Aesthetica (Interior design web App)

By

Urvisha Baldha

Shruti Chavda

Rutika Vaghasiya

23SOECE13001, 6th Sem, CEA

23SOECE13002, 6thSem, CEA

22SOECE11062, 6th Sem, CEA

Department of Engineering

RK University, Rajkot - Gujarat

March-2025

Table of Contents

1. Introduction

1.1 Purpose

Aesthetica is an interior design web app that helps users plan and design rooms in an easy and fun way. It gives tools to:

- See how a room will look before making changes
- Get design ideas based on user style
- Choose from a list of furniture and decoration items
- Save and share design projects

It is built using **ASP.NET Core** for the backend and **React** for the frontend, making it fast, modern, and easy to use.

More key purposes:

- To help users design their dream space without needing special software
- To make interior designing simple for both beginners and professionals
- To bring designers and clients together in one platform

1.2 Document Conventions

This document follows these writing rules:

- **Font**: Times New Roman
- **Font Size**: 12pt for normal text, 14pt bold for main headings
- **Headings**: Numbered (e.g., 1.1, 1.2, etc.) for easy reading
- **Language**: Easy and simple English for everyone to understand
- **Diagrams and Lists**: Used wherever needed to make content clearer

Extra Notes:

- Technical terms are explained in the glossary at the end
- Important words or terms are written in **bold**
- Bullets (•) are used for lists to make them easier to read

1.3 Intended Audience and Reading Suggestions

This document is useful for many people who are part of the project. It helps them understand what the system is, how it works, and what to expect.

People who should read this:

- **Developers**: To build the application correctly
- **UI/UX Designers**: To design the interface and user journey
- **Testers**: To test the features and check if they work
- **Project Managers**: To plan and manage the project
- **Clients and Stakeholders**: To understand the product and its features
- **Support Team**: To help users and fix issues
- **Maintenance Team**: To update and manage the system
- **Security Team**: To ensure user data and system safety

Tips for reading:

- Read from start to end for full understanding
- Use section numbers to find topics quickly
- Use the glossary if you don't know a term

1.4 Project Scope

Aesthetica is designed to make interior design easier and more fun. The project focuses on creating a tool that is helpful for everyone—from someone designing their bedroom to a professional working on a hotel lobby.

Main features and goals:

- Let users **visualize rooms** with furniture, colors, and layout
- Offer a **catalog** of furniture and decor items
- Allow users to **save**, **edit**, and **share** their design projects
- Provide **AI suggestions** to improve room design
- Work well on **all devices**, including phones, tablets, and laptops
- Use **secure login** and a fast backend for a smooth experience

Extra project goals:

- Build a scalable system that can grow in the future
- Support multiple user types (homeowners, designers, businesses)
- Allow integration with other services like payment or furniture store APIs
- Make the app easy to update and maintain by the development team

2. Overall Description

2.1 Product Perspective

Aesthetica is a modern web application made for the **interior design industry**. It helps people design rooms online without needing special software. It fits in as a **digital tool** for home design and decor.

Extra points:

- Works as a one-stop solution for users who want to **visualize home interiors**
- Can be used along with online furniture stores or design service platforms
- Designed with **modular components** so new features can be added easily
- Provides an experience similar to physical showrooms, but online

2.2 Product Features

Aesthetica offers many tools to help users plan their dream space.

Core features include:

- **Room Visualization**: Build room layouts and see changes live on the screen
- **Furniture & Decor Catalog**: Choose furniture, wall colors, decor items from a list
- **Project Management**: Save your work, come back to edit, or share it with others
- **AI Design Suggestions**: Smart ideas based on your style and room type

Additional features:

- **Drag-and-drop design tools** for easy placement of furniture
- **User dashboard** with saved projects, recent activity, and favorites
- **Style selector** to help pick a design theme (e.g., modern, vintage, minimalist)
- **Budget planner** to keep design ideas within your price range
- **PDF export** to print or email a room plan

2.3 User Classes and Characteristics

Different people will use Aesthetica, each with different needs.

