

```

portfolio/
├── public/
│   ├── assets/
│   │   ├── images/
│   │   │   ├── profile/
│   │   │   ├── projects/
│   │   │   ├── skills/
│   │   │   └── logos/
│   │   ├── icons/
│   │   └── fonts/
│   └── favicon.ico
├── src/
│   ├── app/
│   │   ├── layout.js
│   │   ├── page.js
│   │   └── globals.css
│   ├── about/
│   │   └── page.js
│   ├── projects/
│   │   ├── page.js
│   │   └── layout.js
└── pages
    ├── [slug]/
    │   ├── page.js
    │   ├── ProjectDetailTemplate.jsx
    │   ├── ProjectHeader.jsx
    │   ├── ProjectGallery.jsx
    │   ├── ProjectMeta.jsx
    │   └── index.js
    └── components/
used in project pages
    ├── ProjectList.jsx
    └── index.js

    ├── skills/
    │   └── page.js

    ├── contact/
    │   └── page.js

    └── api/
        └── contact/route.js

    ├── components/
    │   ├── atoms/
    │   │   ├── buttons/
    │   │   │   ├── Button.jsx
    │   │   │   └── IconButton.jsx

```

← Projects listing page
 ← Optional layout **for** all project
 ← Dynamic route **for** each project
 ← Dynamic detail page
 ← Template **for** project details
 ← Optional subcomponent
 ← Optional subcomponent
 ← Optional subcomponent
 ← Export file **for** the dynamic page
 ← Optional **local** components only
 ← Example API route

	<ul style="list-style-type: none">└─ index.js└─ forms/<ul style="list-style-type: none">└─ Input.jsx└─ TextArea.jsx└─ Label.jsx└─ images/<ul style="list-style-type: none">└─ Avatar.jsx└─ Logo.jsx└─ typography/<ul style="list-style-type: none">└─ Heading.jsx└─ Text.jsx└─ index.js└─ icons/<ul style="list-style-type: none">└─ Icon.jsx└─ index.js└─ index.js	
	<ul style="list-style-type: none">└─ molecules/<ul style="list-style-type: none">└─ cards/<ul style="list-style-type: none">└─ ProjectCard.jsx└─ SkillCard.jsx└─ index.js└─ menus/<ul style="list-style-type: none">└─ DropdownMenu.jsx└─ SideMenu.jsx└─ index.js└─ navigation/<ul style="list-style-type: none">└─ MenuButton.jsx└─ NavLink.jsx└─ carousel/<ul style="list-style-type: none">└─ CarouselItem.jsx└─ CarouselControls.jsx└─ index.js└─ index.js	
	<ul style="list-style-type: none">└─ organisms/<ul style="list-style-type: none">└─ Navbar.jsx└─ Banner.jsx└─ ProjectsCarousel.jsx└─ SkillsCarousel.jsx└─ SkillsCategory.jsx└─ ContactForm.jsx└─ Footer.jsx└─ ProjectDetailSection.jsx	← optional reusable section for
detail pages	<ul style="list-style-type: none">└─ index.js	
	<ul style="list-style-type: none">└─ templates/<ul style="list-style-type: none">└─ HomeTemplate.jsx└─ AboutTemplate.jsx└─ ProjectsTemplate.jsx└─ SkillsTemplate.jsx└─ ContactTemplate.jsx└─ ProjectDetailTemplate.jsx	← reusable template for all

```

project detail pages
├── index.js
├── layout/
│   ├── PageWrapper.jsx
│   ├── SectionWrapper.jsx
│   ├── Container.jsx
│   └── index.js
├── contexts/
│   ├── ThemeContext.jsx
│   ├── MenuContext.jsx
│   └── index.js
├── hooks/
│   ├── useToggle.js
│   ├── useTheme.js
│   ├── useMediaQuery.js
│   ├── useCarousel.js
│   ├── useScrollPosition.js
│   └── useProjectNavigation.js
└── navigation)
    ├── index.js
    ├── lib/
    │   ├── data/
    │   │   ├── projects.js
    │   │   ├── skills.js
    │   │   ├── categories.js
    │   │   ├── socials.js
    │   │   └── navLinks.js
    │   ├── constants/
    │   │   ├── routes.js
    │   │   ├── theme.js
    │   │   └── breakpoints.js
    │   ├── utils/
    │   │   ├── classNames.js
    │   │   ├── formatDate.js
    │   │   ├── scrollToSection.js
    │   │   └── getProjectBySlug.js
    └── techStack, etc.
        ├── slug
        │   ├── index.js
        │   └── index.js
        ├── styles/
        │   ├── base/
        │   │   ├── animations.css
        │   │   ├── typography.css
        │   │   └── variables.css
        │   ├── components/
        │   │   ├── buttons.css
        │   │   ├── cards.css
        │   │   └── navbar.css

```

← optional (for next/prev

← includes slug, title, desc,

← helper to fetch project data by

```
├── project-detail.css          ← optional project-specific styles
├── themes/
│   ├── light.css
│   └── dark.css
├── tailwind.css
├── index.css
├── tests/
│   ├── unit/
│   │   ├── Button.test.js
│   │   ├── Navbar.test.js
│   │   ├── Carousel.test.js
│   │   ├── ProjectCard.test.js
│   │   └── ProjectDetail.test.js
│   ├── integration/
│   │   ├── HomePage.test.js
│   │   └── ProjectPage.test.js
│   └── mocks/
│       └── dataMock.js
├── .env.local
├── .gitignore
├── package.json
├── next.config.js
├── postcss.config.js
├── tailwind.config.js
└── README.md
```

1. Atoms (Basic Building Blocks)

Buttons ([components/atoms/buttons/](#))

Button.jsx

```
import { motion } from 'framer-motion';

const Button = ({
  children,
  variant = 'primary',
  size = 'md',
  onClick,
  disabled = false,
  className = '',
  ...props
}) => {
  const baseClasses = 'font-semibold rounded-lg transition-all duration-200
focus:outline-none focus:ring-2';

  const variants = {
    primary: 'bg-blue-600 text-white hover:bg-blue-700 focus:ring-blue-300',
    secondary: 'bg-gray-600 text-white hover:bg-gray-700 focus:ring-gray-300',
```

```

    outline: 'border-2 border-blue-600 text-blue-600 hover:bg-blue-600 hover:text-
white',
    ghost: 'text-gray-700 hover:bg-gray-100'
  };

  const sizes = {
    sm: 'px-3 py-1.5 text-sm',
    md: 'px-4 py-2 text-base',
    lg: 'px-6 py-3 text-lg'
  };

  const classes = `${baseClasses} ${variants[variant]} ${sizes[size]} ${disabled ?
'opacity-50 cursor-not-allowed' : ''} ${className}`;

  return (
    <motion.button
      whileHover={!disabled ? { scale: 1.02 } : {}}
      whileTap={!disabled ? { scale: 0.98 } : {}}
      className={classes}
      onClick={onClick}
      disabled={disabled}
      {...props}
    >
      {children}
    </motion.button>
  );
};

export default Button;

