
      </figure>
    <?php endforeach; ?>
  </div>
</body></html>

```

`public/upload_handler.php`:

```

<?php
require_once __DIR__ . '/../includes/db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (empty($_FILES['photo']) || $_FILES['photo']['error'] !== UPLOAD_ERR_OK)
    die('Upload failed');
    $ext = pathinfo($_FILES['photo']['name'], PATHINFO_EXTENSION);
    $filename = uniqid('ph_').'.'.preg_replace('/^[^a-zA-Z0-9]/', '', strtolower($ext));
    $dest = __DIR__ . '/assets/uploads/' . $filename;
    if (!move_uploaded_file($_FILES['photo']['tmp_name'], $dest)) die('Move
    failed');

    $stmt = $pdo->prepare('INSERT INTO photos
    (event_id,uploader_email,filename,caption,status) VALUES
    (:eid,:email,:fn,:cap,"pending")');
    $stmt->execute([
        ':eid'=>$_POST['event_id'] ?: null,
        ':email'=>$_POST['email'], ':fn'=>$filename, ':cap'=>$_POST['caption'] ?? ''
    ]);
    echo 'Upload received – pending admin approval.';
}

```

public/sponsors.php

```

<?php
require_once __DIR__ . '/../includes/db.php';
$spts = $pdo->query('SELECT * FROM sponsors ORDER BY id DESC')->fetchAll();
?>
<!doctype html>
<html><head><meta name="viewport" content="width=device-width,initial-
scale=1"><link rel="stylesheet" href="assets/css/style.css"></head>
<body>
    <h2>Sponsors</h2>
    <div class="sponsor-grid">
        <?php foreach($spts as $s): ?>
            <div class="sponsor-card">
                <?php if ($s['logo']): ?> logo"><?php endif; ?>
                <h3><?=esc($s['name'])?></h3>
                <p><?=esc($s['description'])?></p>
                <?php if ($s['website']): ?><a href="<?=esc($s['website'])?>"
target="_blank">Visit</a><?php endif; ?>
            </div>
        <?php endforeach; ?>
    </div>

```

```
</div>
</body></html>
```

5) Admin area (admin/*)

admin/login.php

```
<?php
session_start();
require_once __DIR__ . '/../includes/db.php';
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $u = $_POST['username']; $p = $_POST['password'];
    $stmt = $pdo->prepare('SELECT * FROM admins WHERE username=:u'); $stmt-
>execute([':u'=>$u]);
    $row = $stmt->fetch();
    if ($row && password_verify($p, $row['password_hash'])) {
        $_SESSION['admin'] = $row['id']; header('Location: dashboard.php'); exit;
    } else $err = 'Login failed';
}
?>
<!doctype html><html><body>
<h2>Admin Login</h2>
<?php if (!empty($err)) echo '<p>'.esc($err).'
```

admin/dashboard.php (approve events/photos)

```
<?php
session_start(); if (empty($_SESSION['admin'])) header('Location: login.php');
require_once __DIR__ . '/../includes/db.php';
$pendingEvents = $pdo->query("SELECT * FROM events WHERE status='pending' ORDER
BY created_at DESC")->fetchAll();
$pendingPhotos = $pdo->query("SELECT * FROM photos WHERE status='pending' ORDER
BY created_at DESC")->fetchAll();
?>
<!doctype html><html><body>
<h2>Admin Dashboard</h2>
<a href="logout.php">Logout</a>
<section>
    <h3>Pending Events</h3>
    <?php foreach($pendingEvents as $e): ?>
```

```

<div>
    <strong><?=esc($e['title'])?></strong>
    <a href="approve_event.php?id=<?=$e['id']?>&action=approve">Approve</a>
    <a href="approve_event.php?id=<?=$e['id']?>&action=reject">Reject</a>
</div>
<?php endforeach; ?>
</section>
<section>
    <h3>Pending Photos</h3>
    <?php foreach($pendingPhotos as $p): ?>
        <div>
            
            <a href="approve_photo.php?id=<?=$p['id']?>&action=approve">Approve</a>
            <a href="approve_photo.php?id=<?=$p['id']?>&action=reject">Reject</a>
        </div>
    <?php endforeach; ?>
</section>
</body></html>

```

admin/approve_event.php and admin/approve_photo.php

These files update `status` for the selected id and redirect back to dashboard.

6) API endpoint for external scraper: public/api_receive.php

This endpoint accepts POSTed JSON from an external scraper and inserts into `events` with `status='pending'`.

