🏆 FitDuel - Transform Fitness into Epic Duels

       <h3>🎮 Gamified Fitness Platform for Teenagers</h3> Challenge friends • Level up • Become champion! Live Demo • Features • Tech Stack • Get Started

🚀 Overview

FitDuel transforms fitness into an engaging gaming experience where users compete in real-time exercise duels, earn XP, unlock achievements, and climb leaderboards. Built with cutting-edge AI validation and zero video storage for maximum privacy and scalability.

🎯 Key Features

- ⚔️ **Real-time Duels** - Challenge friends to push-up, squat, and plank battles
- 🤖 **AI Validation** - MediaPipe pose detection ensures proper form
- 📱 **Motion Tracking** - Accelerometer validation prevents cheating
- 🏅 **Gamification** - XP system, levels, badges, and leaderboards
- 🔒 **Zero Video Storage** - Privacy-first approach with real-time validation only
- ⚡ **Lightning Fast** - Serverless architecture scales automatically

🛠️ Tech Stack

Frontend

- **Framework**: Next.js 14 (App Router)
- **Language**: TypeScript
- **Styling**: TailwindCSS + Framer Motion
- **State**: Zustand
- **UI Components**: Custom components with Lucide icons

Backend

- **Database**: Supabase (PostgreSQL)
- **Auth**: Supabase Auth
- **Storage**: Supabase Storage (thumbnails only)

- **API**: Next.js API Routes

AI & Validation

- **Pose Detection**: MediaPipe (TensorFlow.js)
- **Motion Tracking**: Device Motion API
- **Anti-cheat**: Multi-layer validation system

Infrastructure

- **Hosting**: Vercel
- **CDN**: Vercel Edge Network
- **Monitoring**: Sentry
- **Analytics**: PostHog (optional)

📁 Project Structure

```
fitduel-app/
├── src/
│   ├── app/                # Next.js App Router pages
│   │   ├── (auth)/         # Authentication pages
│   │   ├── (dashboard)/    # Protected pages
│   │   ├── api/            # API routes
│   │   └── globals.css     # Global styles
│   ├── components/         # React components
│   │   ├── ui/             # Base UI components
│   │   ├── game/           # Game-specific components
│   │   └── providers/      # Context providers
│   ├── lib/                # Core libraries
│   │   ├── validation/     # AI & motion validation
│   │   └── supabase/       # Database client
│   ├── hooks/              # Custom React hooks
│   ├── utils/              # Helper functions
│   ├── config/             # Configuration files
│   └── types/              # TypeScript types
├── public/                 # Static assets
├── database/               # SQL schemas
└── docs/                   # Documentation
```

🚀 Getting Started

Prerequisites

- Node.js 18+

- npm or yarn
- Supabase account
- Vercel account (for deployment)

Installation

1. **Clone the repository**

```bash
git clone https://github.com/yourusername/fitduel-app.git
cd fitduel-app
```

2. **Install dependencies**

```bash
npm install
```

3. **Set up environment variables**

```bash
cp .env.example .env.local
```

Edit `.env.local` with your credentials:

```env
NEXT_PUBLIC_SUPABASE_URL=your-supabase-url
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-key
SUPABASE_SERVICE_ROLE_KEY=your-service-key
```

4. **Set up the database**

Go to your Supabase dashboard and run the SQL schema from `database/schema.sql`

5. **Run the development server**

```bash
npm run dev
```

Open http://localhost:3000

🎮 Features

Core Gameplay

- **Exercise Duels**: Real-time 1v1 challenges
- **AI Form Validation**: Ensures proper exercise technique
- **Motion Detection**: Validates real movement via accelerometer
- **XP & Levels**: Progress system with 12 levels
- **Badges**: Unlock achievements for milestones
- **Leaderboards**: Global and friend rankings

Exercises Supported

- 💪 Push-ups
- 🦵 Squats
- 🏋️ Planks
- 🔥 Burpees
- ⭐ Jumping Jacks
- 🏔️ Mountain Climbers

Anti-Cheat System

- **Multi-layer Validation**: AI + Motion + Pattern analysis
- **Device Fingerprinting**: Prevents multiple accounts
- **Shadowban System**: Automatic detection of cheaters
- **Cryptographic Proofs**: Tamper-proof result validation

🔒 Privacy & Security

- **Zero Video Storage**: Videos are processed in real-time and never stored
- **Local AI Processing**: Pose detection runs on user's device
- **Encrypted Data**: All sensitive data is encrypted
- **GDPR Compliant**: Full compliance with privacy regulations
- **Minimal Data Collection**: Only essential metrics are stored

📊 Database Schema

Key tables:

- `users` - User profiles and stats
- `challenges` - Duel records
- `challenge_results` - Exercise results

- `user_badges` - Unlocked achievements
- `leaderboards` - Ranking data
- `friendships` - Social connections

See `database/schema.sql` for complete schema.