```

IconButton.jsx

```

import { motion } from 'framer-motion';

const IconButton = ({
  icon: Icon,
  onClick,
  size = 'md',
  variant = 'primary',
  className = '',
  ...props
}) => {
  const baseClasses = 'rounded-full flex items-center justify-center transition-
all duration-200 focus:outline-none focus:ring-2';

  const variants = {
    primary: 'bg-blue-600 text-white hover:bg-blue-700 focus:ring-blue-300',
    secondary: 'bg-gray-200 text-gray-700 hover:bg-gray-300 focus:ring-gray-300',
    ghost: 'text-gray-500 hover:bg-gray-100 hover:text-gray-700'
  };

  const sizes = {

```

```

    sm: 'w-8 h-8',
    md: 'w-10 h-10',
    lg: 'w-12 h-12'
  };

  const iconSizes = {
    sm: 16,
    md: 20,
    lg: 24
  };

  const classes = `${baseClasses} ${variants[variant]} ${sizes[size]} ${className}`;

  return (
    <motion.button
      whileHover={{ scale: 1.05 }}
      whileTap={{ scale: 0.95 }}
      className={classes}
      onClick={onClick}
      {...props}
    >
      <Icon size={iconSizes[size]} />
    </motion.button>
  );
};

export default IconButton;

```

index.js

```

export { default as Button } from './Button';
export { default as IconButton } from './IconButton';

```

Purpose: Basic interactive elements that can't be broken down further.

Typography ([components/atoms/typography/](#))

Heading.jsx

```

const Heading = ({
  as: Component = 'h1',
  size = 'xl',
  children,
  className = '',
  ...props
}) => {
  const sizes = {
    xs: 'text-lg font-semibold',

```

```
    sm: 'text-xl font-semibold',
    md: 'text-2xl font-bold',
    lg: 'text-3xl font-bold',
    xl: 'text-4xl font-bold',
    '2xl': 'text-5xl font-bold'
  };

  const classes = `${sizes[size]} ${className}`;

  return (
    <Component className={classes} {...props}>
      {children}
    </Component>
  );
};

export default Heading;
```

Text.jsx

```
const Text = ({
  children,
  variant = 'body',
  className = '',
  ...props
}) => {
  const variants = {
    body: 'text-base text-gray-700',
    caption: 'text-sm text-gray-500',
    lead: 'text-lg text-gray-600',
    small: 'text-sm text-gray-600'
  };

  const classes = `${variants[variant]} ${className}`;

  return (
    <p className={classes} {...props}>
      {children}
    </p>
  );
};

export default Text;
```

Purpose: Basic text elements with consistent styling.

2. Molecules (Groups of Atoms)

Cards ([components/molecules/cards/](#))

ProjectCard.jsx

```

import { motion } from 'framer-motion';
import Image from 'next/image';
import { Button } from '../../atoms/buttons';
import { Heading, Text } from '../../atoms/typography';

const ProjectCard = ({
  project,
  onViewDetails,
  className = ''
}) => {
  return (
    <motion.div
      whileHover={{ y: -5 }}
      className={`bg-white rounded-xl shadow-lg overflow-hidden hover:shadow-xl
        transition-all duration-300 ${className}`}
    >
      <div className="relative h-48 overflow-hidden">
        <Image
          src={project.image}
          alt={project.name}
          fill
          className="object-cover transition-transform duration-300 hover:scale-
110"
        />
        <div className="absolute inset-0 bg-black bg-opacity-0 hover:bg-opacity-20
        transition-all duration-300" />
      </div>

      <div className="p-6">
        <Heading as="h3" size="sm" className="mb-2">
          {project.name}
        </Heading>
        <Text variant="small" className="mb-4 line-clamp-2">
          {project.description}
        </Text>

        <div className="flex flex-wrap gap-2 mb-4">
          {project.technologies.slice(0, 3).map((tech) => (
            <span
              key={tech}
              className="px-2 py-1 bg-blue-100 text-blue-800 text-xs rounded"
            >
              {tech}
            </span>
          ))}
        </div>

        <Button
          variant="outline"
          size="sm"
          onClick={() => onViewDetails(project.id)}

```



```

        className="w-full"
      >
        View Details
      </Button>
    </div>
  </motion.div>
);
};

export default ProjectCard;

```

Purpose: Combine multiple atoms to create reusable card components.

Carousel Components ([components/molecules/carousel/](#))

CarouselItem.jsx

```

import { motion } from 'framer-motion';
import Image from 'next/image';

const CarouselItem = ({
  item,
  isActive,
  onClick,
  position = 'center',
  className = ''
}) => {
  const getPositionStyles = () => {
    switch (position) {
      case 'left':
        return 'scale-75 opacity-60 -translate-x-20 z-10';
      case 'right':
        return 'scale-75 opacity-60 translate-x-20 z-10';
      case 'center':
      default:
        return 'scale-100 opacity-100 z-20';
    }
  };

  return (
    <motion.div
      layout
      initial={{ opacity: 0, scale: 0.8 }}
      animate={{ opacity: isActive ? 1 : 0.6, scale: isActive ? 1 : 0.8 }}
      transition={{ duration: 0.5 }}
      className={`absolute inset-0 transition-all duration-500 cursor-pointer
        ${getPositionStyles()} ${className}`}
      onClick={onClick}
    >
      <Image
        src={item.image}

```

```

      alt={item.name}
      fill
      className="object-cover rounded-xl"
    />
    {isActive && (
      <motion.div
        initial={{ opacity: 0, y: 20 }}
        animate={{ opacity: 1, y: 0 }}
        className="absolute bottom-0 left-0 right-0 p-6 bg-gradient-to-t from-black/80 to-transparent text-white rounded-b-xl"
      >
        <h3 className="text-xl font-bold mb-2">{item.name}</h3>
        <p className="text-sm opacity-90">{item.description}</p>
      </motion.div>
    )}
  </motion.div>
);
};

export default CarouselItem;

```

CarouselControls.jsx

```

import { IconButton } from '../../atoms/buttons';
import { FaChevronLeft, FaChevronRight } from 'react-icons/fa';

const CarouselControls = ({
  onPrevious,
  onNext,
  className = ''
}) => {
  return (
    <div className={`flex gap-2 ${className}`}>
      <IconButton
        icon={FaChevronLeft}
        variant="secondary"
        onClick={onPrevious}
        aria-label="Previous slide"
      />
      <IconButton
        icon={FaChevronRight}
        variant="secondary"
        onClick={onNext}
        aria-label="Next slide"
      />
    </div>
  );
};

export default CarouselControls;

```

Purpose: Reusable carousel components that can be composed together.

