Bank of AJO - Funder Portal: Project Overview

This document provides a comprehensive overview of the Bank of AJO Funder Portal application. It includes details on the project's purpose, structure, and functionality, as well as the full source code for all relevant files.

1. Project Purpose

The Bank of AJO Funder Portal is a single-page web application (SPA) that allows internal users to create and manage credit proposals. It provides a simple interface for entering applicant and loan details, and it persists this information in a local SQLite database. The application also generates a PDF summary of each proposal.

2. Project Structure

The project is organized into the following files and directories:

- `index.html` : The main entry point for the application. It contains the basic HTML structure and links to the necessary CSS and JavaScript files.
- `css/style.css` : Contains all the styles for the application, including a design system with light and dark modes.
- `js/app.js` : The core of the application. This file contains all the client-side logic for routing, state management, and rendering the different views of the application.
- `server.js` : A simple Node.js/Express server that provides a REST API for interacting with the SQLite database.
- `database.sqlite` : The SQLite database file where all proposal data is stored.
- `package.json` : Defines the project's dependencies and scripts.
- `vendor/jspdf.umd.min.js` : A local copy of the jsPDF library, used for generating PDFs.

3. How it Works

The application is a single-page application (SPA) that uses a hash-based routing system to navigate between different views. The core logic is contained within the `js/app.js` file.

3.1. Authentication

The application has a simple authentication system with a hardcoded list of users and passwords. When a user successfully logs in, their username is stored in the global `state` object, and they are redirected to the dashboard.

3.2. Routing

The `router` function in `js/app.js` is responsible for handling changes in the URL hash and rendering the appropriate view. It also includes navigation guards to prevent unauthenticated users from accessing protected pages.

3.3. Views

The application has the following views:

- **Login (`#login`)**: A simple login form.
- **Dashboard (`#dashboard`)**: The main landing page after logging in. It provides links to create a new proposal or view existing submissions.
- **New Proposal (`#new`)**: A form for creating a new credit proposal.
- **Submissions (`#submissions`)**: A table displaying all submitted proposals.

- **Confirmation (`#confirm/:id`)**: A page that confirms the successful submission of a proposal.

3.4. Data Persistence

The application uses a simple Node.js/Express server (`server.js`) to provide a REST API for interacting with a SQLite database. The API has endpoints for getting all proposals, creating a new proposal, and deleting a proposal.

3.5. PDF Generation

When a new proposal is submitted, the application uses the jsPDF library to generate a PDF summary of the proposal details. The PDF is then saved as a data URI and can be downloaded from the submissions page.

4. Source Code

`index.html`

```
<!DOCTYPE html>
<html lang="en" data-color-scheme="light">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Bank of AJO – Funder Portal</title>
    <!-- Bootstrap 5 CSS -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-STqaR2zt2MtwLecxB2S/bR5E2wE8p8nH3UX0YPD6/COM24kTx5c5NyEJD7BqXc9N"
      crossorigin="anonymous"
    />
    <!-- Custom Styles   -->
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <div id="app"></div>

    <!-- Bootstrap 5 Bundle (incl. Popper) -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-Gr8XEZTknmo9dZYkTfGZNVVRFlE0YyqS/fWznuaCG2lKT9IAXoJ1i8L6Rxur+8W0"
      crossorigin="anonymous"
    ></script>
    <!-- jsPDF for PDF generation (local copy to avoid CDN issues) -->
    <script src="vendor/jspdf.umd.min.js"></script>
    <!-- Main application script -->
    <script src="js/app.js"></script>
  </body>
</html>
```

