# Cosmic Dressing — 3D Outfit Builder

> **A Next.js application for creating personalized 3D outfits with real-time preview on customizable avatars.**


## Live Demo

**Project is partially deployed on Vercel:**
https://v0-fork-of-image-analysis-brown-nine.vercel.app/

---

## Table of Contents



---

## Features

- **3D Avatar Customization** — Adjust height, build, skin tone, and colors
- **Virtual Try-On** — Preview outfits on your custom avatar in real-time
- **Interactive 3D Scene** — Rotate, zoom, and explore with intuitive controls
- **Save & Share** — Store your favorite outfit combinations
- **Secure Authentication** — Email-based login with Supabase Auth
- **Responsive Design** — Works seamlessly on desktop, tablet, and mobile

---

## Vision & Roadmap

### Our Ultimate Goal

**Transform online fashion shopping into an immersive, personalized 3D experience where customers can:**

1. **Create their digital twin** with accurate body measurements
2. **Virtually try on clothes** with realistic fabric simulation and physics
3. **See how outfits look** on *their* body type before purchasing
4. **Shop confidently** with AR try-on and AI-powered style recommendations
5. **Reduce returns** by 60%+ through accurate size prediction and visualization

> **Vision:** Become the leading platform for hyper-personalized virtual fashion experiences, eliminating the guesswork from online clothing purchases and reducing fashion waste through informed buying decisions.

---

## 🚧 Current Limitations & What We're Missing

| Area | Current State | Gap |
|---|---|---|
| **Avatar Realism** | Basic 3D model with simple customization | Need photorealistic avatars with accurate body scanning |
| **Clothing Physics** | Static clothing placement | Require real-time fabric simulation (draping, stretching) |
| **Product Catalog** | Sample products in database | Need integration with real e-commerce inventory APIs |
| **AI Features** | Not implemented | Missing AI style recommendations, size prediction |
| **AR Try-On** | Desktop 3D only | No mobile AR (ARKit/ARCore) support yet |
| **Measurements** | Manual height/build selection | Need automatic body measurement from photos/camera |
| **Checkout** | Basic cart functionality | Missing payment gateway integration (Stripe/PayPal) |
| **Social Features** | Not implemented | No outfit sharing, community feed, or social shopping |

---

## 🛠️ Planned Updates & Improvements

**Phase 1: Enhanced Realism (Next 3 months)**

- ☐ Integrate **Ready Player Me** or **MetaHuman** for photorealistic avatars
- ☐ Implement **cloth simulation** using Three.js physics libraries
- ☐ Add **texture mapping** for realistic fabric materials (cotton, denim, silk)
- ☐ Improve **lighting & shadows** for better product visualization

**Phase 2: AI-Powered Features (3-6 months)**

- ☐ **AI Size Recommendation** — ML model to predict best fit based on body measurements
- ☐ **Style Assistant** — GPT-powered chatbot for outfit suggestions
- ☐ **Virtual Fitting Room** — Upload photo → Auto-generate 3D avatar
- ☐ **Trend Analysis** — AI-curated outfit collections based on fashion trends

**Phase 3: Mobile & AR (6-9 months)**

- ☐ **Mobile App** (React Native) with camera-based body scanning

- ☐ **AR Try-On** — iOS/Android AR overlay for real-time preview
- ☐ **Social Shopping** — Share outfits, get feedback from friends
- ☐ **Live Try-On Sessions** — Video calls with virtual styling

**Phase 4: E-Commerce Integration (9-12 months)**

- ☐ **Payment Gateway** — Stripe/Razorpay/PayPal integration
- ☐ **Inventory Sync** — Real-time product availability from partner brands
- ☐ **Order Management** — Full checkout, shipping, and return flow
- ☐ **Brand Partnerships** — Onboard fashion retailers and designers