```

<?php
require_once __DIR__ . '/../includes/db.php';
$input = json_decode(file_get_contents('php://input'), true);
if (!$input) { http_response_code(400); echo 'Bad JSON'; exit; }
$required = ['title', 'date', 'city', 'state'];
foreach($required as $r) if (empty($input[$r])) { http_response_code(400); echo
'Missing '.$r; exit; }
$stmt = $pdo->prepare('INSERT INTO events
(title,description,date,address,city,state,zip,website,contact_email,source,status)
VALUES (:t,:d,:date,:addr,:city,:state,:zip,:web,:email,:source,"pending")');
$stmt->execute([
    ':t'=>$input['title'], ':d'=>$input['description'] ?? '',
    ':date'=>$input['date'],
    ':addr'=>$input['address'] ?? '', ':city'=>$input['city'],
    ':state'=>$input['state'], ':zip'=>$input['zip'] ?? '',
    ':web'=>$input['website'] ?? '', ':email'=>$input['contact_email'] ?? '',

```

```
'source'=>$input['source'] ?? 'scraper'
]);
http_response_code(201);
echo json_encode(['status'=>'ok']);
```

Security note: When you deploy the scraper, set an **API key** header and check it here. For simplicity the example omitted that; add `if ($_SERVER['HTTP_X_API_KEY'] !== 'your_key') guard.`

7) Minimal CSS (public/assets/css/style.css)

```
/* automotive-themed, responsive */
body{font-family:Inter,Arial,Helvetica,sans-serif;margin:
0;color:#111;background:#0f0f10}
.site-header{background:#111;padding:12px;color:#fff;display:flex;align-
items:center;justify-content:space-between}
.site-header h1{margin:0;font-size:20px;color:#e53935}
nav a{color:#fff;margin-left:12px;text-decoration:none}
.hero{padding:20px;text-align:center;color:#fff}
.events-list{list-style:none;padding:0}
.events-list li{background:#121214;color:#fff;margin:8px;padding:12px;border-
radius:8px}
.gallery-grid{display:grid;grid-template-columns:repeat(auto-fit,minmax(160px,
1fr));gap:8px}
.sponsor-grid{display:flex;flex-wrap:wrap;gap:12px}
.sponsor-card{background:#151517;padding:12px;border-radius:10px;color:#fff;min-
width:180px}
@media(max-width:600px){.site-header{flex-direction:column;align-items:flex-
start}}
```

8) External scraper example (scraper/scraper_example.js)

This is a minimal Node.js scraper that demonstrates posting discovered events to your `api_receive.php`. Host it on Render or Railway and use a cron to run daily.

```
// scraper_example.js (run on a small Node host)
const fetch = require('node-fetch');
const eventsFound = [
  { title: 'Sample Car Meet', date: '2025-11-01', city: 'Augusta', state: 'GA',
    zip: '30901', description: 'Scraped from example.com', website: 'https://
example.com/event', source: 'example.com' }
];
```

```

async function send(e){
  const res = await fetch('https://your-domain.com/public/api_receive.php',{
    method:'POST',headers:{'Content-Type':'application/json','X-API-
KEY':'REPLACE_WITH_KEY'},body:JSON.stringify(e)
  });
  console.log(res.status, await res.text());
}

(async()=>{
  for(const e of eventsFound) await send(e);
})();

```

Notes: - Replace `your-domain.com` and set an `X-API-KEY`. Add API key validation in `api_receive.php`. - For real scraping, use `cheerio` or `puppeteer` and implement robust duplicate detection.

9) Deployment steps (GoDaddy shared hosting)

1. Create MySQL database in cPanel. Create a DB user and assign to the `carshow_db` DB.
2. Edit `includes/db.php` with DB credentials.
3. Upload the `public` folder contents into your site's **public_html** (or desired folder). Upload `includes`, `admin`, `sql` to a safe folder above `public_html` if possible.
4. Import `sql/schema.sql` using phpMyAdmin.
5. Create `assets/uploads` directory and make it writable (`chmod 755` or `775`) via cPanel file manager.
6. Change the default admin password in DB: use a PHP script or `php -r "echo password_hash('newpass', PASSWORD_BCRYPT);"` to get a bcrypt hash and replace in `admins` table.
7. Set file permissions and test site pages.
8. Deploy scraper on Render/Railway and schedule daily runs; use API key security.

10) Security & improvements (recommended)

- Add CSRF protection to forms.
- Add rate-limiting on uploads.
- Validate and resize images server-side.
- Add email notifications for admin on pending items.
- Harden admin area (IP allowlist / 2FA).
- Implement duplicate detection for scraped events (compare title+date+city).

11) Next steps I can do for you

- Generate the full set of files (ready to upload) and package as a ZIP.
 - Replace placeholders with your domain / DB credentials and a real admin password.
 - Expand the scraper to target 2-3 public car event sources (Eventbrite example, local calendars).
-

End of document.