Project Documentation

1. Project Overview

The **Pet Adoption Portal** is a web application designed to connect animals in need of homes with potential adopters. The platform serves three main user groups: adopters (regular users), shelters/rescue organizations, and administrators. It supports the entire adoption process—from posting listings to managing applications and finalizing adoptions.

1.1 Project Goals

- Create an accessible platform for shelters to post adoptable pets
- Provide tools for potential adopters to search for pets that match their preferences
- Streamline the application process for pet adoption
- Provide shelters with a comprehensive management system to track pets and applications
- Allow administrators to monitor all activities on the platform

1.2 Key Features

For Adopters:

- Browse available pets with filtering options
- View detailed pet profiles with photos and descriptions
- Submit adoption applications
- Track application status
- Manage user profile and adoption history

For Shelters:

- Manage shelter profile information
- Add and update pet listings with images and details
- Review and process adoption applications
- Track adoption statistics

For Administrators:

- Verify shelters
- Monitor all platform activity
- Access detailed statistics
- Manage users, shelters, and pets

---

2. Tech Stack

2.1 Frontend

- **React.js** – JavaScript library for building user interfaces
- **React Router** – Routing management
- **React Bootstrap** – Responsive and modern UI design
- **Axios** – HTTP request library

- **React Context API** – State management
- **React Icons** – Icon integration
- **React Toastify** – Notifications
- **Chart.js & React-Chartjs-2** – Data visualizations
- **Formik & Yup** – Form handling and validation

2.2 Backend

- **Node.js** – Server-side JavaScript runtime
- **Express.js** – Web framework for Node.js
- **MongoDB** – NoSQL database
- **Mongoose** – ODM for MongoDB and Node.js
- **JWT (JSON Web Token)** – Authentication and authorization
- **Bcrypt.js** – Password hashing
- **Express-fileupload** – File upload handling

2.3 Deployment

- **Docker & Docker Compose** – Containerization and management
- **Nginx (optional)** – Reverse proxy server

---

3. Project Architecture

3.1 Project Structure

The project follows a client-server architecture and is divided into two main components:

```
csharp
КопироватьРедактировать
pet-adoption-portal/
├── client/                  # React frontend
│   ├── public/              # Static files
│   └── src/                 # Source code
│       ├── assets/          # Styles and images
│       ├── components/      # React components
│       ├── contexts/        # React Contexts
│       ├── pages/           # Page components
│       ├── services/        # API services
│       └── utils/           # Utility functions
└── server/                  # Express backend
    ├── config/              # Configuration files
    ├── controllers/         # API controllers
    ├── middleware/          # Express middleware
    ├── models/              # Mongoose models
    ├── routes/              # API routes
    └── utils/               # Helper functions
```

3.2 Data Models

User

- Basic info (name, email, password)
- User role (regular user, shelter, admin)

- Contact info and address
- Adoption history

Shelter

- Shelter profile (name, description, logo)
- Contact person info
- List of pets
- Verification status

Pet

- Main details (name, type, breed, age, gender, size)
- Photos and description
- Health and behavior status
- Adoption status (available, pending, adopted)

Adoption

- Pet and adopter information
- Application details (living conditions, pet care experience)
- Application status (pending, approved, rejected, completed)
- Approval details

---

4. Key Feature Implementation

4.1 Authentication & Authorization

Authentication system uses **JWT**:

- User registration with password encryption
- Login with token generation
- Route protection with JWT validation
- Role-based access control

4.2 Pet Management

- Full CRUD operations for pet records
- Uploading and managing pet photos
- Tracking availability status

4.3 Adoption Process

- Submitting detailed adoption applications
- Shelters reviewing and processing applications
- Updating pet status based on application outcome

4.4 Admin Features

- Shelter verification

- Monitoring statistics and platform activity
- Managing users and content moderation

---

5. UI and UX

5.1 Design and Responsiveness

- Modern, clean design using **React Bootstrap**
- Fully responsive interface for mobile, tablet, and desktop
- Consistent design system with CSS variables

5.2 User Flows

- **Pet Search:** Filters by type, breed, age, gender, size
- **Pet View:** Detailed profiles with images and descriptions
- **Adoption Process:** Multi-step forms for comprehensive info
- **Dashboards:** Dedicated dashboards for each user role

---

6. Security

- Password hashing using **bcrypt**
- Protection against common web vulnerabilities (XSS, CSRF)
- Input validation and sanitization
- Role-based access control
- Secure HTTP headers using **Helmet**

---

7. Deployment

The project is containerized using Docker and Docker Compose:

- Separate containers for frontend, backend, and MongoDB
- Configurable environment variables
- Easy to scale and deploy

---

8. Future Improvements

- Integration with payment systems to support shelters
- Advanced search with geolocation
- Recommendation system to match adopters with suitable pets
- Social networking features for adopters' community
- Mobile app for wider access

---

9. Conclusion

The **Pet Adoption Portal** is a comprehensive solution that leverages modern web technologies to build a helpful and user-friendly platform. With its modular architecture and flexible design, the project can be easily extended and adapted to meet the evolving needs of adopters and shelters. The system successfully achieves its core goals by offering an intuitive user interface and reliable server-side infrastructure, supporting the mission of connecting pets with their future families.