`css/style.css`

```
:root {
  /* Primitive Color Tokens */
  --color-white: rgba(255, 255, 255, 1);
  --color-black: rgba(0, 0, 0, 1);
  --color-cream-50: rgba(252, 252, 249, 1);
  --color-cream-100: rgba(255, 255, 253, 1);
  --color-gray-200: rgba(245, 245, 245, 1);
  --color-gray-300: rgba(167, 169, 169, 1);
  --color-gray-400: rgba(119, 124, 124, 1);
  --color-slate-500: rgba(98, 108, 113, 1);
  --color-brown-600: rgba(94, 82, 64, 1);
  --color-charcoal-700: rgba(31, 33, 33, 1);
  --color-charcoal-800: rgba(38, 40, 40, 1);
  --color-slate-900: rgba(19, 52, 59, 1);
  --color-teal-300: rgba(50, 184, 198, 1);
  --color-teal-400: rgba(45, 166, 178, 1);
  --color-teal-500: rgba(33, 128, 141, 1);
  --color-teal-600: rgba(29, 116, 128, 1);
  --color-teal-700: rgba(26, 104, 115, 1);
  --color-teal-800: rgba(41, 150, 161, 1);
  --color-red-400: rgba(255, 84, 89, 1);
  --color-red-500: rgba(192, 21, 47, 1);
  --color-orange-400: rgba(230, 129, 97, 1);
  --color-orange-500: rgba(168, 75, 47, 1);

  /* RGB versions for opacity control */
  --color-brown-600-rgb: 94, 82, 64;
  --color-teal-500-rgb: 33, 128, 141;
  --color-slate-900-rgb: 19, 52, 59;
  --color-slate-500-rgb: 98, 108, 113;
  --color-red-500-rgb: 192, 21, 47;
  --color-red-400-rgb: 255, 84, 89;
  --color-orange-500-rgb: 168, 75, 47;
  --color-orange-400-rgb: 230, 129, 97;

  /* Background color tokens (Light Mode) */
  --color-bg-1: rgba(59, 130, 246, 0.08); /* Light blue */
  --color-bg-2: rgba(245, 158, 11, 0.08); /* Light yellow */
  --color-bg-3: rgba(34, 197, 94, 0.08); /* Light green */
  --color-bg-4: rgba(239, 68, 68, 0.08); /* Light red */
  --color-bg-5: rgba(147, 51, 234, 0.08); /* Light purple */
  --color-bg-6: rgba(249, 115, 22, 0.08); /* Light orange */
  --color-bg-7: rgba(236, 72, 153, 0.08); /* Light pink */
  --color-bg-8: rgba(6, 182, 212, 0.08); /* Light cyan */

  /* Semantic Color Tokens (Light Mode) */
  --color-background: var(--color-cream-50);
  --color-surface: var(--color-cream-100);
  --color-text: var(--color-slate-900);
  --color-text-secondary: var(--color-slate-500);
  --color-primary: var(--color-teal-500);
```

```css
  --color-primary-hover: var(--color-teal-600);
  --color-primary-active: var(--color-teal-700);
  --color-secondary: rgba(var(--color-brown-600-rgb), 0.12);
  --color-secondary-hover: rgba(var(--color-brown-600-rgb), 0.2);
  --color-secondary-active: rgba(var(--color-brown-600-rgb), 0.25);
  --color-border: rgba(var(--color-brown-600-rgb), 0.2);
  --color-btn-primary-text: var(--color-cream-50);
  --color-card-border: rgba(var(--color-brown-600-rgb), 0.12);
  --color-card-border-inner: rgba(var(--color-brown-600-rgb), 0.12);
  --color-error: var(--color-red-500);
  --color-success: var(--color-teal-500);
  --color-warning: var(--color-orange-500);
  --color-info: var(--color-slate-500);
  --color-focus-ring: rgba(var(--color-teal-500-rgb), 0.4);
  --color-select-caret: rgba(var(--color-slate-900-rgb), 0.8);

  /* Common style patterns */
  --focus-ring: 0 0 0 3px var(--color-focus-ring);
  --focus-outline: 2px solid var(--color-primary);
  --status-bg-opacity: 0.15;
  --status-border-opacity: 0.25;
  --select-caret-light: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23134252' stroke-width='2'
stroke-linecap='round' stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");
  --select-caret-dark: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23f5f5f5' stroke-width='2'
stroke-linecap='round' stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");

  /* RGB versions for opacity control */
  --color-success-rgb: 33, 128, 141;
  --color-error-rgb: 192, 21, 47;
  --color-warning-rgb: 168, 75, 47;
  --color-info-rgb: 98, 108, 113;

  /* Typography */
  --font-family-base: "FKGroteskNeue", "Geist", "Inter", -apple-system,
    BlinkMacSystemFont, "Segoe UI", Roboto, sans-serif;
  --font-family-mono: "Berkeley Mono", ui-monospace, SFMono-Regular, Menlo,
    Monaco, Consolas, monospace;
  --font-size-xs: 11px;
  --font-size-sm: 12px;
  --font-size-base: 14px;
  --font-size-md: 14px;
  --font-size-lg: 16px;
  --font-size-xl: 18px;
  --font-size-2xl: 20px;
  --font-size-3xl: 24px;
  --font-size-4xl: 30px;
  --font-weight-normal: 400;
  --font-weight-medium: 500;
  --font-weight-semibold: 550;
  --font-weight-bold: 600;
  --line-height-tight: 1.2;
  --line-height-normal: 1.5;
  --letter-spacing-tight: -0.01em;

  /* Spacing */
  --space-0: 0;
  --space-1: 1px;
  --space-2: 2px;
  --space-4: 4px;
  --space-6: 6px;
  --space-8: 8px;
  --space-10: 10px;
  --space-12: 12px;
  --space-16: 16px;
  --space-20: 20px;
  --space-24: 24px;
  --space-32: 32px;

  /* Border Radius */
  --radius-sm: 6px;
  --radius-base: 8px;
  --radius-md: 10px;
  --radius-lg: 12px;
  --radius-full: 9999px;

  /* Shadows */
  --shadow-xs: 0 1px 2px rgba(0, 0, 0, 0.02);
  --shadow-sm: 0 1px 3px rgba(0, 0, 0, 0.04), 0 1px 2px rgba(0, 0, 0, 0.02);
  --shadow-md: 0 4px 6px -1px rgba(0, 0, 0, 0.04),
    0 2px 4px -1px rgba(0, 0, 0, 0.02);
  --shadow-lg: 0 10px 15px -3px rgba(0, 0, 0, 0.04),
    0 4px 6px -2px rgba(0, 0, 0, 0.02);
  --shadow-inset-sm: inset 0 1px 0 rgba(255, 255, 255, 0.15),
    inset 0 -1px 0 rgba(0, 0, 0, 0.03);

  /* Animation */
  --duration-fast: 150ms;
  --duration-normal: 250ms;
  --ease-standard: cubic-bezier(0.16, 1, 0.3, 1);

  /* Layout */
  --container-sm: 640px;
  --container-md: 768px;
  --container-lg: 1024px;
  --container-xl: 1280px;
}

/* Dark mode colors */
@media (prefers-color-scheme: dark) {
  :root {
