Crowdfunding Platform – JavaScript Project

Deadline: 17-01-2026

Project Overview

Build a **full-stack crowdfunding platform** using **HTML, CSS, JavaScript, JSON Server Package, and JSON Server Auth Package**.
The platform simulates a real crowdfunding system where:

- **Users** can create and support campaigns
- **Admins** moderate users and campaigns
- **Anonymous visitors** can browse approved campaigns

The backend is a **mock REST API** powered by JSON Server with authentication handled by JSON Server Auth.

User Roles & Feature Breakdown

1. Admin (40%)

Admins manage platform integrity through a dedicated dashboard.

Backend Interaction

- Direct interaction with `db.json`
- CRUD operations using JSON Server endpoints

Key Features

User Management

- View all registered users

Ban users by updating:
isActive: false

- in the `users[]` collection

Campaign Moderation

Approve or reject campaigns by updating:
isApproved: true | false

- in the `campaigns[]` collection
- Delete campaigns that violate platform guidelines

Data Access

- View all data stored in:
    - `/users`
    - `/campaigns`
    - `/pledges`

---

2. Registered User (55%)

Authenticated users can launch campaigns and support others.

A. Campaign Creation & Management

Frontend Interaction

- Use JavaScript `fetch()` to communicate with JSON Server

Campaign Creation

Create new campaigns via:
POST /campaigns

- Required fields:
    - `title`
    - `description`
    - `goal`
    - `deadline`
- Upload images using **Base64 encoding**
    - Images stored as strings inside `campaigns[]`

Campaign Updates

Update campaign details using:
PATCH /campaigns/:id

- Editable fields:
  - `deadline`
  - Other campaign metadata

Campaign Insights

Retrieve pledges for a campaign:
GET /pledges?campaignId=:id

B. Supporting Campaigns

Account Management

Register and log in using:
POST /users

- Authentication handled via **JSON Server Auth**

View pledge history:
GET /pledges?userId=:id

Campaign Interaction

Browse and filter campaigns:
GET /campaigns?category=:category&_sort=deadline

- Support campaigns by submitting pledges:
  POST /pledges
- Required pledge fields:
  - `amount`
  - `campaignId`
  - `userId`

---

3. Anonymous User (5%)

No authentication required.

Features
Browse approved campaigns only:
GET /campaigns?isApproved=true

- Search campaigns using query parameters:
GET /campaigns?q=music

Technology Stack

Frontend

- HTML5
- CSS3
- JavaScript (ES6+)

Backend (Mock API)

- **JSON Server**
  - RESTful routes:
    - `/users`
    - `/campaigns`
    - `/pledges`
- **JSON Server Auth**
  - Handles user registration and login
  - Stores credentials in a local JSON file

Database Schema (`db.json`)

```json
{
  "users": [
    {
      "id": 1,
      "name": "Jane Doe",
      "role": "user",
      "isActive": true,
      "email": "jane@example.com",
      "password": "hashed_password"
    }
  ],
  "campaigns": [
    {
      "id": 1,
      "title": "Smart Watch",
```

```json
    "creatorId": 1,
    "goal": 5000,
    "deadline": "31-12-2026",
    "isApproved": false,
  }
 ],
 "pledges": [
  {
   "id": 1,
   "campaignId": 1,
   "userId": 2,
   "amount": 100,
  }
 ]
}
```

Additional Requirements

- **Payments**
  - Implement a **mock payment flow**
  - Use a confirmation dialog (no real payment processing)
- **Version Control**
  - Use Git locally
  - Push all code to GitHub with clear commit history

Design Inspiration

- https://www.gofundme.com
- https://www.kickstarter.com
- https://www.crowdfunding.com

Deliverables

- Fully functional frontend using **ONLY** HTML5, CSS3 and JavaScript (ES6+).
- Mock backend using JSON Server & JSON Server Auth packages.
- GitHub repository with complete source code.
- Project ready for demonstration and PDF documentation.