Of course. Here is the consolidated output, combining the file structure and all the detailed prompts into a single, well-formatted document ready for you to export.

Aapda-Prashikshan: Project Development Toolkit

This document contains the complete file structure, detailed AI code generation prompts for each page, and a prompt for creating a professional project proposal.

---

1. Final Project File Structure (HTML, CSS, JS)

This structure is organized for clarity, scalability, and easy maintenance.

```
/aapda-prashikshan/
|
|--- README.md            # Main project overview, setup guide, and key links.
|
|--- login.html           # The primary landing and authentication page.
|--- dashboard.html       # Main dashboard view after successful login.
|--- trainings.html       # Page for viewing, adding, and managing training events.
|--- participants.html    # Page for viewing and managing the cadre registry.
|--- reports.html         # Page for generating and downloading reports.
|
|--- /assets/
|   |--- /css/
|   |   |--- main.css       # Global styles: fonts, colors, navigation, layout.
|   |   |--- login.css      # Styles specific to the login page.
|   |   |--- dashboard.css  # Styles for dashboard components (charts, maps).
|   |   |--- forms.css      # Styles for all data entry forms.
|   |   |--- tables.css     # Styles for all data tables.
|   |
|   |--- /js/
|   |   |--- /lib/         # Folder for third-party libraries (e.g., Chart.js, Leaflet.js).
|   |   |   |--- chart.umd.js
|   |   |   |--- leaflet.js
|   |   |
|   |   |--- config.js      # Central configuration: Supabase URL, API keys.
|   |   |--- auth.js        # Core authentication logic (login, logout, session check).
|   |   |--- main.js        # Global script for shared elements like sidebars/menus.
|   |   |--- dashboard.js   # Logic for fetching data and populating the dashboard.
|   |   |--- trainings.js   # CRUD operations for the training management page.
|   |   |--- participants.js# CRUD operations for the participant registry page.
|   |
|   |--- /images/
|   |   |--- logo.svg       # Main application logo (e.g., NDMA logo).
|   |   |--- favicon.ico    # Browser tab icon.
|   |
|   |--- /icons/
|       |--- dashboard.svg  # Icon for the dashboard link.
|       |--- training.svg   # Icon for the training link.
|       |--- users.svg      # Icon for the participants link.
|       |--- report.svg     # Icon for the reports link.
|
|--- /docs/
    |--- database_schema.sql # SQL script for the database structure.
    |--- api_reference.md    # Documentation for any custom API endpoints.
    |--- project_plan.docx   # Your initial project planning document.
```

---

2. AI Code Generation Prompts ("Vibe Coding")

Here are the individual prompts to generate the HTML and CSS for each page of the application.

Prompt for login.html (Login Page)

Prompt:
"You are an expert front-end developer. Your task is to create the login page for a high-profile government project called **'Aapda-Prashikshan'**, a national disaster management training monitoring system.
The Vibe: The design must be clean, secure, and professional, reflecting an official Government of India portal. The layout should be simple and centered, focusing solely on the authentication task. Use a color scheme of whites and light greys, with a prominent call-to-action button in a navy blue or saffron color.
Task: Generate a single, self-contained HTML file named login.html. The file must include all necessary HTML and inline <style> for the CSS. The page should be responsive and centered both vertically and horizontally.
Specific Requirements for login.html:

1. **Background:** A plain, light grey background color (#f0f2f5).

2. **Login Box:** A single, centered white box with a subtle box shadow and rounded corners. The box should have a maximum width of around 400px.
3. **Content Inside the Box:**
   - **Top:** Display the official NDMA logo (use a placeholder image) followed by the project title "Aapda-Prashikshan" in a bold, professional font.
   - **Form:** A login form with a label and input field for "Email Address," a label and input field for "Password," a prominent "Login" button, and a "Forgot Password?" link.

Do not include the main application header or sidebar on this page."

Prompt for dashboard.html (Dashboard Page)

Prompt:
"You are an expert front-end developer tasked with creating a dashboard for a high-profile government project: **'Aapda-Prashikshan'**.
The Vibe: The design must be clean, professional, and authoritative, like an official Government of India digital platform. It should be data-dense but not cluttered. The color scheme should be inspired by the Indian flag and official aesthetics—primarily whites and light greys, with accents of saffron, navy blue, and green.
Task: Generate a single, self-contained HTML file named dashboard.html with all necessary HTML and inline <style>. The page must be fully responsive and use modern CSS like Flexbox or Grid.
Specific Requirements for dashboard.html:

1. **Layout:** A fixed header at the top and a fixed vertical sidebar on the left.
2. **Header:** Logo and title on the left; user profile and logout button on the right.
3. **Sidebar:** Links with icons for Dashboard, Training Management, Participant Registry, and Reports. "Dashboard" should be the active link.
4. **Main Content Area:**
   - **Top Row:** Four "Stat Cards" for "Total Trainings," "Total Participants," "Active States/UTs," and "Upcoming Events."
   - **Middle Row:** A 60/40 split with a large container for a **Geospatial Training Map** (map-container) on the left and a **Bar Chart** (theme-chart) on the right.
   - **Bottom Row:** A full-width data table for "Recent Training Activities."

Do not use any external CSS frameworks."

