**Technical Requirements Document (TRD)**

1. **Architecture and Components**

**Architecture Overview:**

Petopia will be built using a modern, scalable architecture to ensure performance, reliability, and maintainability.

* **Frontend**: The frontend will be developed using React.js, a popular JavaScript library for building user interfaces. React Router will be used for client-side routing.
* **Backend**: The backend will be built with Node.js, providing a lightweight and efficient server environment. Express.js will be used as the web application framework to handle routing and middleware.
* **Database**: MongoDB will be used as the database management system, offering flexibility and scalability for managing pet data, user information, and adoption applications.

**Components:**

* **User Interface**: React.js components for the frontend interface, including pet listings, profiles, and adoption application forms.
* **Server**: Node.js server running Express.js for handling HTTP requests and responses.
* **Database Management**: MongoDB for storing and retrieving pet data, user information, and adoption applications.
* **Authentication**: JSON Web Tokens (JWT) for user authentication and authorization.
* **Messaging System**: Integration with a messaging service (e.g., Gmail or Twilio) for communication between users and adoption agencies.

**2. Infrastructure Requirements**

**Database:**

* MongoDB Deployment: Deploy MongoDB on a separate server or cluster for data storage, ensuring adequate resources and redundancy for reliability.

**Third-Party Services:**

* Messaging Service: Integrate with a third-party messaging service (e.g., Gmail, Twilio) for sending notifications and facilitating communication between users and adoption agencies.

**3. Development Tools, Frameworks, and Libraries**

**Frontend Development:**

* React.js: JavaScript library for building user interfaces.
* React Router: Declarative routing for React applications.
* Axios: Promise-based HTTP client for making API requests.

**Backend Development:**

* Node.js: JavaScript runtime for server-side development.
* Express.js: Web application framework for Node.js.
* Mongoose: MongoDB object modeling for Node.js.
* Version Control: Git for version control, with GitHub or GitLab as the repository hosting platform.

**4. APIs and Data Formats**

* **Authentication API**: Implement JWT-based authentication for securing API endpoints and managing user sessions.