Avions Club Website - Project Requirements

1. Project Overview

The **Avions Club Website** is a platform for university students involved in aviation projects and drone building. It serves as a hub to showcase **club members, projects, blogs, and resources** in an organized and professional manner.

Goals of the Website

- Provide an **informational hub** for the Avions Club.
- Showcase **club projects** with details and images.
- Allow members to **write and edit markdown-based blogs**.
- Maintain a structured **resource library** for aviation-related content.
- Offer a clean, modern, and responsive **UI/UX**.

Sitemap

Primary Sitemap

Home

Navbar

Hero Header Section
Introduction to Avions Club with a captivating hero image and a brief summary about the club's mission and activities.

Features List Section
Overview of key club activities including drone building, aviation projects, and member collaboration.

Services Section
Description of the main services and activities provided by the club, such as project collaboration and skill workshops.

Benefits Section
Highlight the benefits of joining the Avions Club, such as hands-on experience, networking opportunities, and skill development.

Testimonial Section
Share customer testimonials

CTA Section
Encourage visitors to join the club or attend an upcoming event.

Footer

Projects

Navbar

Header Section
Introduction to the Projects page with a brief overview of the types of aviation and drone projects undertaken by the Avions Club.

Project List Section
A list of all the aviation and drone projects, each with a brief description and a link to the detailed project page.

Stats Section
Showcase key statistics related to the projects, such as the number of projects completed, hours invested, and any awards or recognitions received.

CTA Section
Encourage visitors to get involved in a project or propose a new project idea.

Footer

Blog

Navbar

Featured Blog List Header Section
A header section introducing the blog page with a brief description of the blog's purpose and the type of content readers can expect.

Blog List Section
A list of blog posts with titles, summaries, and publication dates, allowing users to easily browse through the latest articles and click on the ones they want to read in full.

Footer

Blog Post

Navbar

Members

Navbar

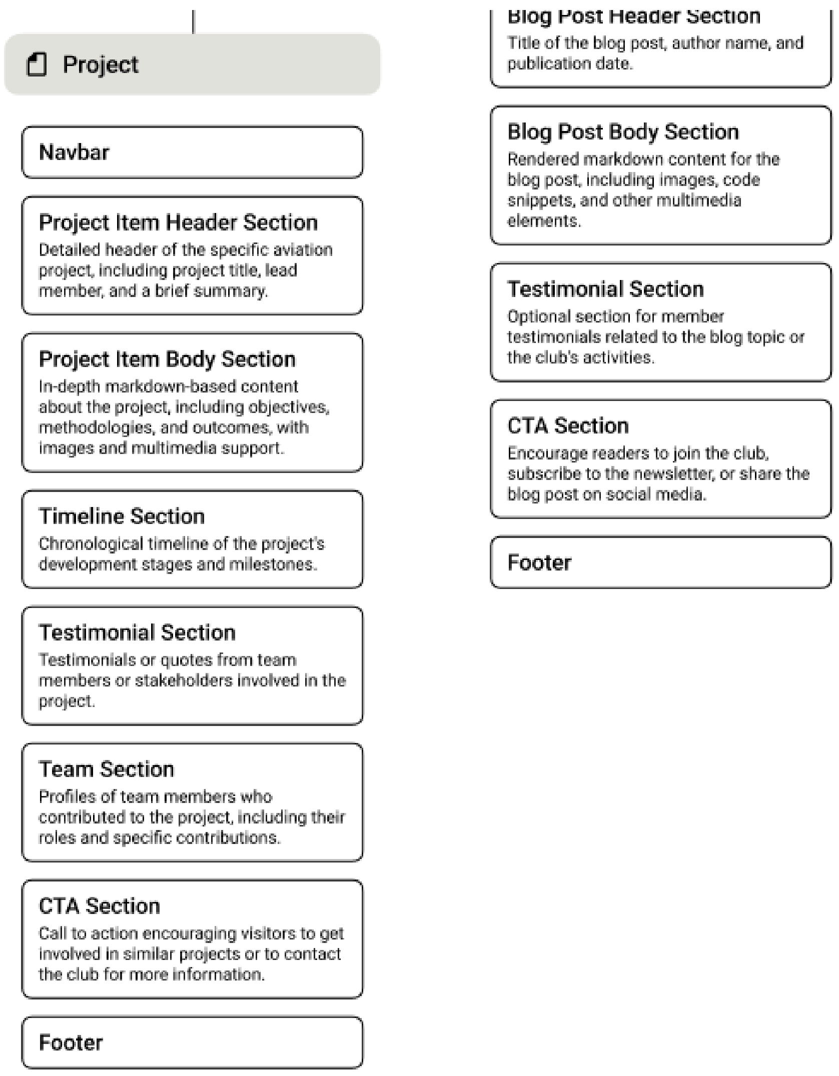
Header Section
Introduction to the club members, emphasizing the diverse skill sets and contributions each member brings to the club.