3. Organisms (Complex Components)

ProjectsCarousel.jsx

```
"use client";
import { useState, useCallback, useRef, useEffect } from 'react';
import { motion, AnimatePresence } from 'framer-motion';
import { CarouselItem, CarouselControls } from '../../molecules/carousel';
import { Button } from '../../atoms/buttons';
import { useRouter } from 'next/navigation';

const ProjectsCarousel = ({ projects }) => {
  const router = useRouter();
  const [currentIndex, setCurrentIndex] = useState(0);
  const [isAutoPlaying, setIsAutoPlaying] = useState(true);
  const intervalRef = useRef(null);

  const nextSlide = useCallback(() => {
    setCurrentIndex((prev) => (prev + 1) % projects.length);
  }, [projects.length]);

  const prevSlide = useCallback(() => {
    setCurrentIndex((prev) => (prev - 1 + projects.length) % projects.length);
  }, [projects.length]);

  const startAutoPlay = useCallback(() => {
    intervalRef.current = setInterval(nextSlide, 5000);
  }, [nextSlide]);

  const stopAutoPlay = useCallback(() => {
    clearInterval(intervalRef.current);
  }, []);

  useEffect(() => {
    if (isAutoPlaying) {
      startAutoPlay();
    }
    return () => stopAutoPlay();
  }, [isAutoPlaying, startAutoPlay, stopAutoPlay]);

  const handleViewDetails = (projectId) => {
    router.push(`/projects/${projectId}`);
  };

  const getItemPosition = (index) => {
    const diff = (index - currentIndex + projects.length) % projects.length;
    if (diff === 0) return 'center';
    if (diff === 1) return 'right';
    if (diff === projects.length - 1) return 'left';
    return 'hidden';
  };
};
```

```

    };

    return (
      <div
        className="relative h-screen w-full overflow-hidden"
        onMouseEnter={() => setIsAutoPlaying(false)}
        onMouseLeave={() => setIsAutoPlaying(true)}
      >
        <AnimatePresence mode="wait">
          {projects.map((project, index) => {
            const position = getItemPosition(index);
            if (position === 'hidden') return null;

            return (
              <CarouselItem
                key={project.id}
                item={project}
                isActive={position === 'center'}
                position={position}
                onClick={() => position !== 'center' && setCurrentIndex(index)}
              />
            );
          })}
        </AnimatePresence>

        {/* Active Project Info */}
        <motion.div
          initial={{ opacity: 0, x: -50 }}
          animate={{ opacity: 1, x: 0 }}
          className="absolute top-1/2 left-20 -translate-y-1/2 z-30 text-white max-w-md"
        >
          <h2 className="text-5xl font-bold mb-4">
            {projects[currentIndex].name}
          </h2>
          <p className="text-lg mb-6 opacity-90">
            {projects[currentIndex].description}
          </p>
          <Button
            onClick={() => handleViewDetails(projects[currentIndex].id)}
            size="lg"
          >
            View Project
          </Button>
        </motion.div>

        {/* Controls */}
        <div className="absolute bottom-8 left-1/2 -translate-x-1/2 z-30">
          <CarouselControls
            onPrevious={prevSlide}
            onNext={nextSlide}
          />
        </div>

```

```

    /* Indicators */
    <div className="absolute bottom-8 right-8 z-30 flex gap-2">
      {projects.map((_, index) => (
        <button
          key={index}
          onClick={() => setCurrentIndex(index)}
          className={`w-3 h-3 rounded-full transition-all ${
            index === currentIndex
              ? 'bg-white scale-125'
              : 'bg-white/50 hover:bg-white/80'
            }`}
        />
      ))}
    </div>
  </div>
);
};

export default ProjectsCarousel;

```

ProjectDetailSection.jsx

```

import { useState } from 'react';
import { motion, AnimatePresence } from 'framer-motion';
import Image from 'next/image';
import { Button, IconButton } from '../../atoms/buttons';
import { Heading, Text } from '../../atoms/typography';
import { FaExternalLinkAlt, FaGithub, FaTimes, FaExpand } from 'react-icons/fa';

const ProjectDetailSection = ({ project }) => {
  const [selectedImage, setSelectedImage] = useState(null);

  // Bento grid layout patterns
  const bentoLayouts = [
    [
      { row: "row-span-2", col: "col-span-2" },
      { row: "row-span-1", col: "col-span-1" },
      { row: "row-span-1", col: "col-span-1" },
      { row: "row-span-1", col: "col-span-1" },
      { row: "row-span-1", col: "col-span-2" },
      { row: "row-span-1", col: "col-span-1" }
    ]
  ];

  const getBentoLayout = (projectId) => {
    return bentoLayouts[projectId % bentoLayouts.length];
  };

  return (
    <section className="min-h-screen bg-gray-50 py-16">
      <div className="max-w-7xl mx-auto px-4">

```

```

    { /* Project Header */ }
    <motion.div
      initial={{ opacity: 0, y: 30 }}
      animate={{ opacity: 1, y: 0 }}
      className="text-center mb-16"
    >
      <Heading as="h1" size="2xl" className="mb-6">
        {project.name}
      </Heading>
      <Text variant="lead" className="max-w-3xl mx-auto mb-8">
        {project.description}
      </Text>

      <div className="flex flex-wrap justify-center gap-4 mb-8">
        <span className="bg-blue-100 text-blue-800 px-4 py-2 rounded-full
text-sm font-medium">
          {project.category}
        </span>
        <span className="bg-green-100 text-green-800 px-4 py-2 rounded-full
text-sm font-medium">
          {project.year}
        </span>
        <span className="bg-purple-100 text-purple-800 px-4 py-2 rounded-full
text-sm font-medium">
          {project.client}
        </span>
      </div>

      <div className="flex justify-center gap-4">
        <Button
          onClick={() => window.open(project.liveUrl, '_blank')}
          className="flex items-center gap-2"
        >
          <FaExternalLinkAlt />
          Live Demo
        </Button>
        <Button
          variant="outline"
          onClick={() => window.open(project.githubUrl, '_blank')}
          className="flex items-center gap-2"
        >
          <FaGithub />
          GitHub
        </Button>
      </div>
    </motion.div>

    { /* Bento Grid */ }
    <motion.div
      initial={{ opacity: 0 }}
      animate={{ opacity: 1 }}
      transition={{ delay: 0.3 }}
      className="grid grid-cols-3 grid-rows-3 gap-4 h-[70vh] mb-16"
    >

```

```

    {project.images.slice(0, 6).map((image, index) => {
      const layout = getBentoLayout(project.id)[index];
      return (
        <motion.div
          key={index}
          className={`relative ${layout.row} ${layout.col} rounded-2xl
overflow-hidden cursor-pointer group`}
          initial={{ opacity: 0, scale: 0.8 }}
          animate={{ opacity: 1, scale: 1 }}
          transition={{ duration: 0.5, delay: index * 0.1 }}
          whileHover={{ scale: 1.02 }}
          onClick={() => setSelectedImage(image)}
        >
          <Image
            src={image}
            alt={` ${project.name} - Image ${index + 1}`}
            fill
            className="object-cover transition-transform duration-300 group-
hover:scale-110"
          />
          <div className="absolute inset-0 bg-black bg-opacity-0 group-
hover:bg-opacity-30 transition-all duration-300 flex items-center justify-center">
            <FaExpand className="text-white text-2xl opacity-0 group-
hover:opacity-100 transition-all duration-300" />
          </div>
        </motion.div>
      );
    })}
  </motion.div>

  {/* Project Details */}
  <div className="grid lg:grid-cols-2 gap-12 max-w-6xl mx-auto">
    <motion.div
      initial={{ opacity: 0, x: -30 }}
      animate={{ opacity: 1, x: 0 }}
      transition={{ delay: 0.4 }}
    >
      <Heading as="h2" size="xl" className="mb-6">
        Project Overview
      </Heading>
      <Text className="mb-8 leading-relaxed">
        {project.longDescription}
      </Text>

      <div>
        <Heading as="h3" size="lg" className="mb-4">
          Key Features
        </Heading>
        <ul className="space-y-3">
          {project.features.map((feature, index) => (
            <motion.li
              key={index}
              initial={{ opacity: 0, x: -20 }}
              animate={{ opacity: 1, x: 0 }}

