AI-BUDDY Universal React Frontend Structure

📁 Project Directory Structure

```
ai-buddy-frontend/
├── public/
│   ├── index.html
│   ├── favicon.ico
│   ├── manifest.json
│   └── assets/
│       ├── images/
│       ├── sounds/
│       └── animations/
├── src/
│   ├── main.tsx                # App entry point
│   ├── App.tsx                 # Root component
│   ├── vite-env.d.ts           # Vite type definitions
│   │
│   ├── api/                    # Backend API Integration Layer
│   │   ├── client.ts           # Axios client configuration
│   │   ├── auth.ts             # Authentication APIs
│   │   ├── users.ts            # User management APIs
│   │   ├── students.ts         # Student-specific APIs
│   │   ├── teachers.ts         # Teacher-specific APIs
│   │   ├── parents.ts          # Parent-specific APIs
│   │   ├── content.ts          # Curriculum & chapters
│   │   ├── ai-buddy.ts         # AI Buddy personality system
│   │   ├── assessments.ts      # Quiz, chat, challenges
│   │   ├── learning-sessions.ts  # 20-question sessions
│   │   ├── progress.ts         # Analytics & reporting
│   │   ├── games.ts            # Educational games
│   │   └── admin.ts            # Admin panel APIs
│   │
│   ├── types/                  # TypeScript Type Definitions
│   │   ├── api/                # API response types
│   │   │   ├── auth.ts
│   │   │   ├── user.ts
│   │   │   ├── student.ts
│   │   │   ├── ai-buddy.ts
│   │   │   ├── content.ts
│   │   │   ├── assessment.ts
│   │   │   ├── progress.ts
```

```
│   │   │   └── common.ts
│   │   ├── database/           # Database model types
│   │   │   ├── user-management.ts
│   │   │   ├── ai-buddy-evolution.ts
│   │   │   ├── content-hierarchy.ts
│   │   │   ├── learning-progress.ts
│   │   │   └── assessment-content.ts
│   │   ├── ui/                 # UI component types
│   │   │   ├── components.ts
│   │   │   ├── forms.ts
│   │   │   └── navigation.ts
│   │   └── index.ts            # Type exports
│   │
│   ├── store/                  # State Management (Redux Toolkit)
│   │   ├── index.ts            # Store configuration
│   │   ├── slices/             # Redux slices
│   │   │   ├── authSlice.ts    # Authentication state
│   │   │   ├── userSlice.ts    # User profile state
│   │   │   ├── studentSlice.ts # Student data state
│   │   │   ├── aiBuddySlice.ts # AI Buddy personality state
│   │   │   ├── contentSlice.ts # Curriculum & chapters
│   │   │   ├── progressSlice.ts # Learning progress state
│   │   │   ├── gameSlice.ts    # Game state management
│   │   │   ├── chatSlice.ts    # Chat session state
│   │   │   └── uiSlice.ts      # UI state (modals, notifications)
│   │   ├── middleware/         # Redux middleware
│   │   │   ├── apiMiddleware.ts # API error handling
│   │   │   └── persistMiddleware.ts # State persistence
│   │   └── hooks/              # Custom Redux hooks
│   │       ├── useAppDispatch.ts
│   │       ├── useAppSelector.ts
│   │       └── useAuth.ts
│   │
│   ├── components/             # Reusable UI Components
│   │   ├── common/             # Universal components
│   │   │   ├── Layout/
│   │   │   │   ├── Header.tsx
│   │   │   │   ├── Footer.tsx
│   │   │   │   ├── Sidebar.tsx
│   │   │   │   └── Navigation.tsx
│   │   │   ├── UI/             # Base UI components
│   │   │   │   ├── Button.tsx
│   │   │   │   ├── Input.tsx
│   │   │   │   ├── Modal.tsx
```

```
│   │   │   ├── Card.tsx
│   │   │   ├── Loading.tsx
│   │   │   ├── ErrorBoundary.tsx
│   │   │   └── ProgressBar.tsx
│   │   ├── Forms/           # Form components
│   │   │   ├── FormField.tsx
│   │   │   ├── FormValidation.tsx
│   │   │   └── FormLayout.tsx
│   │   └── Charts/          # Analytics charts
│   │       ├── ProgressChart.tsx
│   │       ├── SkillChart.tsx
│   │       └── AnalyticsChart.tsx
│   │
│   ├── auth/                # Authentication components
│   │   ├── LoginForm.tsx
│   │   ├── RegisterForm.tsx
│   │   ├── PasswordReset.tsx
│   │   ├── FirstTimeSetup.tsx
│   │   └── ProtectedRoute.tsx
│   │
│   ├── ai-buddy/            # AI Buddy System Components
│   │   ├── BuddyAvatar.tsx    # Visual buddy representation
│   │   ├── BuddyChat.tsx      # Chat interface
│   │   ├── BuddyEvolution.tsx # Evolution progress display
│   │   ├── PersonalitySettings.tsx # Personality customization
│   │   ├── VisualComponents.tsx # Component restoration view
│   │   └── BuddyStats.tsx     # Buddy statistics
│   │
│   ├── learning/            # Learning & Assessment Components
│   │   ├── ChapterContent.tsx # Chapter reading interface
│   │   ├── ChatSession.tsx    # 20-question chat system
│   │   ├── QuizInterface.tsx  # Quiz taking interface
│   │   ├── ChallengeGame.tsx  # Game challenges
│   │   ├── ProgressTracker.tsx # Progress visualization
│   │   └── SkillDevelopment.tsx # Skill tracking
│   │
│   ├── games/               # Educational Games
│   │   ├── PromptQuest/       # Prompt engineering game
│   │   ├── LabelItTrainIt/    # Computer vision game
│   │   ├── SmartEyesVisionBot/ # Vision recognition game
│   │   ├── TextTrainer/       # NLP training game
│   │   └── BuildAIBuddy/      # Buddy building game