    /* RGB versions for opacity control (Dark Mode) */
    --color-gray-400-rgb: 119, 124, 124;
    --color-teal-300-rgb: 50, 184, 198;
    --color-gray-300-rgb: 167, 169, 169;
    --color-gray-200-rgb: 245, 245, 245;

    /* Background color tokens (Dark Mode) */
    --color-bg-1: rgba(29, 78, 216, 0.15); /* Dark blue */
    --color-bg-2: rgba(180, 83, 9, 0.15); /* Dark yellow */
    --color-bg-3: rgba(21, 128, 61, 0.15); /* Dark green */
    --color-bg-4: rgba(185, 28, 28, 0.15); /* Dark red */
    --color-bg-5: rgba(107, 33, 168, 0.15); /* Dark purple */
    --color-bg-6: rgba(194, 65, 12, 0.15); /* Dark orange */
    --color-bg-7: rgba(190, 24, 93, 0.15); /* Dark pink */
    --color-bg-8: rgba(8, 145, 178, 0.15); /* Dark cyan */

    /* Semantic Color Tokens (Dark Mode) */
    --color-background: var(--color-charcoal-700);
    --color-surface: var(--color-charcoal-800);
    --color-text: var(--color-gray-200);
    --color-text-secondary: rgba(var(--color-gray-300-rgb), 0.7);
    --color-primary: var(--color-teal-300);
    --color-primary-hover: var(--color-teal-400);
    --color-primary-active: var(--color-teal-800);
    --color-secondary: rgba(var(--color-gray-400-rgb), 0.15);
    --color-secondary-hover: rgba(var(--color-gray-400-rgb), 0.25);
    --color-secondary-active: rgba(var(--color-gray-400-rgb), 0.3);
    --color-border: rgba(var(--color-gray-400-rgb), 0.3);
    --color-error: var(--color-red-400);
    --color-success: var(--color-teal-300);
    --color-warning: var(--color-orange-400);
    --color-info: var(--color-gray-300);
    --color-focus-ring: rgba(var(--color-teal-300-rgb), 0.4);
    --color-btn-primary-text: var(--color-slate-900);
    --color-card-border: rgba(var(--color-gray-400-rgb), 0.2);
    --color-card-border-inner: rgba(var(--color-gray-400-rgb), 0.15);
    --shadow-inset-sm: inset 0 1px 0 rgba(255, 255, 255, 0.1),
      inset 0 -1px 0 rgba(0, 0, 0, 0.15);
    --button-border-secondary: rgba(var(--color-gray-400-rgb), 0.2);
    --color-border-secondary: rgba(var(--color-gray-400-rgb), 0.2);
    --color-select-caret: rgba(var(--color-gray-200-rgb), 0.8);

    /* Common style patterns - updated for dark mode */
    --focus-ring: 0 0 0 3px var(--color-focus-ring);
    --focus-outline: 2px solid var(--color-primary);
    --status-bg-opacity: 0.15;
    --status-border-opacity: 0.25;
    --select-caret-light: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23134252' stroke-width='2'
stroke-linecap='round' stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");
    --select-caret-dark: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23f5f5f5' stroke-width='2'
stroke-linecap='round' stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");

    /* RGB versions for dark mode */
    --color-success-rgb: var(--color-teal-300-rgb);
    --color-error-rgb: var(--color-red-400-rgb);
    --color-warning-rgb: var(--color-orange-400-rgb);
    --color-info-rgb: var(--color-gray-300-rgb);
  }
}

/* Data attribute for manual theme switching */
[data-color-scheme="dark"] {
  /* RGB versions for opacity control (dark mode) */
  --color-gray-400-rgb: 119, 124, 124;
  --color-teal-300-rgb: 50, 184, 198;
  --color-gray-300-rgb: 167, 169, 169;
  --color-gray-200-rgb: 245, 245, 245;

  /* Colorful background palette - Dark Mode */
  --color-bg-1: rgba(29, 78, 216, 0.15); /* Dark blue */
  --color-bg-2: rgba(180, 83, 9, 0.15); /* Dark yellow */
  --color-bg-3: rgba(21, 128, 61, 0.15); /* Dark green */
  --color-bg-4: rgba(185, 28, 28, 0.15); /* Dark red */
  --color-bg-5: rgba(107, 33, 168, 0.15); /* Dark purple */
  --color-bg-6: rgba(194, 65, 12, 0.15); /* Dark orange */
  --color-bg-7: rgba(190, 24, 93, 0.15); /* Dark pink */
  --color-bg-8: rgba(8, 145, 178, 0.15); /* Dark cyan */

  /* Semantic Color Tokens (Dark Mode) */
  --color-background: var(--color-charcoal-700);
  --color-surface: var(--color-charcoal-800);
  --color-text: var(--color-gray-200);
  --color-text-secondary: rgba(var(--color-gray-300-rgb), 0.7);
  --color-primary: var(--color-teal-300);
  --color-primary-hover: var(--color-teal-400);
  --color-primary-active: var(--color-teal-800);
  --color-secondary: rgba(var(--color-gray-400-rgb), 0.15);
  --color-secondary-hover: rgba(var(--color-gray-400-rgb), 0.25);
  --color-secondary-active: rgba(var(--color-gray-400-rgb), 0.3);
  --color-border: rgba(var(--color-gray-400-rgb), 0.3);
  --color-error: var(--color-red-400);
  --color-success: var(--color-teal-300);
  --color-warning: var(--color-orange-400);
  --color-info: var(--color-gray-300);
  --color-focus-ring: rgba(var(--color-teal-300-rgb), 0.4);
  --color-btn-primary-text: var(--color-slate-900);
  --color-card-border: rgba(var(--color-gray-400-rgb), 0.15);
  --color-card-border-inner: rgba(var(--color-gray-400-rgb), 0.15);
  --shadow-inset-sm: inset 0 1px 0 rgba(255, 255, 255, 0.1),
    inset 0 -1px 0 rgba(0, 0, 0, 0.15);
  --color-border-secondary: rgba(var(--color-gray-400-rgb), 0.2);
  --color-select-caret: rgba(var(--color-gray-200-rgb), 0.8);

  /* Common style patterns - updated for dark mode */
  --focus-ring: 0 0 0 3px var(--color-focus-ring);
  --focus-outline: 2px solid var(--color-primary);
  --status-bg-opacity: 0.15;
  --status-border-opacity: 0.25;
  --select-caret-light: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23134252' stroke-width='2'
stroke-linecap='round' stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");
  --select-caret-dark: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' viewBox='0 0 24 24' fill='none' stroke='%23f5f5f5' stroke-width='2'