```

```

        transition={{ delay: 0.5 + index * 0.1 }}
        className="flex items-center gap-3"
      >
        <div className="w-2 h-2 bg-blue-500 rounded-full"></div>
        <Text>{feature}</Text>
      </motion.li>
    )))
  </ul>
</div>
</motion.div>

<motion.div
  initial={{ opacity: 0, x: 30 }}
  animate={{ opacity: 1, x: 0 }}
  transition={{ delay: 0.6 }}
>
  <Heading as="h2" size="xl" className="mb-6">
    Technologies
  </Heading>
  <div className="flex flex-wrap gap-3 mb-8">
    {project.technologies.map((tech, index) => (
      <motion.span
        key={tech}
        initial={{ opacity: 0, scale: 0.8 }}
        animate={{ opacity: 1, scale: 1 }}
        transition={{ delay: 0.7 + index * 0.1 }}
        className="bg-gray-200 text-gray-800 px-4 py-2 rounded-lg font-
medium text-sm"
      >
        {tech}
      </motion.span>
    ))}
  </div>

  <div className="bg-white rounded-2xl p-6 shadow-lg">
    <Heading as="h3" size="lg" className="mb-4">
      Project Information
    </Heading>
    <div className="space-y-3">
      <div className="flex justify-between items-center py-2 border-b
border-gray-100">
        <Text variant="caption">Category</Text>
        <Text className="font-medium">{project.category}</Text>
      </div>
      <div className="flex justify-between items-center py-2 border-b
border-gray-100">
        <Text variant="caption">Year</Text>
        <Text className="font-medium">{project.year}</Text>
      </div>
      <div className="flex justify-between items-center py-2">
        <Text variant="caption">Client</Text>
        <Text className="font-medium">{project.client}</Text>
      </div>
    </div>
  </div>

```



```

        </div>
      </motion.div>
    </div>
  </div>

  { /* Image Modal */ }
  <AnimatePresence>
    {selectedImage && (
      <motion.div
        initial={{ opacity: 0 }}
        animate={{ opacity: 1 }}
        exit={{ opacity: 0 }}
        className="fixed inset-0 bg-black bg-opacity-95 z-50 flex items-center
justify-center p-4"
        onClick={() => setSelectedImage(null)}
      >
        <motion.div
          initial={{ scale: 0.8, opacity: 0 }}
          animate={{ scale: 1, opacity: 1 }}
          exit={{ scale: 0.8, opacity: 0 }}
          className="relative max-w-6xl max-h-full"
          onClick={(e) => e.stopPropagation()}
        >
          <Image
            src={selectedImage}
            alt="Enlarged view"
            width={1200}
            height={800}
            quality={100}
            className="object-contain max-h-[90vh]"
          />
          <IconButton
            icon={FaTimes}
            onClick={() => setSelectedImage(null)}
            className="absolute top-4 right-4 bg-black/50 text-white hover:bg-
black/70"
          />
        </motion.div>
      </motion.div>
    )}
  </AnimatePresence>
</section>
);
};

export default ProjectDetailSection;

```

4. Templates (Page Layouts)

ProjectDetailTemplate.jsx

```

"use client";
import { useRouter } from 'next/navigation';
import { motion } from 'framer-motion';
import { IconButton } from '../../atoms/buttons';
import { FaChevronLeft } from 'react-icons/fa';
import ProjectDetailSection from '../../organisms/ProjectDetailSection';

const ProjectDetailTemplate = ({ project }) => {
  const router = useRouter();

  if (!project) {
    return (
      <div className="min-h-screen flex items-center justify-center">
        <div className="text-center">
          <h1 className="text-4xl font-bold mb-4">Project Not Found</h1>
          <button
            onClick={() => router.back()}
            className="bg-blue-600 text-white px-6 py-3 rounded-lg hover:bg-blue-
700 transition"
          >
            Go Back
          </button>
        </div>
      </div>
    );
  }

  return (
    <div className="min-h-screen bg-gray-50">
      { /* Back Button */ }
      <motion.div
        initial={{ opacity: 0, x: -20 }}
        animate={{ opacity: 1, x: 0 }}
        className="fixed top-6 left-6 z-40"
      >
        <IconButton
          icon={FaChevronLeft}
          variant="secondary"
          onClick={() => router.back()}
          aria-label="Go back"
        />
      </motion.div>

      <ProjectDetailSection project={project} />
    </div>
  );
};

export default ProjectDetailTemplate;

```

5. Layout Components

PageWrapper.jsx

```
import { motion } from 'framer-motion';

const PageWrapper = ({
  children,
  className = ''
}) => {
  return (
    <motion.div
      initial={{ opacity: 0, y: 20 }}
      animate={{ opacity: 1, y: 0 }}
      exit={{ opacity: 0, y: 20 }}
      transition={{ duration: 0.5 }}
      className={`min-h-screen ${className}`}
    >
      {children}
    </motion.div>
  );
};

export default PageWrapper;
```

SectionWrapper.jsx

```
import { motion } from 'framer-motion';

const SectionWrapper = ({
  children,
  className = '',
  delay = 0
}) => {
  return (
    <motion.section
      initial={{ opacity: 0, y: 30 }}
      whileInView={{ opacity: 1, y: 0 }}
      viewport={{ once: true }}
      transition={{ duration: 0.6, delay }}
      className={className}
    >
      {children}
    </motion.section>
  );
};

export default SectionWrapper;
```

Purpose Summary:

- **Atoms:** Basic UI elements (buttons, text, inputs)
- **Molecules:** Simple combinations of atoms (cards, form groups)
- **Organisms:** Complex components (navigation, carousels, forms)
- **Templates:** Page-level layouts and structures
- **Layout:** Reusable wrapper components for consistent spacing and animations

This structure provides:

- **Reusability:** Components can be used across multiple pages
- **Maintainability:** Easy to update and modify
- **Consistency:** Uniform design system
- **Scalability:** Easy to add new components
- **Performance:** Optimized with Next.js and proper component splitting! I'll explain each component category with proper code examples and purposes based on the Atomic Design methodology:

1. Atoms (Basic Building Blocks)

Buttons ([components/atoms/buttons/](#))

Button.jsx

```
import { motion } from 'framer-motion';

const Button = ({
  children,
  variant = 'primary',
  size = 'md',
  onClick,
  disabled = false,
  className = '',
  ...props
}) => {
  const baseClasses = 'font-semibold rounded-lg transition-all duration-200 focus:outline-none focus:ring-2';

  const variants = {
    primary: 'bg-blue-600 text-white hover:bg-blue-700 focus:ring-blue-300',
    secondary: 'bg-gray-600 text-white hover:bg-gray-700 focus:ring-gray-300',
    outline: 'border-2 border-blue-600 text-blue-600 hover:bg-blue-600 hover:text-white',
    ghost: 'text-gray-700 hover:bg-gray-100'
  };

  const sizes = {
    sm: 'px-3 py-1.5 text-sm',
    md: 'px-4 py-2 text-base',
    lg: 'px-6 py-3 text-lg'
  };

  const classes = `${baseClasses} ${variants[variant]} ${sizes[size]} ${disabled ? 'opacity-50 cursor-not-allowed' : ''} ${className}`;
}
```

```

    return (
      <motion.button
        whileHover={!disabled ? { scale: 1.02 } : {}}
        whileTap={!disabled ? { scale: 0.98 } : {}}
        className={classes}
        onClick={onClick}
        disabled={disabled}
        {...props}
      >
        {children}
      </motion.button>
    );
  };

  export default Button;