Team Section
Display profiles of all active members, including their roles, joining dates, and a brief bio or description of their contributions to the club.

Testimonial Section
Showcase testimonials from current and past members about their experiences and the impact the club has had on their personal and professional growth.

Contact Section
Provide contact information or a form for potential members or collaborators to reach out for more information about joining the club.

Footer


Project
Navbar
Project Item Header Section
Detailed header of the specific aviation project, including project title, lead member, and a brief summary.
Project Item Body Section
In-depth markdown-based content about the project, including objectives, methodologies, and outcomes, with images and multimedia support.
Timeline Section
Chronological timeline of the project's development stages and milestones.
Testimonial Section
Testimonials or quotes from team members or stakeholders involved in the project.
Team Section
Profiles of team members who contributed to the project, including their roles and specific contributions.
CTA Section
Call to action encouraging visitors to get involved in similar projects or to contact the club for more information.
Footer
Blog Post Header Section
Title of the blog post, author name, and publication date.
Blog Post Body Section
Rendered markdown content for the blog post, including images, code snippets, and other multimedia elements.
Testimonial Section
Optional section for member testimonials related to the blog topic or the club's activities.
CTA Section
Encourage readers to join the club, subscribe to the newsletter, or share the blog post on social media.
Footer

---

2. Tech Stack & Services Used

Frontend

- **Framework:** Next.js (React-based)
- **Styling:** Tailwind CSS
- **Markdown Rendering:** `react-markdown`
- **Markdown Editing:** `react-simplemde-editor`
- **State Management:** React Hooks (useState, useEffect)

Backend

- **Framework:** Go (Gin)
- **Database:** PostgreSQL (hosted on Neon.tech)
- **API Hosting:** Railway or Render
- **File Storage:** Supabase Storage (for markdown & images)

Deployment

- **Frontend Hosting:** `Vercel`
- **Backend Hosting:** `Railway or Render`
- **Database Hosting:** `Neon.tech (PostgreSQL)`
- **File Storage:** `Supabase Storage (Markdown & images)`

---

3. Folder Structure

```
avions-club-website/
├── backend/                      # Go Backend
│   ├── main.go                   # Entry point
│   ├── handlers/                 # API Handlers
│   │   ├── md_handler.go         # Markdown API
│   │   ├── members_handler.go    # Members API
│   │   ├── projects_handler.go   # Projects API
│   │   ├── admin_handler.go      # Admin API
│   ├── models/                   # Database models
│   │   ├── member.go
│   │   ├── project.go
│   │   ├── blog.go
│   ├── storage/                   # Markdown & Images
│   │   ├── blogs/
│   │   ├── projects/
│   │   ├── resources/
│   ├── database/                  # Database setup
│   │   ├── db.go
│   │   ├── migrations.sql
│   ├── routes/                    # API Routes
│   │   ├── routes.go
├── frontend/                      # Next.js Frontend
│   ├── components/                # UI Components
│   │   ├── Navbar.tsx
│   │   ├── Footer.tsx
│   │   ├── MarkdownEditor.tsx
│   ├── pages/                      # Next.js Pages
│   │   ├── index.tsx              # Home Page
│   │   ├── blogs/
│   │   │   ├── [id].tsx           # Blog Page
│   │   ├── projects/
│   │   │   ├── [id].tsx           # Project Page
│   │   ├── members.tsx            # Members Listing
│   │   ├── admin.tsx              # Admin Dashboard
│   ├── public/                     # Static Assets
│    ├── styles/                   # Global Styles
│    ├── utils/                     # Helper Functions
│    ├── next.config.js
│    ├── package.json
├── .gitignore
├── README.md
```