stroke-linecap='round' stroke-linejoin='round'%3E%3Cpolyline points='6 9 12 15 18
9'%3E%3C/polyline%3E%3C/svg%3E");

  /* RGB versions for dark mode */
  --color-success-rgb: var(--color-teal-300-rgb);
  --color-error-rgb: var(--color-red-400-rgb);
  --color-warning-rgb: var(--color-orange-400-rgb);
  --color-info-rgb: var(--color-gray-300-rgb);
}

[data-color-scheme="light"] {
  /* RGB versions for opacity control (light mode) */
  --color-brown-600-rgb: 94, 82, 64;
  --color-teal-500-rgb: 33, 128, 141;
  --color-slate-900-rgb: 19, 52, 59;

  /* Semantic Color Tokens (Light Mode) */
  --color-background: var(--color-cream-50);
  --color-surface: var(--color-cream-100);
  --color-text: var(--color-slate-900);
  --color-text-secondary: var(--color-slate-500);
  --color-primary: var(--color-teal-500);
  --color-primary-hover: var(--color-teal-600);
  --color-primary-active: var(--color-teal-700);
  --color-secondary: rgba(var(--color-brown-600-rgb), 0.12);
  --color-secondary-hover: rgba(var(--color-brown-600-rgb), 0.2);
  --color-secondary-active: rgba(var(--color-brown-600-rgb), 0.25);
  --color-border: rgba(var(--color-brown-600-rgb), 0.2);
  --color-btn-primary-text: var(--color-cream-50);
  --color-card-border: rgba(var(--color-brown-600-rgb), 0.12);
  --color-card-border-inner: rgba(var(--color-brown-600-rgb), 0.12);
  --color-error: var(--color-red-500);
  --color-success: var(--color-teal-500);
  --color-warning: var(--color-orange-500);
  --color-info: var(--color-slate-500);
  --color-focus-ring: rgba(var(--color-teal-500-rgb), 0.4);

  /* RGB versions for light mode */
```

```css
  --color-success-rgb: var(--color-teal-500-rgb);
  --color-error-rgb: var(--color-red-500-rgb);
  --color-warning-rgb: var(--color-orange-500-rgb);
  --color-info-rgb: var(--color-slate-500-rgb);
}

/* Base styles */
html {
  font-size: var(--font-size-base);
  font-family: var(--font-family-base);
  line-height: var(--line-height-normal);
  color: var(--color-text);
  background-color: var(--color-background);
  -webkit-font-smoothing: antialiased;
  box-sizing: border-box;
}

body {
  margin: 0;
  padding: 0;
}

*,
*::before,
*::after {
  box-sizing: inherit;
}

/* Typography */
h1,
h2,
h3,
h4,
h5,
h6 {
  margin: 0;
  font-weight: var(--font-weight-semibold);
  line-height: var(--line-height-tight);
  color: var(--color-text);
  letter-spacing: var(--letter-spacing-tight);
}

h1 {
  font-size: var(--font-size-4xl);
}
h2 {
  font-size: var(--font-size-3xl);
}
h3 {
  font-size: var(--font-size-2xl);
}
h4 {
  font-size: var(--font-size-xl);
}
h5 {
  font-size: var(--font-size-lg);
}
h6 {
  font-size: var(--font-size-md);
}

p {
  margin: 0 0 var(--space-16) 0;
}

a {
  color: var(--color-primary);
  text-decoration: none;
  transition: color var(--duration-fast) var(--ease-standard);
}

a:hover {
  color: var(--color-primary-hover);
}

code,
pre {
  font-family: var(--font-family-mono);
  font-size: calc(var(--font-size-base) * 0.95);
  background-color: var(--color-secondary);
  border-radius: var(--radius-sm);
}

code {
  padding: var(--space-1) var(--space-4);
}

pre {
  padding: var(--space-16);
  margin: var(--space-16) 0;
  overflow: auto;
  border: 1px solid var(--color-border);
}

pre code {
  background: none;
  padding: 0;
}

/* Buttons */
.btn {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  padding: var(--space-8) var(--space-16);
  border-radius: var(--radius-base);
  font-size: var(--font-size-base);
  font-weight: 500;
  line-height: 1.5;
  cursor: pointer;
  transition: all var(--duration-normal) var(--ease-standard);
  border: none;
  text-decoration: none;
  position: relative;
}

.btn:focus-visible {
  outline: none;
  box-shadow: var(--focus-ring);
}

.btn--primary {
  background: var(--color-primary);
  color: var(--color-btn-primary-text);
}

.btn--primary:hover {
  background: var(--color-primary-hover);
}

.btn--primary:active {
  background: var(--color-primary-active);
}

.btn--secondary {
  background: var(--color-secondary);
  color: var(--color-text);
}

.btn--secondary:hover {
  background: var(--color-secondary-hover);
}

.btn--secondary:active {
  background: var(--color-secondary-active);
}

.btn--outline {
  background: transparent;
  border: 1px solid var(--color-border);
  color: var(--color-text);
}

.btn--outline:hover {
  background: var(--color-secondary);
}
```

```css
.btn--sm {
  padding: var(--space-4) var(--space-12);
  font-size: var(--font-size-sm);
  border-radius: var(--radius-sm);
}

.btn--lg {
  padding: var(--space-10) var(--space-20);
  font-size: var(--font-size-lg);
  border-radius: var(--radius-md);
}

.btn--full-width {
  width: 100%;
}

.btn:disabled {
  opacity: 0.5;
  cursor: not-allowed;
}

/* Form elements */
.form-control {
  display: block;
  width: 100%;
  padding: var(--space-8) var(--space-12);
  font-size: var(--font-size-md);
  line-height: 1.5;
  color: var(--color-text);
  background-color: var(--color-surface);
  border: 1px solid var(--color-border);
  border-radius: var(--radius-base);
  transition: border-color var(--duration-fast) var(--ease-standard),
    box-shadow var(--duration-fast) var(--ease-standard);
}

textarea.form-control {
  font-family: var(--font-family-base);
  font-size: var(--font-size-base);
}

select.form-control {
  padding: var(--space-8) var(--space-12);
  -webkit-appearance: none;
  -moz-appearance: none;
  appearance: none;
  background-image: var(--select-caret-light);