```

IconButton.jsx

```

import { motion } from 'framer-motion';

const IconButton = ({
  icon: Icon,
  onClick,
  size = 'md',
  variant = 'primary',
  className = '',
  ...props
}) => {
  const baseClasses = 'rounded-full flex items-center justify-center transition-all duration-200 focus:outline-none focus:ring-2';

  const variants = {
    primary: 'bg-blue-600 text-white hover:bg-blue-700 focus:ring-blue-300',
    secondary: 'bg-gray-200 text-gray-700 hover:bg-gray-300 focus:ring-gray-300',
    ghost: 'text-gray-500 hover:bg-gray-100 hover:text-gray-700'
  };

  const sizes = {
    sm: 'w-8 h-8',
    md: 'w-10 h-10',
    lg: 'w-12 h-12'
  };

  const iconSizes = {
    sm: 16,
    md: 20,
    lg: 24
  };

  const classes = `${baseClasses} ${variants[variant]} ${sizes[size]} ${className}`;

```

```

    return (
      <motion.button
        whileHover={{ scale: 1.05 }}
        whileTap={{ scale: 0.95 }}
        className={classes}
        onClick={onClick}
        {...props}
      >
        <Icon size={iconSizes[size]} />
      </motion.button>
    );
  };

  export default IconButton;

```

index.js

```

export { default as Button } from './Button';
export { default as IconButton } from './IconButton';

```

Purpose: Basic interactive elements that can't be broken down further.

Typography ([components/atoms/typography/](#))

Heading.jsx

```

const Heading = ({
  as: Component = 'h1',
  size = 'xl',
  children,
  className = '',
  ...props
}) => {
  const sizes = {
    xs: 'text-lg font-semibold',
    sm: 'text-xl font-semibold',
    md: 'text-2xl font-bold',
    lg: 'text-3xl font-bold',
    xl: 'text-4xl font-bold',
    '2xl': 'text-5xl font-bold'
  };

  const classes = `${sizes[size]} ${className}`;

  return (
    <Component className={classes} {...props}>
      {children}
    </Component>
  );
};

```

```
);  
};  
  
export default Heading;
```

Text.jsx

```
const Text = ({  
  children,  
  variant = 'body',  
  className = '',  
  ...props  
) => {  
  const variants = {  
    body: 'text-base text-gray-700',  
    caption: 'text-sm text-gray-500',  
    lead: 'text-lg text-gray-600',  
    small: 'text-sm text-gray-600'  
  };  
  
  const classes = `${variants[variant]} ${className}`;  
  
  return (  
    <p className={classes} {...props}>  
      {children}  
    </p>  
  );  
};  
  
export default Text;
```

Purpose: Basic text elements with consistent styling.

2. Molecules (Groups of Atoms)

Cards ([components/molecules/cards/](#))

ProjectCard.jsx

```
import { motion } from 'framer-motion';  
import Image from 'next/image';  
import { Button } from '../../atoms/buttons';  
import { Heading, Text } from '../../atoms/typography';  
  
const ProjectCard = ({  
  project,  
  onViewDetails,  
  className = ''  
) => {
```

```

    return (
      <motion.div
        whileHover={{ y: -5 }}
        className={`bg-white rounded-xl shadow-lg overflow-hidden hover:shadow-xl
transition-all duration-300 ${className}`}
      >
        <div className="relative h-48 overflow-hidden">
          <Image
            src={project.image}
            alt={project.name}
            fill
            className="object-cover transition-transform duration-300 hover:scale-
110"
          />
          <div className="absolute inset-0 bg-black bg-opacity-0 hover:bg-opacity-20
transition-all duration-300" />
        </div>

        <div className="p-6">
          <Heading as="h3" size="sm" className="mb-2">
            {project.name}
          </Heading>
          <Text variant="small" className="mb-4 line-clamp-2">
            {project.description}
          </Text>

          <div className="flex flex-wrap gap-2 mb-4">
            {project.technologies.slice(0, 3).map((tech) => (
              <span
                key={tech}
                className="px-2 py-1 bg-blue-100 text-blue-800 text-xs rounded"
              >
                {tech}
              </span>
            ))}
          </div>

          <Button
            variant="outline"
            size="sm"
            onClick={() => onViewDetails(project.id)}
            className="w-full"
          >
            View Details
          </Button>
        </div>
      </motion.div>
    );
  };

  export default ProjectCard;

```

Purpose: Combine multiple atoms to create reusable card components.

Carousel Components ([components/molecules/carousel/](#))

CarouselItem.jsx

```
import { motion } from 'framer-motion';
import Image from 'next/image';

const CarouselItem = ({
  item,
  isActive,
  onClick,
  position = 'center',
  className = ''
}) => {
  const getPositionStyles = () => {
    switch (position) {
      case 'left':
        return 'scale-75 opacity-60 -translate-x-20 z-10';
      case 'right':
        return 'scale-75 opacity-60 translate-x-20 z-10';
      case 'center':
      default:
        return 'scale-100 opacity-100 z-20';
    }
  };

  return (
    <motion.div
      layout
      initial={{ opacity: 0, scale: 0.8 }}
      animate={{ opacity: isActive ? 1 : 0.6, scale: isActive ? 1 : 0.8 }}
      transition={{ duration: 0.5 }}
      className={`absolute inset-0 transition-all duration-500 cursor-pointer
        ${getPositionStyles()} ${className}`}
      onClick={onClick}
    >
      <Image
        src={item.image}
        alt={item.name}
        fill
        className="object-cover rounded-xl"
      />
      {isActive && (
        <motion.div
          initial={{ opacity: 0, y: 20 }}
          animate={{ opacity: 1, y: 0 }}
          className="absolute bottom-0 left-0 right-0 p-6 bg-gradient-to-t from-black/80 to-transparent text-white rounded-b-xl"
        >
          <h3 className="text-xl font-bold mb-2">{item.name}</h3>
          <p className="text-sm opacity-90">{item.description}</p>
        </motion.div>
      )}
    </motion.div>
  );
};
```

```

    })
  </motion.div>
);
};

export default CarouselItem;

```

CarouselControls.jsx

```

import { IconButton } from '../../../atoms/buttons';
import { FaChevronLeft, FaChevronRight } from 'react-icons/fa';

const CarouselControls = ({
  onPrevious,
  onNext,
  className = ''
}) => {
  return (
    <div className={`flex gap-2 ${className}`}>
      <IconButton
        icon={FaChevronLeft}
        variant="secondary"
        onClick={onPrevious}
        aria-label="Previous slide"
      />
      <IconButton
        icon={FaChevronRight}
        variant="secondary"
        onClick={onNext}
        aria-label="Next slide"
      />
    </div>
  );
};

export default CarouselControls;

```

Purpose: Reusable carousel components that can be composed together.