---

## 🔧 Required Tech Stack for Full Vision

| Feature | Current Stack | Additional Requirements |
|---|---|---|
| **Photorealistic Avatars** | Basic Three.js models | **Ready Player Me API**, **MetaHuman**, or **Custom ML models** |
| **Cloth Simulation** | Static meshes | **Oimo.js** or **Cannon.js** (physics), **Three.js Cloth** |
| **Body Scanning** | Manual input | **TensorFlow.js** (pose estimation), **MediaPipe** (body landmarks) |
| **AI Recommendations** | None | **OpenAI GPT-4**, **TensorFlow** (size prediction model) |
| **Mobile AR** | Web only | **React Native**, **ARKit** (iOS), **ARCore** (Android), **AR.js** |
| **Payment Processing** | None | **Stripe**, **Razorpay**, **PayPal SDK** |
| **Real-time Collaboration** | None | **WebRTC**, **Socket.io** (live sessions) |
| **Analytics** | None | **Mixpanel**, **Amplitude** (user behavior tracking) |
| **Image Processing** | Basic uploads | **Cloudinary** or **AWS S3** (CDN), **Sharp.js** (optimization) |
| **Search & Discovery** | None | **Algolia** or **Elasticsearch** (product search) |

## 🚀 How to Achieve Our Goals

**1. Immediate Next Steps (You Can Help!)**

```
✅ Improve 3D model loading (better GLTF models with animations)
✅ Add more customization options (accessories, shoes, bags)
✅ Implement outfit combination saving/loading
✅ Build social sharing features
✅ Create comprehensive product database schema
```

**2. Developer Contributions Needed**

- **3D Artists** — Create realistic clothing models with proper UV mapping
- **ML Engineers** — Build size prediction and body measurement models
- **Mobile Developers** — React Native app with AR capabilities
- **Backend Engineers** — Scalable API for inventory and user management
- **UI/UX Designers** — Intuitive interfaces for complex 3D interactions

**3. Technical Challenges to Solve**

- **Performance** — Render complex 3D scenes smoothly on low-end devices
- **Accuracy** — Body measurements from 2D photos/video
- **Scalability** — Handle thousands of concurrent 3D rendering sessions
- **Realism** — Fabric behavior, lighting, and fit accuracy

**4. Business Model (Future)**

- **B2C:** Commission on sales (10-15% per transaction)
- **B2B:** SaaS for fashion brands (white-label virtual fitting rooms)
- **Freemium:** Basic features free, premium for AI styling and AR
- **Data Licensing:** Anonymized fit data for brands to improve sizing

---

### 💡 Why This Matters

**The Problem:**

- 📄 **30-40% return rates** in online fashion due to poor fit
- 💸 **$550B+ lost annually** in returns and unsold inventory
- 🌍 **Fashion waste crisis** from unnecessary production and returns
- 😞 **Customer frustration** with sizing inconsistency across brands

**Our Solution:**

- ✅ **"Try before you buy"** virtually — see realistic fit on your body
- ✅ **Accurate sizing** — AI predicts your size across brands
- ✅ **Personalized shopping** — Discover styles that actually suit you
- ✅ **Sustainability** — Buy only what fits, reduce waste

---

## 🚀 Quick Start

### Prerequisites

Before you begin, make sure you have:

- **Node.js** 18 or higher (Download)
- **pnpm** or **npm** (pnpm recommended: `npm install -g pnpm`)
- **Supabase account** (Sign up free)
- **Git** (Download)

### Get Running in 5 Minutes

```
# 1. Clone the repository
git clone https://github.com/vivek-kumar-P/cosmic-dressing.git
cd cosmic-dressing-main

# 2. Install dependencies
pnpm install
# or: npm install

# 3. Set up environment variables (see below)
# Create .env.local and add your Supabase keys

# 4. Run development server
pnpm dev
# or: npm run dev

# 5. Open your browser
# Visit http://localhost:3000
```

---

## 🔧 Installation

### Step 1: Get Supabase Credentials

1. Go to supabase.com and create a new project
2. Wait for project initialization (~2 minutes)
3. Navigate to **Settings** → **API**
4. Copy these values:
   - **Project URL** (looks like `https://xxx.supabase.co`)
   - **anon/public key** (starts with `eyJ...`)
   - **service_role key** (starts with `eyJ...`, keep this secret!)

### Step 2: Create Environment File

Create a file named `.env.local` in your project root:

```
NEXT_PUBLIC_SUPABASE_URL=your_project_url_here
NEXT_PUBLIC_SUPABASE_ANON_KEY=your_anon_key_here
SUPABASE_SERVICE_ROLE_KEY=your_service_role_key_here
```

⚠ **Important:** Never commit `.env.local` to Git. It's already in `.gitignore`.