  background-repeat: no-repeat;
  background-position: right var(--space-12) center;
  background-size: 16px;
  padding-right: var(--space-32);
}

/* Add a dark mode specific caret */
@media (prefers-color-scheme: dark) {
  select.form-control {
    background-image: var(--select-caret-dark);
  }
}

/* Also handle data-color-scheme */
[data-color-scheme="dark"] select.form-control {
  background-image: var(--select-caret-dark);
}

[data-color-scheme="light"] select.form-control {
  background-image: var(--select-caret-light);
}

.form-control:focus {
  border-color: var(--color-primary);
  outline: var(--focus-outline);
}

.form-label {
  display: block;
  margin-bottom: var(--space-8);
  font-weight: var(--font-weight-medium);
  font-size: var(--font-size-sm);
}

.form-group {
  margin-bottom: var(--space-16);
}

/* Card component */
.card {
  background-color: var(--color-surface);
  border-radius: var(--radius-lg);
  border: 1px solid var(--color-card-border);
  box-shadow: var(--shadow-sm);
  overflow: hidden;
  transition: box-shadow var(--duration-normal) var(--ease-standard);
}

.card:hover {
  box-shadow: var(--shadow-md);
}

.card__body {
  padding: var(--space-16);
}
```

```css
.card__header,
.card__footer {
  padding: var(--space-16);
  border-bottom: 1px solid var(--color-card-border-inner);
}

/* Status indicators - simplified with CSS variables */
.status {
  display: inline-flex;
  align-items: center;
  padding: var(--space-6) var(--space-12);
  border-radius: var(--radius-full);
  font-weight: var(--font-weight-medium);
  font-size: var(--font-size-sm);
}

.status--success {
  background-color: rgba(
    var(--color-success-rgb, 33, 128, 141),
    var(--status-bg-opacity)
  );
  color: var(--color-success);
  border: 1px solid
    rgba(var(--color-success-rgb, 33, 128, 141), var(--status-border-opacity));
}

.status--error {
  background-color: rgba(
    var(--color-error-rgb, 192, 21, 47),
    var(--status-bg-opacity)
  );
  color: var(--color-error);
  border: 1px solid
    rgba(var(--color-error-rgb, 192, 21, 47), var(--status-border-opacity));
}

.status--warning {
  background-color: rgba(
    var(--color-warning-rgb, 168, 75, 47),
    var(--status-bg-opacity)
  );
  color: var(--color-warning);
  border: 1px solid
    rgba(var(--color-warning-rgb, 168, 75, 47), var(--status-border-opacity));
}

.status--info {
  background-color: rgba(
    var(--color-info-rgb, 98, 108, 113),
    var(--status-bg-opacity)
  );
  color: var(--color-info);
  border: 1px solid
    rgba(var(--color-info-rgb, 98, 108, 113), var(--status-border-opacity));
}

/* Container layout */
.container {
  width: 100%;
  margin-right: auto;
  margin-left: auto;
  padding-right: var(--space-16);
  padding-left: var(--space-16);
}

@media (min-width: 640px) {
  .container {
    max-width: var(--container-sm);
  }
}
@media (min-width: 768px) {
  .container {
    max-width: var(--container-md);
  }
}
@media (min-width: 1024px) {
  .container {
    max-width: var(--container-lg);
  }
}
@media (min-width: 1280px) {
  .container {
    max-width: var(--container-xl);
  }
}

/* Utility classes */
.flex {
  display: flex;
}
.flex-col {
  flex-direction: column;
}
.items-center {
  align-items: center;
}
.justify-center {
  justify-content: center;
}
.justify-between {
  justify-content: space-between;
}
.gap-4 {
  gap: var(--space-4);
```

```
}
.gap-8 {
  gap: var(--space-8);
}
.gap-16 {
  gap: var(--space-16);
}

.m-0 {
  margin: 0;
}
.mt-8 {
  margin-top: var(--space-8);
}
.mb-8 {
  margin-bottom: var(--space-8);
}
.mx-8 {
  margin-left: var(--space-8);
  margin-right: var(--space-8);
}
.my-8 {
  margin-top: var(--space-8);
  margin-bottom: var(--space-8);
}

.p-0 {
  padding: 0;
}
.py-8 {
  padding-top: var(--space-8);
  padding-bottom: var(--space-8);
}
.px-8 {
  padding-left: var(--space-8);
  padding-right: var(--space-8);
}
.py-16 {
  padding-top: var(--space-16);
  padding-bottom: var(--space-16);
}
.px-16 {
  padding-left: var(--space-16);
  padding-right: var(--space-16);
}

.block {
  display: block;
}
.hidden {
  display: none;
}
```

```css
/* Accessibility */
.sr-only {
  position: absolute;
  width: 1px;
  height: 1px;
  padding: 0;
  margin: -1px;
  overflow: hidden;
  clip: rect(0, 0, 0, 0);
  white-space: nowrap;
  border-width: 0;
}

:focus-visible {
  outline: var(--focus-outline);
  outline-offset: 2px;
}

/* Dark mode specifics */
[data-color-scheme="dark"] .btn--outline {
  border: 1px solid var(--color-border-secondary);
}

@font-face {
  font-family: 'FKGroteskNeue';
  src: url('https://r2cdn.perplexity.ai/fonts/FKGroteskNeue.woff2')
    format('woff2');
}

/* END PERPLEXITY DESIGN SYSTEM */
/* Custom styles for Bank of AJO portal */
@import url("https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600&display=swap");

#app {
  min-height: 100vh;
  display: flex;
  flex-direction: column;
}

/* Center utility for login screen */
.center-screen {
  flex: 1;
  display: flex;
  align-items: center;
  justify-content: center;
  padding: var(--space-16);
}

.logo {
  max-width: 180px;
}
```

```
/* Header */
.app-header {
  background: var(--color-surface);
  box-shadow: var(--shadow-sm);
}
.app-header .navbar-brand img {
  height: 40px;
}

/* Proposal form sections */
.section-card {
  margin-bottom: var(--space-16);
}

/* Large dashboard buttons */
.dashboard-btn {
  height: 140px;
  font-size: var(--font-size-2xl);
}

/* Table scroll on mobile */
.table-responsive {
  overflow-x: auto;
}
```