3. Organisms (Complex Components)

ProjectsCarousel.jsx

```

"use client";
import { useState, useCallback, useRef, useEffect } from 'react';
import { motion, AnimatePresence } from 'framer-motion';
import { CarouselItem, CarouselControls } from '../../../molecules/carousel';
import { Button } from '../../../atoms/buttons';
import { useRouter } from 'next/navigation';

```

```
const ProjectsCarousel = ({ projects }) => {
  const router = useRouter();
  const [currentIndex, setCurrentIndex] = useState(0);
  const [isAutoPlaying, setIsAutoPlaying] = useState(true);
  const intervalRef = useRef(null);

  const nextSlide = useCallback(() => {
    setCurrentIndex((prev) => (prev + 1) % projects.length);
  }, [projects.length]);

  const prevSlide = useCallback(() => {
    setCurrentIndex((prev) => (prev - 1 + projects.length) % projects.length);
  }, [projects.length]);

  const startAutoPlay = useCallback(() => {
    intervalRef.current = setInterval(nextSlide, 5000);
  }, [nextSlide]);

  const stopAutoPlay = useCallback(() => {
    clearInterval(intervalRef.current);
  }, []);

  useEffect(() => {
    if (isAutoPlaying) {
      startAutoPlay();
    }
    return () => stopAutoPlay();
  }, [isAutoPlaying, startAutoPlay, stopAutoPlay]);

  const handleViewDetails = (projectId) => {
    router.push(`/projects/${projectId}`);
  };

  const getItemPosition = (index) => {
    const diff = (index - currentIndex + projects.length) % projects.length;
    if (diff === 0) return 'center';
    if (diff === 1) return 'right';
    if (diff === projects.length - 1) return 'left';
    return 'hidden';
  };

  return (
    <div
      className="relative h-screen w-full overflow-hidden"
      onMouseEnter={() => setIsAutoPlaying(false)}
      onMouseLeave={() => setIsAutoPlaying(true)}
    >
      <AnimatePresence mode="wait">
        {projects.map((project, index) => {
          const position = getItemPosition(index);
          if (position === 'hidden') return null;

          return (
```

```

        <CarouselItem
          key={project.id}
          item={project}
          isActive={position === 'center'}
          position={position}
          onClick={() => position !== 'center' && setCurrentIndex(index)}
        />
      );
    }
  }
</AnimatePresence>

{/* Active Project Info */}
<motion.div
  initial={{ opacity: 0, x: -50 }}
  animate={{ opacity: 1, x: 0 }}
  className="absolute top-1/2 left-20 -translate-y-1/2 z-30 text-white max-
w-md"
>
  <h2 className="text-5xl font-bold mb-4">
    {projects[currentIndex].name}
  </h2>
  <p className="text-lg mb-6 opacity-90">
    {projects[currentIndex].description}
  </p>
  <Button
    onClick={() => handleViewDetails(projects[currentIndex].id)}
    size="lg"
  >
    View Project
  </Button>
</motion.div>

{/* Controls */}
<div className="absolute bottom-8 left-1/2 -translate-x-1/2 z-30">
  <CarouselControls
    onPrevious={prevSlide}
    onNext={nextSlide}
  />
</div>

{/* Indicators */}
<div className="absolute bottom-8 right-8 z-30 flex gap-2">
  {projects.map((_, index) => (
    <button
      key={index}
      onClick={() => setCurrentIndex(index)}
      className={`w-3 h-3 rounded-full transition-all ${
        index === currentIndex
          ? 'bg-white scale-125'
          : 'bg-white/50 hover:bg-white/80'
      }`}
    />
  ))}
</div>

```

```

    </div>
  );
};

export default ProjectsCarousel;

```

ProjectDetailSection.jsx

```

import { useState } from 'react';
import { motion, AnimatePresence } from 'framer-motion';
import Image from 'next/image';
import { Button, IconButton } from '../../atoms/buttons';
import { Heading, Text } from '../../atoms/typography';
import { FaExternalLinkAlt, FaGithub, FaTimes, FaExpand } from 'react-icons/fa';

const ProjectDetailSection = ({ project }) => {
  const [selectedImage, setSelectedImage] = useState(null);

  // Bento grid layout patterns
  const bentoLayouts = [
    [
      { row: "row-span-2", col: "col-span-2" },
      { row: "row-span-1", col: "col-span-1" },
      { row: "row-span-1", col: "col-span-1" },
      { row: "row-span-1", col: "col-span-1" },
      { row: "row-span-1", col: "col-span-2" },
      { row: "row-span-1", col: "col-span-1" }
    ]
  ];

  const getBentoLayout = (projectId) => {
    return bentoLayouts[projectId % bentoLayouts.length];
  };

  return (
    <section className="min-h-screen bg-gray-50 py-16">
      <div className="max-w-7xl mx-auto px-4">
        {/* Project Header */}
        <motion.div
          initial={{ opacity: 0, y: 30 }}
          animate={{ opacity: 1, y: 0 }}
          className="text-center mb-16">
          >
            <Heading as="h1" size="2xl" className="mb-6">
              {project.name}
            </Heading>
            <Text variant="lead" className="max-w-3xl mx-auto mb-8">
              {project.description}
            </Text>

            <div className="flex flex-wrap justify-center gap-4 mb-8">

```

```

        <span className="bg-blue-100 text-blue-800 px-4 py-2 rounded-full
text-sm font-medium">
            {project.category}
        </span>
        <span className="bg-green-100 text-green-800 px-4 py-2 rounded-full
text-sm font-medium">
            {project.year}
        </span>
        <span className="bg-purple-100 text-purple-800 px-4 py-2 rounded-full
text-sm font-medium">
            {project.client}
        </span>
    </div>

    <div className="flex justify-center gap-4">
        <Button
            onClick={() => window.open(project.liveUrl, '_blank')}
            className="flex items-center gap-2"
        >
            <FaExternalLinkAlt />
            Live Demo
        </Button>
        <Button
            variant="outline"
            onClick={() => window.open(project.githubUrl, '_blank')}
            className="flex items-center gap-2"
        >
            <FaGithub />
            GitHub
        </Button>
    </div>
</motion.div>

{/* Bento Grid */}
<motion.div
    initial={{ opacity: 0 }}
    animate={{ opacity: 1 }}
    transition={{ delay: 0.3 }}
    className="grid grid-cols-3 grid-rows-3 gap-4 h-[70vh] mb-16"
>
    {project.images.slice(0, 6).map((image, index) => {
        const layout = getBentoLayout(project.id)[index];
        return (
            <motion.div
                key={index}
                className={`relative ${layout.row} ${layout.col} rounded-2xl
overflow-hidden cursor-pointer group`}
                initial={{ opacity: 0, scale: 0.8 }}
                animate={{ opacity: 1, scale: 1 }}
                transition={{ duration: 0.5, delay: index * 0.1 }}
                whileHover={{ scale: 1.02 }}
                onClick={() => setSelectedImage(image)}
            >
                <Image