User types:

- **Homeowners & Enthusiasts**: Everyday users who want to design their homes
  - May not be tech-savvy, so the app should be easy to use

- **Interior Designers & Architects**: Experts who need professional tools
  - Will look for precision, export tools, and sharing features
- **Furniture & Decor Businesses**: Shops and sellers who want to showcase items
  - Want to display products in mock rooms and link to purchases

More user types:

- **Project Managers**: Oversee work for interior projects
- **Support Teams**: Help users and maintain system performance
- **Admins**: Manage users, content, and technical settings

2.4 Operating Environment

Aesthetica will run on common web technologies and modern hosting services.

Hardware:

- **Web Servers**: Host the website and application files securely
- **Database Servers**: Store all user information and design data

Software:

- **Backend**: ASP.NET Core (MVC framework)
- **Frontend**: React, HTML5, CSS3, Bootstrap
- **Databases**:
  - **MySQL**: For user accounts, projects, structured content
  - **MongoDB**: For catalog items, design settings, flexible data
- **Operating System**: Can run on **Windows** or **Linux** web servers

Supported Browsers:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari

Supported Devices:

- Desktop computers
- Laptops
- Tablets
- Smartphones

2.5 Design and Implementation Constraints

Some rules and restrictions apply while building the system.

Key constraints:

- Must use **ASP.NET Core** for all backend processing
- **MySQL** is used for saving structured records
- **MongoDB** handles flexible data used in design elements
- The app should support **responsive design** for all screen sizes
- Follow **security standards** for user authentication and data protection

Extra constraints:

- Maintain clean separation between frontend and backend logic
- Support for **REST APIs** to allow third-party integrations
- Should be compatible with **cloud hosting platforms** like Azure or AWS

2.6 User Documentation

Helpful guides will be made for different types of users.

Available documents:

- **User Manual**: Step-by-step guide for regular users to use the app
- **Admin and Support Guide**: Help for technical teams and administrators
- **System Setup Document**: Instructions for setting up and deploying the system
- **FAQs and Troubleshooting Tips**
- **Video tutorials or walkthroughs**

2.7 Assumptions and Dependencies

This section lists things that must be true for the system to work properly.

Key assumptions:

- Users have a working device and stable internet connection
- Web servers and databases are up and running
- Admins will update the furniture and decor catalog regularly

Dependencies:

- Third-party services (like **login APIs**, **payment gateways**) must be available

- Browser support must continue for key web technologies (e.g., JavaScript, CSS Grid)
- Hosting services provide proper security, uptime, and backups

3. System Features

3.1 Room Visualization

Aesthetica helps users design their space easily with interactive tools.

Features:

- **Drag-and-drop interface** to place furniture and decor items in the room
- **Real-time preview**: Changes appear instantly as users adjust layouts
- **Multiple room types** supported (bedroom, kitchen, living room, etc.)
- **Wall and floor customization**: Change colors, textures, and patterns
- **Room dimension input**: Users can set exact room size for better planning
- **Furniture auto-fit option**: Items automatically adjust to available space
- **View from different angles**: Switch between top, 3D, and front views
- **Undo/Redo functionality** to fix mistakes quickly
- **Zoom in/out and pan** tools for better visibility

3.2 Furniture and Decor Catalog

The app includes a smart catalog of products users can explore and add.

Features:

- **Search bar with suggestions** to find items easily
- **Filter by style, price, color, brand, material**
- **Product preview** in the design workspace before adding
- **Item descriptions** include price, dimensions, material, availability
- **Favorites or wishlist** feature to save liked products
- **Link to purchase**: Direct user to online store to buy the item
- **Product rating and reviews** (optional future integration)
- **Dynamic inventory** updates using admin backend tools

3.3 Project Management

Users can manage multiple designs and track their work.