### Step 3: Set Up Database

Run these SQL scripts in your **Supabase SQL Editor** (in order):

1. `scripts/00-complete-database-setup.sql` — Creates all tables
2. `scripts/02-create-rls-policies.sql` — Sets up security policies
3. `scripts/09-create-storage-bucket.sql` — Creates storage for uploads

4. `scripts/14-add-onboarding-field.sql` — Adds user onboarding
5. `scripts/15-optimize-rls-policies.sql` — Optimizes security

**Quick tip:** Open each file, copy all content, paste into Supabase SQL Editor, and click "Run".

### Step 4: Configure Supabase Authentication

In your Supabase Dashboard:

1. Go to **Authentication** → **URL Configuration**
2. Set **Site URL**: `http://localhost:3000`
3. Add **Redirect URL**: `http://localhost:3000/auth/callback`

### Step 5: Start Development Server

```
pnpm dev
```

Visit **http://localhost:3000** — you should see the landing page! 🎉

---

## 🐛 Common Issues & Solutions

### ❌ "Module not found" or dependency errors

```
# Delete node_modules and reinstall
rm -rf node_modules pnpm-lock.yaml
pnpm install
# or: rm -rf node_modules package-lock.json && npm install
```

### ❌ Supabase connection errors

**Problem:** `Error: Invalid Supabase URL` or connection timeouts

**Solutions:**

1. Double-check `.env.local` has correct values (no extra spaces)
2. Verify Supabase project is active (not paused)
3. Test connection at `http://localhost:3000/test-connection`
4. Restart dev server after changing `.env.local`

### ❌ Database / RLS policy errors

**Problem:** `new row violates row-level security policy` or `permission denied`

**Solutions:**

1. Ensure all SQL scripts ran successfully (check Supabase logs)
2. Verify RLS policies exist: **Database** → **Policies** in Supabase

3. Re-run `scripts/07-fix-rls-security.sql` and `scripts/15-optimize-rls-policies.sql`

### ❌ 3D model not loading / black screen

**Problem:** 3D canvas appears but no avatar shows

**Solutions:**

1. Open browser DevTools (F12) → Console tab
2. Look for errors related to GLTFLoader or model paths
3. Check Network tab: verify `/models/human/scene.gltf` returns 200 OK
4. Ensure WebGL is supported: visit https://get.webgl.org/
5. Try a different browser (Chrome recommended)

### ❌ "Port 3000 is already in use"

```
# Find and kill the process using port 3000
Get-Process -Id (Get-NetTCPConnection -LocalPort 3000).OwningProcess | Stop-
Process -Force

# Or run on a different port
$env:PORT='3001'; pnpm dev
```

### ❌ Build/TypeScript errors

```
# Clear Next.js cache and rebuild
rm -rf .next
pnpm build
```

**Note:** The project has some TypeScript warnings in non-critical files. These won't prevent the app from running.

### ❌ Image upload fails

**Problem:** Profile picture or avatar images won't upload

**Solutions:**

1. Run `scripts/09-create-storage-bucket.sql` to create storage bucket
2. Check file size (max 2MB recommended)
3. Verify Supabase Storage is enabled in your project
4. Check storage policies: **Storage** → **Policies** in Supabase

---

## 💻 Tech Stack

## 💻 Tech Stack

| Category | Technology |
|---|---|
| **Framework** | Next.js 14 (App Router) |
| **Language** | TypeScript |
| **3D Graphics** | Three.js with GLTFLoader |
| **Styling** | Tailwind CSS + shadcn/ui |
| **Database** | Supabase (PostgreSQL) |
| **Authentication** | Supabase Auth |
| **State Management** | React Context API |
| **Package Manager** | pnpm (or npm) |

## 🗀 Project Structure

```
cosmic-dressing-main/
├── app/                      # Next.js App Router pages
│   ├── 3d-playground/        # 3D customization studio
│   ├── auth/                 # Login, register, callbacks
│   ├── dashboard/            # User dashboard
│   ├── gallery/              # Product gallery
│   ├── cart/                 # Shopping cart
│   └── api/                  # API routes
├── components/               # Reusable React components
│   ├── 3d/                   # 3D rendering components
│   ├── ui/                   # UI components (shadcn)
│   ├── auth/                 # Auth-related components
│   └── layout/               # Layout components
├── lib/                      # Utility functions
│   ├── supabase.ts           # Supabase client
│   └── utils.ts              # Helper functions
├── public/                   # Static assets
│   └── models/               # 3D model files (.gltf)
├── scripts/                  # Database SQL scripts
├── types/                    # TypeScript type definitions
├── .env.local               # Environment variables (create this!)
└── package.json             # Dependencies
```