`js/app.js`

```
// app.js – Bank of AJO Funder Portal (client-side SPA)
// Communicates with a backend server to persist proposal data.

/*****************
 * CONFIGURATION *
 *****************/
const USERS = {
  User1: "password1",
  User2: "password2",
  User3: "password3",
};

/****************
 * GLOBAL STATE *
 ****************/
const state = {
  authUser: null, // current username string, or null if not authenticated
  proposals: [],  // array of submitted proposal objects
};

async function loadProposals() {
  try {
    const response = await fetch('/api/proposals');
    const proposals = await response.json();
    state.proposals = proposals.map(p => ({...p, data: JSON.parse(p.data)}));
  } catch (error) {
    console.error('Failed to load proposals:', error);
    state.proposals = [];
  }
}

function calculateAge(dob) {
  const birth = new Date(dob);
  const today = new Date();
  let age = today.getFullYear() - birth.getFullYear();
  const m = today.getMonth() - birth.getMonth();
  if (m < 0 || (m === 0 && today.getDate() < birth.getDate())) {
    age--;
  }
  return age;
}

async function saveProposal(proposal) {
  try {
    await fetch('/api/proposals', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(proposal),
    });
  } catch (error) {
    console.error('Failed to save proposal:', error);
  }
}

async function deleteProposal(id) {
  try {
    await fetch(`/api/proposals/${id}`, { method: 'DELETE' });
  } catch (error) {
    console.error('Failed to delete proposal:', error);
  }
}

/*******************
 * UTILITY HELPERS *
 *******************/
const $ = (sel, scope = document) => scope.querySelector(sel);
const createEl = (tag, attrs = {}, html = "") => {
  const el = document.createElement(tag);
  Object.entries(attrs).forEach(([k, v]) => el.setAttribute(k, v));
  el.innerHTML = html;
  return el;
};
function navigate(hash) {
  window.location.hash = hash;
}
```

```
function clearApp() {
  const app = document.getElementById("app");
  app.innerHTML = "";
  return app;
}
function randomStatus() {
  return ["Accepted", "Declined", "Referred"][Math.floor(Math.random() * 3)];
}

/********************
 * ROUTER & NAV GUARD *
 ********************/
window.addEventListener("hashchange", router);
document.addEventListener("DOMContentLoaded", router);
if (document.readyState !== "loading") {
  router();
}

async function router() {
  const hash = window.location.hash || "#login";

  // Guard: redirect to login if not authenticated (except for #login)
  if (!state.authUser && hash !== "#login") {
    return navigate("#login");
  }

  // Guard: if already authenticated, hitting #login should send to dashboard
  if (state.authUser && hash === "#login") {
    return navigate("#dashboard");
  }

  // Load proposals on every view change that requires them
  if (hash === '#submissions' || hash.startsWith('#confirm')) {
    await loadProposals();
  }

  switch (true) {
    case /^#login$/.test(hash):
      return renderLogin();
    case /^#dashboard$/.test(hash):
      return renderDashboard();
    case /^#new$/.test(hash):
      return renderNewProposal();
    case /^#submissions$/.test(hash):
      return renderSubmissions();
    case /^#confirm\/.+/.test(hash):
      const id = hash.split("/")[1];
      return renderConfirmation(id);
    default:
      navigate("#login");
  }
}
```

```
/**********************
 * SHARED UI ELEMENTS *
 **********************/
function renderHeader(root) {
  const header = createEl(
    "nav",
    { class: "navbar navbar-expand-lg app-header px-3 mb-4" },
    `<a href="#dashboard" class="navbar-brand d-flex align-items-center gap-2">
        <img src="https://user-gen-media-assets.s3.amazonaws.com/gpt4o_images/7863599f-31e7-4d7b-9cec-70eeac7a279e.png" alt="Bank of AJO logo" height="40" />
        <span class="fw-semibold">Bank of AJO</span>
      </a>
      <div class="ms-auto d-flex align-items-center gap-3">
        <span class="text-secondary">Welcome, ${state.authUser}</span>
        <button class="btn btn-sm btn-outline-secondary" id="logoutBtn" type="button">Logout</button>
      </div>`
  );
  header.querySelector("#logoutBtn").addEventListener("click", () => {
    state.authUser = null;
    navigate("#login");
  });
  root.appendChild(header);
}

/****************
 * VIEW: LOGIN  *
 ****************/
function renderLogin() {
  const app = clearApp();
  document.title = "Sign in – Bank of AJO";

  const wrapper = createEl("div", { class: "center-screen w-100" });
  wrapper.innerHTML = `
    <div class="card p-4" style="min-width:320px;max-width:400px;width:100%;">
      <div class="text-center mb-4">
        <img src="https://user-gen-media-assets.s3.amazonaws.com/gpt4o_images/7863599f-31e7-4d7b-9cec-70eeac7a279e.png" alt="Bank of AJO logo" class="logo mb-3" />
        <h2 class="h4">Bank of AJO Portal</h2>
      </div>
      <div id="loginAlert" class="alert alert-danger d-none" role="alert">Invalid credentials. Please try again.</div>
      <form id="loginForm" novalidate>
        <div class="form-floating mb-3">
          <input type="text" name="username" class="form-control" id="username" placeholder="User1" aria-label="Username" />
          <label for="username">Username</label>
        </div>
        <div class="form-floating mb-3">
          <input type="password" name="password" class="form-control" id="password" placeholder="••••" aria-label="Password" />
          <label for="password">Password</label>
        </div>
        <button type="submit" class="btn btn-primary w-100">Sign in</button>
      </form>
    </div>`;
  app.appendChild(wrapper);

  const loginForm = $("#loginForm", wrapper);
  loginForm.addEventListener("submit", (e) => {
    e.preventDefault();
    const usernameInput = loginForm.username.value.trim();
    const passwordInput = loginForm.password.value;

    // case-insensitive username check
    const validUserKey = Object.keys(USERS).find(
      (u) => u.toLowerCase() === usernameInput.toLowerCase()
    );

    if (validUserKey && USERS[validUserKey] === passwordInput) {
      state.authUser = validUserKey; // store original casing
      navigate("#dashboard");
    } else {
      $("#loginAlert", wrapper).classList.remove("d-none");
    }
  });
}

/*******************
 * VIEW: DASHBOARD *
 *******************/
function renderDashboard() {
  const app = clearApp();
  document.title = "Dashboard – Bank of AJO";
  renderHeader(app);

  const container = createEl(
    "div",
    { class: "container flex flex-col items-center gap-16 mt-8" },
    `<div class="row w-100 g-4">
      <div class="col-12 col-md-6 d-grid">
        <button class="btn btn-primary dashboard-btn" id="newProposalBtn">New
Proposal</button>
      </div>
      <div class="col-12 col-md-6 d-grid">
        <button class="btn btn-secondary dashboard-btn" id="checkSubmissionsBtn">Check
Submissions</button>
      </div>
    </div>`
  );
  app.appendChild(container);
  $("#newProposalBtn", container).addEventListener("click", () => navigate("#new"));
  $("#checkSubmissionsBtn", container).addEventListener("click", () =>
navigate("#submissions"));
}

/************************
 * VIEW: NEW PROPOSAL    *
 ************************/
function renderNewProposal() {
  const app = clearApp();
  document.title = "New Proposal – Bank of AJO";
  renderHeader(app);

  const container = createEl("div", { class: "container mb-5" });
  container.innerHTML = `
    <h3 class="mb-4">New Credit Proposal</h3>
    <form id="proposalForm">
      <div id="proposalAlert" class="alert alert-danger d-none" role="alert"></div>
      <!-- Applicant Details -->
      <div class="card section-card">
        <div class="card-header fw-semibold">Applicant Details</div>
        <div class="card-body row g-3">
          <div class="col-md-2">
            <label class="form-label">Title</label>
            <select name="title" class="form-select" aria-label="Title">
              <option>Mr</option><option>Mrs</option><option>Miss</option>
<option>Ms</option>
            </select>
          </div>
          <div class="col-md-5">
            <label class="form-label">First Name</label>
            <input type="text" name="firstName" class="form-control" />
          </div>
          <div class="col-md-5">
            <label class="form-label">Last Name</label>
            <input type="text" name="lastName" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">Date of Birth</label>
            <input type="date" name="dob" class="form-control" />
          </div>
        </div>
      </div>

      <!-- Employment Details -->
      <div class="card section-card">
        <div class="card-header fw-semibold">Employment Details</div>
        <div class="card-body row g-3">
          <div class="col-md-6">
            <label class="form-label">Employer Name</label>
            <input type="text" name="employer" class="form-control" />
          </div>
          <div class="col-md-6">
            <label class="form-label">Job Title</label>
            <input type="text" name="jobTitle" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">Employment Start Date</label>
            <input type="date" name="startDate" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">Annual income (£)</label>
            <input type="number" name="income" class="form-control" />