Features:

- **Save projects** to user profiles for future access
- **Edit saved projects** any time and from any device
- **Create duplicate copies** to try different versions of the same idea
- **Project sharing** via direct links, email, or social media platforms
- **Track history** of edits and compare changes
- **Add notes or comments** to designs (useful for collaboration)
- **Download/export projects** as image, PDF, or 3D model
- **Tag projects** for easy search (e.g., "office design", "kitchen redo")

3.4 Design Suggestions and AI Recommendations

Smart tools help users improve designs and find what they need.

Features:

- **AI analyzes room layout** and suggests furniture arrangements
- **Style matching**: Suggest decor items that match the selected theme
- **Auto-fill suggestions**: Complete unfinished designs using trends
- **Lighting and spacing tips** for better interior layout
- **Color palette recommendations** that work well together
- **Trending styles section** based on latest interior design trends
- **AI chatbot assistant** to answer questions and offer design tips (optional add-on)
- **User preference learning**: Suggestions improve over time based on behavior

4. External Interface Requirements

4.1 User Interfaces

The system will provide a clean and easy-to-use design for different user roles.

Features:

- **Responsive design**: Works smoothly on desktops, tablets, and mobile devices
- **Login and registration pages**: Simple forms with validation and error messages
- **Dashboard**: Users can view, open, edit, and manage all their saved projects
- **Drag-and-drop room editor**: Lets users easily place and arrange furniture
- **Admin panel**: For system administrators to manage content, users, and analytics
- **Tooltips and help sections**: Guide users through each feature
- **Profile management**: Users can update their personal details and preferences
- **Theme options**: Light and dark mode for better user comfort

4.2 Hardware Interfaces

The app will use standard hardware setups that are common and widely supported.

Features:

- **Web browser access**: Users access the system through browsers like Chrome, Firefox, Safari
- **Client-side devices**: Any internet-enabled device (PC, tablet, phone)
- **Server setup**: Hosted on secure web servers that can handle high traffic
- **Database servers**:
  - **MySQL** for storing structured data like users and projects
  - **MongoDB** for storing flexible data like catalog items or layouts

4.3 Software Interfaces

Aesthetica connects different software components to work together seamlessly.

Features:

- **RESTful APIs**: Used for frontend-backend communication (data fetch, update, delete)
- **JWT-based Authentication**: Secure login using JSON Web Tokens
- **OAuth integration**: For social logins like Google or Facebook (optional)
- **Frontend libraries**: React, Bootstrap for modern UI
- **Backend framework**: ASP.NET Core MVC for fast and secure processing
- **Database drivers**:
  - MySQL driver (like `MySqlConnector`) for .NET
  - MongoDB driver (like `MongoDB.Driver`) for React integration

4.4 Communications Interfaces

The app uses standard web communication methods to keep everything in sync.

Features:

- **HTTP/HTTPS protocols**: All data is sent securely using encryption
- **WebSocket support (optional)**: For real-time features like collaborative editing
- **Email service integration**: For account verification, password reset, notifications
- **Push notifications**: (optional) For updates and alerts within the app
- **Third-party API integrations**: For payment gateways, design suggestions, or external catalogs

5. Other Nonfunctional Requirements

5.1 Performance Requirements

These ensure the app works quickly and smoothly for all users:

- The system should respond to user actions like saving projects or adding items **within 2 seconds**.
- It must support **at least 500 users** using the platform at the same time without slowing down.
- Room visualization images should **load in under 3 seconds** on a regular internet connection.
- Searching the furniture catalog should **return results within 1.5 seconds**.
- The system should be available and running **99.9% of the time each month**.
- System should **scale automatically** during traffic spikes to maintain performance.
- Background tasks (e.g., AI recommendations) should complete **within 5 seconds** for each request.

5.2 Safety Requirements

These protect user data and system stability:

- All user data like saved projects and profile information must be **backed up daily** on secure cloud storage.
- If a server crash happens, the system should **restore from backup within 30 minutes**.
- Important actions like deleting a project must **ask for confirmation** to avoid mistakes.
- The app must **block duplicate form submissions** due to double clicks or page refreshes.
- In case of system overload, the app should display a **friendly maintenance page** with retry options.
- **Error logs** should be stored securely and reviewed regularly to avoid repeating failures.