---

4. Step-by-Step Development Process

Step 1: Setup the Backend

- Initialize a Go (Gin) project.
- Create a **PostgreSQL database** on Neon.tech.
- Set up models for members, projects, blogs, and resources.
- Implement **REST APIs** for CRUD operations.
- Store markdown files & images in Supabase Storage.
- Implement authentication middleware for the `/admin` route.
- Deploy backend to **Railway or Render**.

Step 2: Setup the Frontend

- Initialize a **Next.js project**.
- Configure **Tailwind CSS** for styling.
- Set up **API calls to backend**.
- Implement **dynamic pages** for projects, blogs, and resources.
- Integrate **react-markdown** for rendering markdown.
- Add **Markdown Editor** (`react-simplemde-editor`) for blog creation.
- Create an **Admin Dashboard** (`/admin`) for managing content.
- Deploy frontend to **Vercel**.

Step 3: Implement Admin Authentication & Access Control

- Implement a login system for admins.
- Restrict **/admin** route to authenticated users.
- Verify user credentials via API authentication (JWT-based or Firebase Auth).
- Secure API endpoints related to project and blog management.

Step 4: Testing & Deployment

- **Test API endpoints** using Postman.
- Optimize database queries.
- Deploy both frontend & backend.
- Ensure SSL & CORS policies are configured properly.

---

5. Admin Route (`/admin`) - Concept & Implementation

Concept:

The `/admin` route is a protected dashboard where authorized users (admins) can manage club content, including:

- Adding, editing, and deleting blog posts.
- Managing project details and resources.
- Viewing and updating club members' information.
- Handling site-wide announcements.

How to Achieve It:

Backend (Gin – Go):

1. Implement a middleware for authentication.
2. Use JWT-based authentication for secure admin login.
3. Create CRUD endpoints for managing blogs, projects, and members.
4. Protect sensitive routes with authentication checks.

Frontend (Next.js – React):

1. Create a dedicated `admin.tsx` page for the admin dashboard.
2. Implement a login system with Firebase Auth or a custom backend.
3. Use React hooks to fetch and modify data from the API.
4. Secure the UI using role-based access control.

---

6. Database Schema & Structure

Table: Members

Column	Type	Description
id	UUID (PK)	Unique Member ID
name	VARCHAR(255)	Full Name
position	VARCHAR(100)	Role in the club (President, Member, etc.)
joined_at	TIMESTAMP	Joining Date
image_url	TEXT	Profile Picture

Table: Projects

Column	Type	Description
id	UUID (PK)	Unique Project ID
title	VARCHAR(255)	Project Title
description	TEXT	Short Description
markdown_url	TEXT	Markdown File URL
image_url	TEXT	Project Image
created_at	TIMESTAMP	Creation Date

Table: Blogs

Column	Type	Description
id	UUID (PK)	Unique Blog ID
title	VARCHAR(255)	Blog Title
author_id	UUID (FK)	Linked to Members Table
markdown_url	TEXT	Markdown File URL
created_at	TIMESTAMP	Publish Date

Table: Resources

Column	Type	Description
id	UUID (PK)	Unique Resource ID
title	VARCHAR(255)	Resource Title

Column	Type	Description
markdown_url	TEXT	Markdown File URL
created_at	TIMESTAMP	Upload Date

---

7. Key Features

✅ **Member Profiles** – Display members with roles and images.

✅ **Project Showcase** – Detailed pages for each project.

✅ **Markdown-Based Blogs** – Write & render markdown-based blogs.

✅ **Resources Library** – Store and access club resources easily.

✅ **Markdown Editing** – Allow members to edit blog posts dynamically.

✅ **Responsive UI** – Optimized for desktop & mobile.

✅ **Secure API** – Restrict admin functions.

✅ **Free Hosting** – Vercel (frontend), Railway/Render (backend), Neon.tech (DB), Supabase (files).