          </div>
        </div>
      </div>

      <!-- Current Address -->
      <div class="card section-card">
        <div class="card-header fw-semibold">Current Address</div>
        <div class="card-body row g-3 align-items-end">
          <div class="col-md-8">
            <label class="form-label">Address Line 1</label>
            <input type="text" name="currAddress1" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">Address Line 2</label>
            <input type="text" name="currAddress2" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">City</label>
            <input type="text" name="currCity" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">Postcode</label>
            <div class="input-group">
              <input type="text" name="currPostcode" class="form-control" />
              <button type="button" id="lookupBtn" class="btn btn-outline-
secondary">Lookup</button>
            </div>
          </div>
          <div class="col-md-4">
            <label class="form-label">Time at Current Address</label>
            <div class="d-flex gap-2">
              <select name="currYears" class="form-select">
                <option value="" selected disabled>Years</option>
                ${Array.from({ length:26 }, (_,i) => i===25
                  ? `<option value="25">25+</option>`
                  : `<option value="${i}">${i}</option>`).join("")}
              </select>
              <select name="currMonths" class="form-select">
                <option value="" selected disabled>Months</option>
                ${Array.from({ length:12 }, (_,i)=>`<option value="${i}">${i}
</option>`).join("")}
```

```
        </select>
      </div>
    </div>
  </div>
</div>

<!-- Previous Address (conditional) -->
<div class="card section-card d-none" id="prevCard">
  <div class="card-header fw-semibold">
    Previous Address
  </div>
  <div class="card-body row g-3">
      <div class="col-md-8">
        <label class="form-label">Address Line 1</label>
        <input type="text" name="prevAddress1" class="form-control" />
      </div>
      <div class="col-md-4">
        <label class="form-label">Address Line 2</label>
        <input type="text" name="prevAddress2" class="form-control" />
      </div>
      <div class="col-md-4">
        <label class="form-label">City</label>
        <input type="text" name="prevCity" class="form-control" />
      </div>
      <div class="col-md-4">
        <label class="form-label">Postcode</label>
        <input type="text" name="prevPostcode" class="form-control" />
      </div>
  </div>
</div>

<!-- Bank Details -->
<div class="card section-card">
  <div class="card-header fw-semibold">Bank Details</div>
  <div class="card-body row g-3">
    <div class="col-md-4">
      <label class="form-label">Sort Code</label>
      <input type="text" name="sortCode" class="form-control" />
    </div>
    <div class="col-md-4">
      <label class="form-label">Account Number</label>
      <input type="text" name="accountNumber" class="form-control" />
    </div>
  </div>
</div>

<!-- Loan Details -->
<div class="card section-card">
  <div class="card-header fw-semibold">Loan Details</div>
  <div class="card-body row g-3">
    <div class="col-md-4">
      <label class="form-label">Loan Amount (£)</label>
```

```
            <input type="number" name="loanAmount" class="form-control" />
          </div>
          <div class="col-md-4">
            <label class="form-label">Term (months)</label>
            <select name="loanTerm" class="form-
select">${[12,24,36,48,60].map(m=>`<option>${m}</option>`).join("")}</select>
          </div>
        </div>
      </div>

      <div class="d-flex justify-content-between align-items-center mb-5">
        <span id="proposalAlert" class="text-danger fw-bold d-none"></span>
        <button type="submit" class="btn btn-primary">Submit Proposal</button>
      </div>
    </form>`;
  app.appendChild(container);

  const yearsSel = $("select[name='currYears']", container);
  const prevCard = $("#prevCard", container);
  function updatePrevVisibility() {
    const years = parseInt(yearsSel.value, 10);
    if (!isNaN(years) && years < 3) {
      prevCard.classList.remove("d-none");
    } else {
      prevCard.classList.add("d-none");
    }
  }
  yearsSel.addEventListener("change", updatePrevVisibility);
  updatePrevVisibility();

  // Postcode lookup button
  $("#lookupBtn", container).addEventListener("click", () => {
    $("input[name='currAddress1']", container).value = "1 Test Street";
    $("input[name='currCity']", container).value = "Testville";
  });

  // Form submit handler
  $("#proposalForm", container).addEventListener("submit", async (e) => {
    e.preventDefault();
    const formData = Object.fromEntries(new FormData(e.target).entries());
    const alertBox = $("#proposalAlert", container);
    if (alertBox) alertBox.classList.add("d-none");

    const acct = (formData.accountNumber || "").trim();
    if (acct && !/^\d{8}$/.test(acct)) {
      if (alertBox) {
        alertBox.textContent = "Account Number must be 8 digits";
        alertBox.classList.remove("d-none");
      }
      return;
    }
    if (formData.dob) {
      const age = calculateAge(formData.dob);
      if (age < 18) {
        if (alertBox) {
          alertBox.textContent = "Too Young for Credit";
          alertBox.classList.remove("d-none");
        }
        return;
      }
    }
    const proposalId = `PID-${Date.now()}`;
    const status = randomStatus();

    // Generate PDF via global jsPDF (loaded in index.html)
    const { jsPDF } = window.jspdf;
    const doc = new jsPDF();
    doc.setFontSize(16);
    doc.text(`Bank of AJO – Proposal ${proposalId}`, 10, 20);
    doc.setFontSize(12);
    doc.text(`Status: ${status}`, 10, 30);
    let y = 40;
    for (const [key, val] of Object.entries(formData)) {
      doc.text(`${key}: ${val}`, 10, y);
      y += 8;
      if (y > 280) { doc.addPage(); y = 20; }
    }
    const pdfUrl = doc.output("datauristring");

    // Store proposal in state
    const proposalObj = {
      id: proposalId,
      applicant: `${formData.firstName || ""} ${formData.lastName || ""}`.trim(),
      loanAmount: formData.loanAmount || "",
      status,
      pdfUrl,
      data: formData,
    };
    await saveProposal(proposalObj);
    state.proposals.push(proposalObj);

    // Simulate webhook (ignore errors)
    fetch("https://webhook.site/mock", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify(proposalObj),
    }).catch(() => {});

    // Navigate to confirmation page
    navigate(`#confirm/${proposalId}`);
  });
}
```

```
/***********************
 * VIEW: CONFIRMATION  *
 ***********************/
function renderConfirmation(proposalId) {
  const proposal = state.proposals.find((p) => p.id === proposalId);
  if (!proposal) return navigate("#dashboard");

  const app = clearApp();
  document.title = "Proposal Confirmation – Bank of AJO";
  renderHeader(app);

  const container = createEl(
    "div",
    { class: "container mt-4" },
    `<div class="alert alert-info">
       <h4 class="alert-heading">Proposal Submitted!</h4>
       <p class="mb-1">Proposal ID: <strong>${proposal.id}</strong></p>
       <p class="mb-3">Status: <strong>${proposal.status}</strong></p>
       <hr class="my-2">
       <p class="mb-0">You can return to Dashboard or view all submissions.</p>
     </div>
     <div class="d-flex gap-3">
       <a href="#dashboard" class="btn btn-primary">Dashboard</a>
       <a href="#submissions" class="btn btn-secondary">Check Submissions</a>
     </div>`
  );
  app.appendChild(container);
}

/********************
 * VIEW: SUBMISSIONS *
 ********************/
function renderSubmissions() {
  const app = clearApp();
  document.title = "Submissions – Bank of AJO";
  renderHeader(app);

  const container = createEl(
    "div",
    { class: "container mt-4" },
    `<h3 class="mb-4">Submitted Proposals</h3>
     <div class="table-responsive">
       <table class="table table-striped" id="subsTable">
         <thead>
           <tr>
            <th>Proposal ID</th><th>Applicant</th><th>Loan Amount (£)</th><th>Status</th>
<th>PDF</th><th></th>
           </tr>
         </thead>
         <tbody></tbody>
       </table>
     </div>`
  );
  app.appendChild(container);

  populateSubmissionsTable();
}
function populateSubmissionsTable() {
  const tbody = $("#subsTable tbody");
  if (!tbody) return;
  tbody.innerHTML = state.proposals
    .map((p) => {
      const badge = p.status === "Accepted" ? "success" : p.status === "Declined" ? "danger" : "warning";
      return `<tr>
        <td>${p.id}</td>
        <td>${p.applicant}</td>
        <td>${p.loanAmount}</td>
        <td><span class="badge bg-${badge}">${p.status}</span></td>
        <td><a href="${p.pdfUrl}" download="${p.id}.pdf" class="btn btn-sm btn-outline-primary">Download</a></td>
        <td><button class="btn btn-sm btn-outline-danger delete-btn" data-id="${p.id}">Delete</button></td>
      </tr>`;
    })
    .join("");

  tbody.querySelectorAll(".delete-btn").forEach((btn) => {
    btn.addEventListener("click", async () => {
      const id = btn.getAttribute("data-id");
      await deleteProposal(id);
      state.proposals = state.proposals.filter((pr) => pr.id !== id);
      populateSubmissionsTable();
    });
  });
}
```