5.3 Security Requirements

These keep the system and user information safe:

- All personal user data should be **encrypted using AES-256** or a similar encryption method.
- Passwords must be **securely hashed with bcrypt** or an equivalent algorithm.
- The app must use **JWT for login sessions** and **role-based access control** for different users (admin, designer, regular user).
- All communications must go through **HTTPS** to prevent hacking or spying.
- Users should be **automatically logged out** after **30 minutes of no activity**.

- Only users with admin rights should be able to **access the admin dashboard** or sensitive data.
- Failed login attempts should be **limited** (e.g., 5 tries) and **trigger a cooldown** or CAPTCHA.
- All security updates and patches must be **applied regularly** to servers and libraries.

6. Other Requirements

- The application should **support multiple languages** for future versions to allow localization for different countries.
- Aesthetica must follow **WCAG 2.1** standards to ensure accessibility for users with disabilities.
- All pages and components should use **mobile-first responsive design** so they work well on phones, tablets, and desktops.
- The system must **log user activities**, like logins, actions, and errors, to help with analytics and debugging.
- Integration with third-party services like **payment gateways and AI tools** must be **modular, reusable, and well-documented**.
- There should be a **feedback form** where users can report issues or suggest new features.
- Analytics dashboards must allow users to **export reports in formats like PDF and Excel**.
- The UI should provide **helpful tooltips**, onboarding steps, and guidance for first-time users.
- The system should include **dark mode/light mode toggle** for better user comfort.
- Scheduled system maintenance should be communicated to users **in advance with alerts**.
- The app should support **progressive web app (PWA)** features for offline support and quick launch.
- Error messages and alerts should be **clear, helpful, and user-friendly**.
- The system should be built to allow **future plugin/module extensions** (e.g., adding AR/VR capabilities later).
- All forms should include **client-side and server-side validation** for reliability.

Appendix A: Glossary

Term	Definition
UI/UX	User Interface/User Experience – visual layout and interaction design.
JWT	JSON Web Token – a secure method to pass data between frontend and backend.
ASP.NET Core	A backend framework used to build secure, scalable web applications.
MongoDB	A NoSQL database used to store flexible, non-relational data.

Term	Definition
MySQL	A relational database used to store structured data like users and projects.
Catalog	A digital collection of furniture and decor items users can browse and use.
Room Visualization	A feature allowing users to design and view custom room layouts.
OAuth	A secure protocol for allowing third-party access without sharing passwords.
WCAG	Web Content Accessibility Guidelines – rules for making websites accessible.
Responsive Design	A design that adjusts to all screen sizes, from mobile to desktop.
PWA	Progressive Web App – a web app that works offline and behaves like a native app.

Appendix B: Analysis Models

Use Case Diagrams

- **User Registration & Login:** Shows how users sign up, log in, and manage their sessions.
- **Project Creation and Saving:** Visualizes how users start, save, and manage design projects.
- **Catalog Browsing and Selection:** Explains how users explore items and add them to designs.
- **AI Design Suggestions:** Describes how the system recommends items based on user preferences.

Entity-Relationship Diagrams (ERDs)

- **Users ↔ Projects ↔ Items:** Illustrates how users relate to saved projects and catalog items.
- **Users ↔ Preferences ↔ Suggestions:** Maps how preferences influence design suggestions.

Data Flow Diagrams (DFDs)

- **Design Save Flow:** Data path from user edits to saved room layouts in the database.
- **AI Recommendation Flow:** How user input is processed into smart design suggestions.
- **Room Visualization Update Flow:** Steps from user interaction to updated visual display.

State Transition Diagrams

- **User Authentication States:** States such as Login → Active → Idle → Logout.
- **Project Lifecycle:** States including Draft → Saved → Shared → Archived.