Prompt for trainings.html (Training Management Page)

Prompt:
"You are an expert front-end developer building the 'Training Program Management' page for the **'Aapda-Prashikshan'** government project.
The Vibe: The design must be clean, professional, and functional, consistent with the main

dashboard.
Task: Generate a single, self-contained HTML file named trainings.html with all necessary HTML and inline <style>. It must include the standard application layout.
Specific Requirements for trainings.html:

1. **Layout:** Include the same fixed **Header** and **Sidebar Navigation** as the dashboard. The "Training Management" link should be active.
2. **Main Content Area:**
   - **Page Title:** "Training Program Management".
   - **Action Bar:** A search bar on the left and an "Add New Training Program" button on the right.
   - **Data Table:** A full-width table titled "All Training Programs" with sample data. Columns: Program ID, Program Name, Theme, State/UT, Start Date, End Date, Status, and Actions (with "Edit" and "View Details" buttons).

Use the same professional color scheme and typography as the dashboard."

Prompt for participants.html (Participant Registry Page)

Prompt:
"You are an expert front-end developer building the 'National Participant Registry' page for the **'Aapda-Prashikshan'** government project.
The Vibe: The design must be clean, professional, and authoritative, consistent with the main dashboard. The focus is on clarity and searchability.
Task: Generate a single, self-contained HTML file named participants.html with all necessary HTML and inline <style>. It must include the standard application layout.
Specific Requirements for participants.html:

1. **Layout:** Include the same fixed **Header** and **Sidebar Navigation** as the dashboard. The "Participant Registry" link should be active.
2. **Main Content Area:**
   - **Page Title:** "National Participant Registry".
   - **Action Bar:** Dropdown filters for "State" and "Skill" on the left; an "Add New Participant" button on the right.
   - **Data Table:** A full-width table titled "Registered Personnel" with sample data. Columns: Participant ID, Name, State/UT, Primary Skill, Last Training Attended, and Certification Status (with a "View Profile" button).

Use the same professional color scheme and typography as the dashboard."

Prompt for reports.html (Reports Page)

Prompt:
"You are an expert front-end developer building the 'Report Generation' page for the **'Aapda-Prashikshan'** government project.

The Vibe: The design must be clean, functional, and intuitive for officials to extract data.
Task: Generate a single, self-contained HTML file named reports.html with all necessary HTML and inline <style>. It must include the standard application layout.
Specific Requirements for reports.html:

1. **Layout:** Include the same fixed **Header** and **Sidebar Navigation** as the dashboard. The "Reports" link should be active.
2. **Main Content Area:**
   - **Page Title:** "Report Generation".
   - **Report Configuration Panel:** A container with form controls: dropdown for "Report Type," "Start Date" and "End Date" inputs, and dropdowns for "State/UT" and "Training Theme." A large "Generate Report" button at the bottom.
   - **Generated Reports Section:** A container below the panel titled "Generated Reports" with a placeholder message and a sample row showing a report name and a "Download" button.

Use the same professional color scheme and typography as the dashboard."

---

3. Prompt for Project Proposal Document

Use this prompt to generate a comprehensive project overview document in a tool like Google Docs.
Prompt:
"Generate a detailed and professionally formatted project proposal document for a software project named **'Aapda-Prashikshan: A National Real-Time Disaster Management Training Monitoring System'**.
The document should be structured logically for review by senior officials at the **Ministry of Home Affairs** and the **National Disaster Management Authority (NDMA)**. The tone should be formal, confident, and clear.
Context: The core problem is that India's disaster management training is fragmented, with no centralized system to track activities in real-time. This creates a "critical blind spot" in national preparedness. 'Aapda-Prashikshan' solves this by being a unified, geo-aware web platform. Its key innovation is the concept of "Skill Density," visualized on a heatmap, to ensure the right skills are in high-risk areas. The tech stack is a JavaScript frontend with a Supabase (PostgreSQL) backend.
Document Sections to Generate:

1. **Executive Summary:** A brief, high-level overview of the project. State the problem, the proposed solution, and the expected impact in one concise paragraph.
2. **1.0 Problem Statement:**
   - **1.1 Current Systemic Gaps:** Describe the fragmented nature of the current training ecosystem.
   - **1.2 The "Preparedness Paradox":** Explain why existing platforms (IDRN, NDMIS) are insufficient.
3. **2.0 Proposed Solution: 'Aapda-Prashikshan'**

   - **2.1 Vision:** State the vision of the project—to create a "single pane of glass" for decision-makers.
   - **2.2 Core Modules:** Detail the four main modules (Training Management, Participant Registry, Geospatial Dashboard, Reporting & Analytics).
   - **2.3 Key Innovation - "Skill Density":** Explain this unique value proposition.

4. **3.0 Technical Architecture:**
   - **3.1 Technology Stack:** List the chosen technologies.
   - **3.2 Justification:** Explain why the stack was chosen, emphasizing security (Row-Level Security) and scalability.
5. **4.0 Impact and National Benefits:**
   - **4.1 Governance:** How it fosters transparency and data-driven policymaking.
   - **4.2 Social & Economic:** How it enhances national resilience and optimizes training investments.
   - **4.3 Strategic Alignment:** Mention alignment with the Prime Minister's 10-Point Agenda on Disaster Risk Reduction and the Sendai Framework.

Please format the entire output with clear headings, subheadings, and bullet points."