`server.js`

```
const express = require('express');
const sqlite3 = require('sqlite3').verbose();
const path = require('path');

const app = express();
const port = 3000;

// Serve static files from the root directory
app.use(express.static(__dirname));
app.use(express.json());

// Initialize SQLite database
const db = new sqlite3.Database('./database.sqlite', (err) => {
  if (err) {
    console.error(err.message);
  }
  console.log('Connected to the SQLite database.');
});

// Create proposals table if it doesn't exist
db.run(`CREATE TABLE IF NOT EXISTS proposals (
  id TEXT PRIMARY KEY,
  applicant TEXT,
  loanAmount TEXT,
  status TEXT,
  pdfUrl TEXT,
  data TEXT
)`);

// API routes
app.get('/api/proposals', (req, res) => {
  db.all('SELECT * FROM proposals', [], (err, rows) => {
    if (err) {
      res.status(500).json({ error: err.message });
      return;
    }
    res.json(rows);
  });
});

app.post('/api/proposals', (req, res) => {
  const { id, applicant, loanAmount, status, pdfUrl, data } = req.body;
  const stmt = db.prepare('INSERT INTO proposals VALUES (?, ?, ?, ?, ?, ?)');
  stmt.run(id, applicant, loanAmount, status, pdfUrl, JSON.stringify(data), (err) => {
    if (err) {
      res.status(500).json({ error: err.message });
      return;
    }
    res.status(201).json({ message: 'Proposal created' });
  });
  stmt.finalize();
});

app.delete('/api/proposals/:id', (req, res) => {
  const { id } = req.params;
  db.run('DELETE FROM proposals WHERE id = ?', id, (err) => {
    if (err) {
      res.status(500).json({ error: err.message });
      return;
    }
    res.json({ message: 'Proposal deleted' });
  });
});

app.listen(port, () => {
  console.log(`Server listening at http://localhost:${port}`);
});
```

`package.json`

```json
{
  "name": "bank-ajo-portal",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^5.1.0",
    "sqlite3": "^5.1.7"
  }
}
```