🚀 Deployment

Deploy to Vercel

1. Push your code to GitHub
2. Import project in Vercel
3. Add environment variables
4. Deploy!

Mostra immagine

Environment Variables

Required for production:

```
NEXT_PUBLIC_SUPABASE_URL
NEXT_PUBLIC_SUPABASE_ANON_KEY
SUPABASE_SERVICE_ROLE_KEY
NEXT_PUBLIC_APP_URL
```

Optional:

```
SENTRY_DSN
NEXT_PUBLIC_POSTHOG_KEY
```

📈 Scaling Strategy

Performance Optimizations

- **Edge Functions**: API routes run at edge locations
- **ISR**: Incremental Static Regeneration for leaderboards
- **Image Optimization**: Next.js automatic image optimization
- **Code Splitting**: Automatic code splitting per route

Infrastructure Scaling

- **Auto-scaling**: Vercel automatically scales with traffic
- **Database Pooling**: Supabase connection pooling
- **CDN**: Static assets served from edge
- **Rate Limiting**: Configurable per-user limits

🧪 Testing

```bash
# Run tests
npm test

# Type checking
npm run type-check

# Linting
npm run lint
```

🤝 Contributing

1. Fork the repository
2. Create your feature branch (`git checkout -b feature/amazing-feature`)
3. Commit changes (`git commit -m 'Add amazing feature'`)
4. Push to branch (`git push origin feature/amazing-feature`)
5. Open a Pull Request

Development Guidelines

- Write TypeScript, not JavaScript
- Follow the existing code style
- Add tests for new features
- Update documentation
- Keep components under 300 lines
- Use meaningful commit messages

📝 API Documentation

Core Endpoints

```typescript
```

```
POST   /api/duels/create      - Create new duel
POST   /api/duels/accept      - Accept duel challenge
POST   /api/duels/complete    - Submit duel results
GET    /api/duels/list        - List user's duels
GET    /api/leaderboard       - Get leaderboard
POST   /api/users/profile     - Update profile
GET    /api/users/stats       - Get user statistics
```

🔧 Configuration

Infrastructure Config

See `src/config/infrastructure.ts` for:

- Rate limiting settings
- Anti-cheat thresholds
- Caching strategies
- Feature flags

Exercise Config

See `src/utils/constants.ts` for:

- XP rewards
- Level requirements
- Exercise parameters
- Badge criteria

📱 Mobile Support

- **PWA Ready**: Installable as mobile app
- **Responsive Design**: Optimized for all screen sizes
- **Touch Optimized**: Gesture-friendly interface
- **Motion Sensors**: Full accelerometer support

🎯 Roadmap

Phase 1 (Current)

- ✅ Core duel system
- ✅ AI validation
- ✅ Basic gamification
- ✅ User profiles

Phase 2 (Q2 2024)

- [ ] Team battles
- [ ] Weekly tournaments
- [ ] Social features
- [ ] Premium subscriptions

Phase 3 (Q3 2024)

- [ ] Mobile apps (iOS/Android)
- [ ] Live streaming duels
- [ ] AI coaching
- [ ] Wearable integration

Phase 4 (Q4 2024)

- [ ] School partnerships
- [ ] Sponsored challenges
- [ ] Virtual rewards
- [ ] Global events

🐛 Known Issues

- Motion sensors not available on desktop
- iOS requires permission for motion access
- Some Android devices have limited accelerometer accuracy

📄 License

🙏 Acknowledgments

- MediaPipe for pose detection
- Supabase for backend infrastructure
- Vercel for hosting
- TailwindCSS for styling

💬 Support

- **Email**: support@fit-duel.com
- **Discord**: Join our community
- **Issues**: GitHub Issues

🌟 Star History

 Built with 💪 by the FitDuel Team Ready to duel? Start now!