│   │
│   └── dashboard/           # Dashboard Components
```

```
│   │   ├── StudentDashboard.tsx
│   │   ├── TeacherDashboard.tsx
│   │   ├── ParentDashboard.tsx
│   │   ├── AdminDashboard.tsx
│   │   └── DashboardWidgets.tsx
│   │
│   ├── pages/                  # Route-based Page Components
│   │   ├── auth/               # Authentication pages
│   │   │   ├── LoginPage.tsx
│   │   │   ├── RegisterPage.tsx
│   │   │   └── ForgotPasswordPage.tsx
│   │   │
│   │   ├── student/            # Student-specific pages
│   │   │   ├── StudentHome.tsx
│   │   │   ├── LearningPath.tsx
│   │   │   ├── AIBuddyPage.tsx
│   │   │   ├── ProgressPage.tsx
│   │   │   └── games/
│   │   │       ├── GameHub.tsx
│   │   │       ├── PromptQuestPage.tsx
│   │   │       ├── LabelItTrainItPage.tsx
│   │   │       ├── SmartEyesPage.tsx
│   │   │       ├── TextTrainerPage.tsx
│   │   │       └── BuildBuddyPage.tsx
│   │   │
│   │   ├── teacher/            # Teacher-specific pages
│   │   │   ├── TeacherHome.tsx
│   │   │   ├── ClassManagement.tsx
│   │   │   ├── CurriculumBuilder.tsx
│   │   │   ├── StudentAnalytics.tsx
│   │   │   └── AssessmentTools.tsx
│   │   │
│   │   ├── parent/             # Parent-specific pages
│   │   │   ├── ParentHome.tsx
│   │   │   ├── ChildProgress.tsx
│   │   │   ├── ReportsPage.tsx
│   │   │   └── SettingsPage.tsx
│   │   │
│   │   ├── admin/              # Admin panel pages
│   │   │   ├── AdminHome.tsx
│   │   │   ├── UserManagement.tsx
│   │   │   ├── SystemAnalytics.tsx
│   │   │   ├── ContentModeration.tsx
│   │   │   └── SystemSettings.tsx
```

```
│   │   │
│   │   └── public/              # Public pages
│   │       ├── HomePage.tsx
│   │       ├── AboutPage.tsx
│   │       ├── FeaturesPage.tsx
│   │       └── ContactPage.tsx
│   │
│   ├── hooks/                   # Custom React Hooks
│   │   ├── useApi.ts            # Generic API hook
│   │   ├── useAuth.ts           # Authentication hook
│   │   ├── useStudent.ts        # Student data hook
│   │   ├── useAIBuddy.ts        # AI Buddy hook
│   │   ├── useProgress.ts       # Progress tracking hook
│   │   ├── useChatSession.ts    # Chat session hook
│   │   ├── useGameState.ts      # Game state hook
│   │   ├── useWebSocket.ts      # Real-time updates hook
│   │   ├── useLocalStorage.ts   # Local storage hook
│   │   ├── useDebounce.ts       # Debouncing hook
│   │   └── useFormValidation.ts # Form validation hook
│   │
│   ├── services/                # Business Logic Services
│   │   ├── apiService.ts        # API service wrapper
│   │   ├── authService.ts       # Authentication service
│   │   ├── studentService.ts    # Student operations
│   │   ├── aiBuddyService.ts    # AI Buddy operations
│   │   ├── contentService.ts    # Content management
│   │   ├── progressService.ts   # Progress tracking
│   │   ├── gameService.ts       # Game logic
│   │   ├── chatService.ts       # Chat operations
│   │   ├── notificationService.ts # Notifications
│   │   ├── storageService.ts    # Local storage operations
│   │   └── validationService.ts # Form validations
│   │
│   ├── utils/                   # Utility Functions
│   │   ├── constants.ts         # App constants
│   │   ├── helpers.ts           # Helper functions
│   │   ├── formatters.ts        # Data formatters
│   │   ├── validators.ts        # Validation functions
│   │   ├── encryption.ts        # Client-side encryption
│   │   ├── dateUtils.ts         # Date utilities
│   │   ├── mathUtils.ts         # Math calculations
│   │   ├── gameUtils.ts         # Game-specific utilities
│   │   └── debugUtils.ts        # Development utilities
│   │
```

```
│   ├── styles/                # Styling
│   │   ├── globals.css        # Global styles
│   │   ├── variables.css      # CSS variables
│   │   ├── components.css     # Component-specific styles
│   │   ├── themes.css         # Theme configurations
│   │   ├── animations.css     # CSS animations
│   │   ├── responsive.css     # Responsive design
│   │   └── games.css          # Game-specific styles
│   │
│   ├── config/                # Configuration Files
│   │   ├── api.ts             # API configuration
│   │   ├── routes.ts          # Route definitions
│   │   ├── theme.ts           # Theme configuration
│   │   ├── constants.ts       # App-wide constants
│   │   └── environment.ts     # Environment variables
│   │
│   └── tests/                 # Test Files
│       ├── components/        # Component tests
│       ├── hooks/             # Hook tests
│       ├── services/          # Service tests
│       ├── utils/             # Utility tests
│       ├── pages/             # Page tests
│       └── __mocks__/         # Mock files
│
├── .env.development           # Development environment
├── .env.production            # Production environment
├── .gitignore                 # Git ignore rules
├── package.json               # Dependencies
├── tsconfig.json              # TypeScript config
├── vite.config.ts             # Vite configuration
├── tailwind.config.js         # Tailwind CSS config
├── postcss.config.js          # PostCSS config
└── README.md                  # Project documentation
```

