Frontend Application

Frontend application for the Physician Personality Trait Annotation System, built with React and TypeScript.

Technology Stack

- **Framework**: React 18
- **Language**: TypeScript
- **UI Library**: Ant Design
- **State Management**: React Hooks
- **Router**: React Router
- **HTTP Client**: Axios
- **Styling**: CSS + Ant Design Theme

Project Structure

```
frontend/
├── public/                 # Static assets
│   ├── index.html      # HTML template
│   └── favicon.ico     # Website icon
├── src/                # Source code
│   ├── components/     # Reusable components
│   │   ├── Login.tsx          # Login component
│   │   ├── PhysicianInfo.tsx   # Physician info display
│   │   ├── ReviewsList.tsx     # Reviews list
│   │   ├── TraitTabs.tsx       # Trait tabs
│   │   ├── TraitWorkflow.tsx   # Trait evaluation workflow
│   │   ├── HumanAnnotationForm.tsx    # Human annotation form
│   │   ├── MachineEvaluationForm.tsx  # Machine evaluation form
│   │   └── ReviewAndModifyForm.tsx    # Review and modify form
│   ├── pages/          # Page components
│   │   ├── HomePage.tsx        # Home page
│   │   └── TaskPage.tsx        # Task page
│   ├── services/       # API services
│   │   └── api.ts      # API interface encapsulation
│   ├── types/          # TypeScript type definitions
│   │   └── index.ts    # Data types
│   ├── App.tsx         # Root component
│   ├── App.css         # Global styles
│   └── index.tsx       # Application entry
├── package.json        # Project dependencies
└── tsconfig.json       # TypeScript configuration
```

Main Features

1. User Login

- Username input
- NPI number and task ID validation
- Session management

2. Physician Information Display

- Basic information display
- **HTML Content Rendering**: Supports HTML tag display in physician biographies
- **Education Background Parsing**: Automatically parses and formats `<education>` tags
- Responsive layout design

3. Patient Review Browsing

- Review list display
- Review metadata parsing
- Scroll browsing functionality

4. Personality Trait Annotation

- Big Five personality trait assessment (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism)
- Score rating and consistency evaluation
- Evidence text input

5. Machine Learning Model Evaluation

- Multi-model results display
- Model ranking functionality
- Evaluation feedback collection

Quick Start

Prerequisites

- Node.js 14+
- npm or yarn

1. Install Dependencies

```
cd frontend
npm install
```

2. Environment Configuration

Create `.env` file (optional):

```
REACT_APP_API_URL=http://localhost:8080/api
```

3. Start Development Server

```
npm start
```

The application will start at http://localhost:3000.

The page will automatically reload when you edit files.

Available Scripts

npm start

Starts the development mode at http://localhost:3000.

npm test

Launches the test runner.

npm run build

Builds the production version to the `build` folder.

npm run eject

⚠️ **This is a one-way operation that cannot be undone!**

Key Features

HTML Content Rendering

- Uses `dangerouslySetInnerHTML` to safely render physician biographies
- Custom CSS styles to beautify HTML content
- Supports paragraph, bold, link, and other HTML tags

Education Background Parsing

- Regular expression parsing of `<education>` tags
- List format display of educational experiences
- Card-style layout for enhanced readability

Responsive Design

- Two-column layout (reviews + annotation interface)
- Adaptive to screen sizes
- Mobile-friendly design

User Experience Optimization

- Loading state indicators
- Error handling and friendly prompts
- Progress tracking and completion notifications
- Data persistence (sessionStorage)

Component Description

Core Components

- **Login**: User login and authentication
- **PhysicianInfo**: Physician information display with HTML rendering support
- **ReviewsList**: Patient review list display
- **TraitTabs**: Big Five personality trait tab navigation
- **TraitWorkflow**: Single trait evaluation workflow

Form Components

- **HumanAnnotationForm**: Human annotation form
- **MachineEvaluationForm**: Machine learning model evaluation form
- **ReviewAndModifyForm**: Annotation review and modification form

API Integration

The frontend communicates with the backend API via Axios:

```
// Get physician information
const physician = await getPhysicianByNPI(npi);

// Submit human annotation
await submitTraitHumanAnnotation(npi, taskId, trait, annotation);

// Get machine annotations
const machineAnnotations = await getTraitMachineAnnotations(npi, taskId, trait);
```

Style Customization

Ant Design Theme

The project uses the Ant Design component library. You can customize the UI by modifying theme colors:

```
/* App.css */
.physician-biography {
  line-height: 1.6;
```

```
  }

  .physician-biography p {
    margin-bottom: 12px;
    text-align: justify;
  }
```

Custom Styles

- Physician biography HTML content styles
- Review list custom scrollbar
- Card shadows and spacing optimization

Deployment

Build Production Version

```
npm run build
```

Build files will be output to the `build/` directory.

Deployment Options

1. **Static File Server**: Deploy build directory contents to any static file server
2. **Nginx**: Configure Nginx reverse proxy
3. **Vercel/Netlify**: One-click deployment to cloud platforms
4. **Docker**: Containerized deployment

Contributing Guidelines

1. Follow project code standards
2. Use TypeScript type safety
3. Components should be pure function components
4. Use Ant Design components for consistency
5. Add appropriate error handling

License

This project is licensed under the MIT License.