Getting Started

Welcome to the presentation of the PollZ project. This project is a comprehensive polling application that allows users to create, manage, and participate in polls. The project is built using a variety of modern frameworks and follows the Onion Architecture to ensure maintainability, scalability, and testability.

Onion Architeture

The project follows the Onion Architecture, which emphasizes the separation of concerns and promotes a clear dependency rule: code can depend on layers inward but not outward. The main layers are:

1. Domain Layer: Contains the core business logic and domain entities. This layer is independent of any other layers.
2. Application Layer: Contains the application services and interfaces. This layer depends on the Domain Layer.
3. Infrastructure Layer: Contains the implementation of interfaces and external dependencies such as databases and APIs. This layer depends on the Application Layer.
4. Presentation Layer: Contains the user interface and API controllers. This layer depends on the Application Layer.

Implementation in PollZ

1. Domain Layer: The domain entities such as User, Poll, Question, and Answer are defined in the Model directory.
2. Application Layer: The application services such as UserService, PollService, QuestionService, and AnswerService are defined in the Service directory.
3. Infrastructure Layer: The data context and database migrations are defined in the Data directory. The implementation of repository interfaces is also part of this layer.
4. Presentation Layer: The API controllers such as UserRouter, PollRouter, QuestionRouter, and AnswerRouter are defined in the Router directory. The frontend components and pages are defined in the Frontend directory.

Database Schema

Here you can check the database schema designed for this project:

Frameworks and Technologies

Backend

1. ASP.NET Core: A cross-platform framework for building modern, cloud-based, internet-connected applications. It is used to build the backend services and APIs.
2. Entity Framework Core: An object-database mapper for .NET. It is used for database operations and migrations.
3. Npgsql: A .NET data provider for PostgreSQL. It is used to connect and interact with the PostgreSQL database.

Frontend

1. React: A JavaScript library for building user interfaces. It is used to build the frontend components and pages.
2. TypeScript: A typed superset of JavaScript that compiles to plain JavaScript. It is used to add static types to the frontend code.
3. Vite: A build tool that provides a faster and leaner development experience for modern web projects. It is used to bundle and serve the frontend code.
4. Tailwind CSS: A utility-first CSS framework for rapidly building custom user interfaces. It is used to style the frontend components.

Other Tools

1. ESLint: A tool for identifying and fixing problems in JavaScript code. It is used to enforce coding standards and best practices.
2. PostCSS: A tool for transforming CSS with JavaScript plugins. It is used to process and optimize CSS files.

Key Features

User Authentication

1. Registration: Users can register by providing their username, email, password, and role.
2. Login: Users can log in using their credentials to access the application.

Poll Management

1. Create Poll: Users can create new polls by providing the poll title and description.
2. View Polls: Users can view a list of all polls.
3. Poll Details: Users can view the details of a specific poll, including the questions and voters.

Question Management

1. Add Question: Users can add questions to a poll.
2. Update Question: Users can update the details of a question.
3. Delete Question: Users can delete a question from a poll.

Voting

1. Cast Vote: Users can create voters that can vote on a specific poll using an unique token.
2. View Results: Users can view the results of the votes for each question.

Project Structure

The project is organized into several directories, each serving a specific purpose:

1. Backend/: Contains the backend services, routers, models, and data transfer objects (DTOs).
2. Frontend/: Contains the frontend components, pages, and services.
3. Data/: Contains database migration scripts and the data context.
4. configs/: Contains configuration files.

PollZ Platform

The PollZ project is a robust and scalable polling application built using modern frameworks and following the Onion Architecture. It ensures a clear separation of concerns, making the codebase maintainable and testable. The use of React, TypeScript, and Tailwind CSS on the frontend provides a responsive and interactive user experience, while ASP.NET Core and Entity Framework Core on the backend ensure reliable and efficient data management.

Thank you for your attention. If you have any questions, feel free to contact.