TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – TAGUIG
Km. 14 East Service Road, Western Bicutan, Taguig City

RenderHaus: Design Your Dream Home Inside and Out

In Partial Fulfillment of the Requirements for the Subject

IT ELECTIVE 4

Submitted by:

Fabila, Crislhan V.

Nagallo, John Leonard O.

BSIT-S-T-4A-T

Submitted to:

Prof. Maracris M. Lappay

Technological University of the Philippines

Taguig City

October 2025

How to set-up:

PREREQUISITES:

- Node.js (version 14 or higher) - Download from https://nodejs.org/
- Python (version 3.8 or higher) - Download from https://python.org/
- Git - Download from https://git-scm.com/

INSTALLATION STEPS:

1. Clone or download the project repository
2. Install Node.js dependencies for backend:
   - Go to backend directory: cd backend
   - Run: npm install

3. Install Node.js dependencies for frontend:
   - Go to frontend directory: cd frontend
   - Run: npm install

4. Install Python dependencies for AI/ML backend:
   - Go to backend/python_backend directory: cd backend/python_backend
   - Recommended: Create a virtual environment: python -m venv venv
   - Activate virtual environment:

    * Windows: venv¥Scripts¥activate

    * macOS/Linux: source venv/bin/activate

  - Install Python packages: pip install -r requirements.txt

5. Create .env file for both frontend and backend folder. Copy/paste the .env file below.

6. To run the system:

6.1. Backend (Node.js): Go to backend directory then run: npm start

6.2. Frontend (React): Go to frontend directory then run: npm start

6.3. Python Backend (AI/ML): Go to backend/python_backend directory then run: python app.py

REQUIRED LIBRARIES:

Backend (Node.js) Dependencies:

- axios: HTTP requests

- bcryptjs: Password hashing

- cloudinary: Cloud storage for 3D models and images (replaces Uploadcare)

- cors: Cross-origin resource sharing

- crypto-js: Cryptographic functions

- dotenv: Environment variables

- express: Web application framework

- express-validator: Input validation

- jsonwebtoken: JWT authentication

- mongoose: MongoDB object modeling

- multer: File upload handling (multipart/form-data)

- nodemailer: Email sending (OTP verification)

- uuid: Unique identifier generation

- winston: Logging library

Frontend (React) Dependencies:

- @heroicons/react: Icon library

- @react-three/drei: Three.js helpers for React

- @react-three/fiber: React renderer for Three.js

- chart.js & react-chartjs-2: Chart visualization

- framer-motion: Animation library

- html2canvas: Screenshot generation

- jspdf: PDF generation

- react: Core React library

- react-dom: React DOM rendering

- react-router-dom: Routing

- react-scroll: Smooth scrolling

- react-toastify: Toast notifications

- three: 3D graphics library

Python Backend Dependencies:

- flask: Web framework

- pymongo: MongoDB driver

- python-dotenv: Environment variables

- scikit-learn: Machine learning

- numpy: Numerical computing

- Pillow: Image processing

- pygltflib: GLTF file handling

- trimesh: 3D mesh processing

- pyrender: 3D rendering

- requests: HTTP requests

- meshio: Mesh I/O operations

All dependencies will be automatically installed when running:

- npm install (for Node.js packages)

- pip install -r requirements.txt (for Python packages)

Files for .env for backend and frontend:

.env for backend: --------------------------

# MongoDB connection URI

```
MONGODB_URI=mongodb+srv://cocomeme2003:gYMBph9Y7p0DYnK2@cluster0.ssdil.mongodb.net/renderhaus?retryWrites=true&w=majority&appName=Cluster0



# Node.js server port

PORT=5000



# Python backend server port

PYTHON_PORT=5001



# JWT Configuration

JWT_SECRET=12345

JWT_ACCESS_EXPIRY=15m

JWT_REFRESH_EXPIRY=7d



# CORS Configuration

CORS_ORIGIN=http://localhost:3000



# Cloudinary Configuration (for 3D model uploads)

CLOUDINARY_CLOUD_NAME=dqo2p0voh

CLOUDINARY_API_KEY=751995628489729

CLOUDINARY_API_SECRET=mRr_Jq6jMOhlOHi9hmF1ZxOiSJ0



# Mailtrap Configuration (for email testing)
```

```
MAILTRAP_USER=ad370c1730db1c

MAILTRAP_PASS=33145d3248c642


# Email Configuration

EMAIL_FROM=noreply@navibuild.com

EMAIL_FROM_NAME=NaviBuild
```

SYSTEM FEATURES:

1. USER FEATURES:

   - User registration with email verification (OTP)
   - JWT-based authentication (access & refresh tokens)
   - Browse 3D furniture models by category
   - Drag and drop furniture into 3D canvas
   - Edit object properties:
     * Position (X, Y, Z axes with sliders)
     * Scale/Size (0.1x to 3x)
     * Rotation (0° to 360°)
     * Color (manual picker + AI recommendations)
   - AI Color Suggestions (object-specific, context-aware)

- AI Furniture Suggestions (room-level recommendations)
- Save and load multiple design projects
- Export designs to PDF with AI suggestions
- Room templates and pre-designed layouts

2. ADMIN FEATURES:

- Admin dashboard with analytics
- Upload new 3D models (Cloudinary storage)
- Manage existing models (edit/delete)
- User management
- Project management
- Model categories: furniture-seating, furniture-tables, furniture-storage, furniture-beds, lighting, decoration, kitchen

3. AI CAPABILITIES:

- Object-specific color recommendations (8 furniture types × 3 styles = 24 schemes)
- Style detection (modern, traditional, rustic)
- Missing furniture identification
- Database-driven model recommendations
- Room color palette suggestions

4. TECHNICAL HIGHLIGHTS:

- 3D rendering with Three.js and WebGL
- GLTF/GLB model format support (up to 10MB on Cloudinary free tier)
- Real-time 3D transformations
- Cloudinary CDN for fast model loading
- MongoDB Atlas cloud database
- Microservices architecture (Node.js + Python backends)

.env for frontend: ---------------------------

```
GENERATE_SOURCEMAP=false
```