```

```

        src={image}
        alt={` ${project.name} - Image ${index + 1}`}
        fill
        className="object-cover transition-transform duration-300 group-
hover:scale-110"
      />
      <div className="absolute inset-0 bg-black bg-opacity-0 group-
hover:bg-opacity-30 transition-all duration-300 flex items-center justify-center">
        <FaExpand className="text-white text-2xl opacity-0 group-
hover:opacity-100 transition-all duration-300" />
      </div>
    </motion.div>
  );
}
</motion.div>

{/* Project Details */}
<div className="grid lg:grid-cols-2 gap-12 max-w-6xl mx-auto">
  <motion.div
    initial={{ opacity: 0, x: -30 }}
    animate={{ opacity: 1, x: 0 }}
    transition={{ delay: 0.4 }}
  >
    <Heading as="h2" size="xl" className="mb-6">
      Project Overview
    </Heading>
    <Text className="mb-8 leading-relaxed">
      {project.longDescription}
    </Text>

    <div>
      <Heading as="h3" size="lg" className="mb-4">
        Key Features
      </Heading>
      <ul className="space-y-3">
        {project.features.map((feature, index) => (
          <motion.li
            key={index}
            initial={{ opacity: 0, x: -20 }}
            animate={{ opacity: 1, x: 0 }}
            transition={{ delay: 0.5 + index * 0.1 }}
            className="flex items-center gap-3"
          >
            <div className="w-2 h-2 bg-blue-500 rounded-full"></div>
            <Text>{feature}</Text>
          </motion.li>
        ))}
      </ul>
    </div>
  </motion.div>

  <motion.div
    initial={{ opacity: 0, x: 30 }}
    animate={{ opacity: 1, x: 0 }}

```

```

        transition={{ delay: 0.6 }}
      >
      <Heading as="h2" size="xl" className="mb-6">
        Technologies
      </Heading>
      <div className="flex flex-wrap gap-3 mb-8">
        {project.technologies.map((tech, index) => (
          <motion.span
            key={tech}
            initial={{ opacity: 0, scale: 0.8 }}
            animate={{ opacity: 1, scale: 1 }}
            transition={{ delay: 0.7 + index * 0.1 }}
            className="bg-gray-200 text-gray-800 px-4 py-2 rounded-lg font-
medium text-sm"
          >
            {tech}
          </motion.span>
        ))}
      </div>

      <div className="bg-white rounded-2xl p-6 shadow-lg">
        <Heading as="h3" size="lg" className="mb-4">
          Project Information
        </Heading>
        <div className="space-y-3">
          <div className="flex justify-between items-center py-2 border-b
border-gray-100">
            <Text variant="caption">Category</Text>
            <Text className="font-medium">{project.category}</Text>
          </div>
          <div className="flex justify-between items-center py-2 border-b
border-gray-100">
            <Text variant="caption">Year</Text>
            <Text className="font-medium">{project.year}</Text>
          </div>
          <div className="flex justify-between items-center py-2">
            <Text variant="caption">Client</Text>
            <Text className="font-medium">{project.client}</Text>
          </div>
        </div>
      </div>
    </motion.div>
  </div>
</div>

{/* Image Modal */}
<AnimatePresence>
  {selectedImage && (
    <motion.div
      initial={{ opacity: 0 }}
      animate={{ opacity: 1 }}
      exit={{ opacity: 0 }}
      className="fixed inset-0 bg-black bg-opacity-95 z-50 flex items-center
justify-center p-4"
    >

```



```

        onClick={() => setSelectedImage(null)}
      >
        <motion.div
          initial={{ scale: 0.8, opacity: 0 }}
          animate={{ scale: 1, opacity: 1 }}
          exit={{ scale: 0.8, opacity: 0 }}
          className="relative max-w-6xl max-h-full"
          onClick={(e) => e.stopPropagation()}
        >
          <Image
            src={selectedImage}
            alt="Enlarged view"
            width={1200}
            height={800}
            quality={100}
            className="object-contain max-h-[90vh]"
          />
          <IconButton
            icon={FaTimes}
            onClick={() => setSelectedImage(null)}
            className="absolute top-4 right-4 bg-black/50 text-white hover:bg-
black/70"
          />
        </motion.div>
      </motion.div>
    )}
  </AnimatePresence>
</section>
);
};

export default ProjectDetailSection;

```

4. Templates (Page Layouts)

ProjectDetailTemplate.jsx

```

"use client";
import { useRouter } from 'next/navigation';
import { motion } from 'framer-motion';
import { IconButton } from '../../atoms/buttons';
import { FaChevronLeft } from 'react-icons/fa';
import ProjectDetailSection from '../../organisms/ProjectDetailSection';

const ProjectDetailTemplate = ({ project }) => {
  const router = useRouter();

  if (!project) {
    return (
      <div className="min-h-screen flex items-center justify-center">
        <div className="text-center">

```

```

        <h1 className="text-4xl font-bold mb-4">Project Not Found</h1>
        <button
          onClick={() => router.back()}
          className="bg-blue-600 text-white px-6 py-3 rounded-lg hover:bg-blue-
700 transition"
        >
          Go Back
        </button>
      </div>
    </div>
  );
}

return (
  <div className="min-h-screen bg-gray-50">
    { /* Back Button */ }
    <motion.div
      initial={{ opacity: 0, x: -20 }}
      animate={{ opacity: 1, x: 0 }}
      className="fixed top-6 left-6 z-40"
    >
      <IconButton
        icon={FaChevronLeft}
        variant="secondary"
        onClick={() => router.back()}
        aria-label="Go back"
      />
    </motion.div>

    <ProjectDetailSection project={project} />
  </div>
);
};

export default ProjectDetailTemplate;

```

5. Layout Components

PageWrapper.jsx

```

import { motion } from 'framer-motion';

const PageWrapper = ({
  children,
  className = ''
}) => {
  return (
    <motion.div
      initial={{ opacity: 0, y: 20 }}
      animate={{ opacity: 1, y: 0 }}
      exit={{ opacity: 0, y: 20 }}
    >

```

```
        transition={{ duration: 0.5 }}
        className={`min-h-screen ${className}`}
      >
        {children}
      </motion.div>
    );
  };

export default PageWrapper;
```

SectionWrapper.jsx

```
import { motion } from 'framer-motion';

const SectionWrapper = ({
  children,
  className = '',
  delay = 0
}) => {
  return (
    <motion.section
      initial={{ opacity: 0, y: 30 }}
      whileInView={{ opacity: 1, y: 0 }}
      viewport={{ once: true }}
      transition={{ duration: 0.6, delay }}
      className={className}
    >
      {children}
    </motion.section>
  );
};

export default SectionWrapper;
```

Purpose Summary:

- **Atoms:** Basic UI elements (buttons, text, inputs)
- **Molecules:** Simple combinations of atoms (cards, form groups)
- **Organisms:** Complex components (navigation, carousels, forms)
- **Templates:** Page-level layouts and structures
- **Layout:** Reusable wrapper components for consistent spacing and animations

This structure provides:

- **Reusability:** Components can be used across multiple pages
- **Maintainability:** Easy to update and modify
- **Consistency:** Uniform design system
- **Scalability:** Easy to add new components
- **Performance:** Optimized with Next.js and proper component splitting