🔧 Key Architecture Principles

1. Database-Centric Type System

- TypeScript types directly mirror your PostgreSQL schema
- Enum types match database ENUMs exactly
- Relationship types reflect foreign key constraints

2. Role-Based Architecture

- Separate page hierarchies for each user role (Student, Teacher, Parent, Admin)
- Role-specific components and hooks
- Conditional routing based on user permissions

3. AI-First Design

- Dedicated AI Buddy component ecosystem
- Real-time chat integration
- Evolution tracking and visualization

4. Learning-Centric Structure

- Assessment-focused component organization
- Game-based learning integration
- Progress tracking at every level

5. Production-Ready Patterns

- Error boundaries and loading states
- API error handling middleware
- Performance optimization hooks
- Accessibility considerations

📋 Next Steps

1. **Install Dependencies**: Set up the base React + TypeScript + Vite project
2. **Configure API Client**: Align with your FastAPI backend endpoints
3. **Implement Authentication**: JWT token management system
4. **Build Core Components**: Start with layout and common UI components
5. **Develop Role-Specific Features**: Student dashboard → Teacher tools → Parent portal → Admin panel

This structure is specifically designed to:

- **Scale with your database complexity**
- **Support your multi-role system**
- **Integrate seamlessly with your AI Buddy evolution system**
- **Handle your assessment and gaming requirements**
- **Maintain production-grade code organization**