## 🚢 Deployment

### Deploy to Vercel (Recommended)

1. **Push to GitHub** (if not already done)

```
git add .
git commit -m "Ready for deployment"
git push origin main
```

2. **Connect to Vercel**
   - Go to vercel.com
   - Click "New Project"
   - Import your GitHub repository
   - Vercel auto-detects Next.js settings
3. **Add Environment Variables**
   - In Vercel project settings, go to **Environment Variables**
   - Add all variables from your `.env.local`:

```
NEXT_PUBLIC_SUPABASE_URL
NEXT_PUBLIC_SUPABASE_ANON_KEY
SUPABASE_SERVICE_ROLE_KEY
```

4. **Update Supabase Settings**
   - In Supabase Dashboard → **Authentication** → **URL Configuration**
   - Update **Site URL** to your Vercel domain (e.g., `https://your-app.vercel.app`)
   - Add redirect URL: `https://your-app.vercel.app/auth/callback`
5. **Deploy**
   - Click "Deploy" in Vercel
   - Wait ~2 minutes for build to complete
   - Visit your live app! 🎉

### Manual Deployment

```
# Build for production
pnpm build

# Start production server
pnpm start
```

---

## 📚 Additional Resources

### Testing Your Setup

Visit these URLs to verify everything works:

- 🏠 **Homepage:** http://localhost:3000
- 🔐 **Login:** http://localhost:3000/auth/login
- 🎮 **3D Playground:** http://localhost:3000/3d-playground
- 🩺 **Health Check:** http://localhost:3000/api/health
- 🔧 **Connection Test:** http://localhost:3000/test-connection

## Quick Commands

```
# Development
pnpm dev                # Start dev server
pnpm build              # Build for production
pnpm start              # Start production server
pnpm lint               # Run ESLint

# Troubleshooting
rm -rf .next            # Clear Next.js cache
rm -rf node_modules     # Remove dependencies
pnpm install            # Reinstall dependencies
```

## Database Scripts Reference

Located in `scripts/` folder:

| Script | Purpose |
|---|---|
| `00-complete-database-setup.sql` | Creates all tables and initial schema |
| `02-create-rls-policies.sql` | Sets up Row Level Security |
| `07-fix-rls-security.sql` | Fixes RLS permission issues |
| `09-create-storage-bucket.sql` | Creates storage for uploads |
| `14-add-onboarding-field.sql` | Adds user onboarding fields |
| `15-optimize-rls-policies.sql` | Optimizes security policies |

## VS Code Recommended Extensions

- **ES7+ React/Redux/React-Native snippets** — Code snippets
- **Tailwind CSS IntelliSense** — CSS autocomplete
- **Prettier** — Code formatting
- **ESLint** — Linting

## Getting Help

**If you're stuck:**

1. Check the Common Issues section above
2. Look at browser console (F12 → Console) for errors
3. Verify `.env.local` has correct Supabase credentials

4. Ensure all database scripts ran successfully in Supabase
5. Try the test URLs to isolate the problem

**Still having issues?**

- Check existing GitHub Issues
- Create a new issue with:
  - What you were trying to do
  - Exact error message
  - Your Node.js version (`node --version`)
  - Browser and OS

---

## 🎯 What's Next?

After getting the app running:

1. **Customize your avatar** in the 3D Playground
2. **Browse the gallery** and try different outfits
3. **Save your favorite combinations**
4. **Explore the codebase** — check `components/3d/` for 3D logic

---

## 📄 Documentation Archive

This README consolidates content from these original docs (now archived with pointer headers):

- ARCHITECTURE.pdf — System architecture and design
- ENVIRONMENT_SETUP.pdf — Environment configuration
- DEPLOYMENT.pdf — Detailed deployment guide
- DEPLOYMENT_GUIDE.pdf — Step-by-step deployment
- DEPLOYMENT_CHECKLIST.pdf — Pre-deployment checklist
- SUPABASE_DEPLOYMENT_GUIDE.pdf — Supabase-specific setup

**Note:** When viewing this as a PDF, ensure all PDF files are in the same folder for links to work properly.

---

## 🤝 Contributing

Contributions are welcome! Please:

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

---

## 📝 License

This project is licensed under the MIT License.

---

**Built with ♡ using Next.js, Three.js, and Supabase**

*Happy coding! If you create something cool, we'd love to see it!* 🚀