I'll explain each file in the **lib** and **styles** directories with code examples and purposes:

Lib Directory

Data Files - Centralized data management

lib/data/projects.js

```
export const projects = [
  {
    slug: 'ecommerce-platform',
    title: 'E-Commerce Platform',
    description: 'A full-stack e-commerce solution with modern features',
    shortDescription: 'Modern online shopping platform',
    techStack: ['Next.js', 'Node.js', 'MongoDB', 'Stripe', 'Tailwind CSS'],
    images: [
      '/assets/images/projects/ecommerce-1.jpg',
      '/assets/images/projects/ecommerce-2.jpg'
    ],
    liveUrl: 'https://ecommerce-demo.com',
    githubUrl: 'https://github.com/username/ecommerce',
    featured: true,
    category: 'Full Stack',
    startDate: '2024-01-15',
    endDate: '2024-03-20',
    status: 'Completed',
    highlights: [
      'Integrated payment processing with Stripe',
      'Real-time inventory management',
      'Admin dashboard for product management'
    ]
  },
  {
    slug: 'task-management-app',
    title: 'Task Management App',
    description: 'Productivity application for team task management',
    // ... more properties
  }
];

export const getProjectBySlug = (slug) => {
  return projects.find(project => project.slug === slug);
};

export const getFeaturedProjects = () => {
  return projects.filter(project => project.featured);
};

export const getProjectsByCategory = (category) => {
  return projects.filter(project => project.category === category);
};
```

lib/data/skills.js

```
export const skills = [
  {
    name: 'JavaScript',
    category: 'language',
    level: 90,
    icon: 'SiJavascript', // React Icons name
    color: '#F7DF1E',
    years: 4
  },
  {
    name: 'React',
    category: 'frontend',
    level: 85,
    icon: 'SiReact',
    color: '#61DAFB',
    years: 3
  },
  {
    name: 'Node.js',
    category: 'backend',
    level: 80,
    icon: 'SiNodedotjs',
    color: '#339933',
    years: 3
  }
];

export const skillCategories = [
  { id: 'language', name: 'Languages' },
  { id: 'frontend', name: 'Frontend' },
  { id: 'backend', name: 'Backend' },
  { id: 'tool', name: 'Tools' }
];

export const getSkillsByCategory = (category) => {
  return skills.filter(skill => skill.category === category);
};
```

lib/data/socials.js

```
export const socialLinks = [
  {
    name: 'GitHub',
    url: 'https://github.com/username',
    icon: 'SiGithub',
    color: '#333'
  },
  {
```

```
    name: 'LinkedIn',
    url: 'https://linkedin.com/in/username',
    icon: 'SiLinkedin',
    color: '#0077B5'
  },
  {
    name: 'Twitter',
    url: 'https://twitter.com/username',
    icon: 'SiTwitter',
    color: '#1DA1F2'
  }
];
```

lib/data/navLinks.js

```
export const navigationLinks = [
  { name: 'Home', path: '/', icon: 'HiHome' },
  { name: 'About', path: '/about', icon: 'HiUser' },
  { name: 'Projects', path: '/projects', icon: 'HiFolder' },
  { name: 'Skills', path: '/skills', icon: 'HiCode' },
  { name: 'Contact', path: '/contact', icon: 'HiMail' }
];

export const footerLinks = [
  { name: 'Privacy Policy', path: '/privacy' },
  { name: 'Terms of Service', path: '/terms' }
];
```

Constants Files - App-wide constants

lib/constants/routes.js

```
export const ROUTES = {
  HOME: '/',
  ABOUT: '/about',
  PROJECTS: '/projects',
  SKILLS: '/skills',
  CONTACT: '/contact'
};

export const API_ROUTES = {
  CONTACT: '/api/contact',
  PROJECTS: '/api/projects'
};

export const EXTERNAL_LINKS = {
  GITHUB: 'https://github.com/username',
  LINKEDIN: 'https://linkedin.com/in/username'
};
```

lib/constants/theme.js

```
export const THEME = {
  LIGHT: 'light',
  DARK: 'dark'
};

export const COLORS = {
  primary: {
    light: '#3B82F6',
    dark: '#60A5FA'
  },
  secondary: {
    light: '#6B7280',
    dark: '#9CA3AF'
  },
  background: {
    light: 'FFFFFF',
    dark: '#111827'
  }
};

export const BREAKPOINTS = {
  sm: '640px',
  md: '768px',
  lg: '1024px',
  xl: '1280px',
  '2xl': '1536px'
};
```

lib/constants/breakpoints.js

```
export const BREAKPOINT_VALUES = {
  xs: 0,
  sm: 640,
  md: 768,
  lg: 1024,
  xl: 1280,
  '2xl': 1536
};

export const MEDIA_QUERIES = {
  sm: `(min-width: ${BREAKPOINT_VALUES.sm}px)`,
  md: `(min-width: ${BREAKPOINT_VALUES.md}px)`,
  lg: `(min-width: ${BREAKPOINT_VALUES.lg}px)`,
  xl: `(min-width: ${BREAKPOINT_VALUES.xl}px)`,
  '2xl': `(min-width: ${BREAKPOINT_VALUES['2xl']}px)`
};
```

Utils Files - Reusable utility functions

lib/utils/classNames.js

```
/**
 * Conditionally joins classNames together
 * @param {...any} classes - Class names or objects
 * @returns {string} - Combined class string
 */
export function classNames(...classes) {
  return classes
    .filter(Boolean)
    .map(cls => {
      if (typeof cls === 'object') {
        return Object.entries(cls)
          .filter(([, value]) => value)
          .map(([key]) => key)
          .join(' ');
      }
      return cls;
    })
    .join(' ')
    .trim();
}

// Usage: classNames('btn', 'btn-primary', { 'btn-disabled': isDisabled })
```

lib/utils/formatDate.js

```
/**
 * Format date to readable string
 * @param {string|Date} date - Date to format
 * @param {Object} options - Intl.DateTimeFormat options
 * @returns {string} - Formatted date
 */
export function formatDate(date, options = {}) {
  const defaultOptions = {
    year: 'numeric',
    month: 'long',
    day: 'numeric'
  };

  return new Date(date).toLocaleDateString('en-US', {
    ...defaultOptions,
    ...options
  });
}

// Usage: formatDate('2024-01-15') → "January 15, 2024"
```


lib/utils/scrollToSection.js

```
import { animateScroll as scroll } from 'react-scroll';

/**
 * Smooth scroll to section
 * @param {string} sectionId - ID of the section to scroll to
 * @param {Object} options - Scroll options
 */
export function scrollToSection(sectionId, options = {}) {
  const element = document.getElementById(sectionId);
  if (element) {
    const defaultOptions = {
      duration: 800,
      smooth: true,
      offset: -80 // Account for fixed header
    };

    scroll.scrollTo(element.offsetTop - 80, {
      ...defaultOptions,
      ...options
    });
  }
}

/**
 * Scroll to top of page
 */
export function scrollToTop() {
  scroll.scrollToTop({
    duration: 500,
    smooth: true
  });
}
```

lib/utils/getProjectBySlug.js

```
import { projects } from '@lib/data/projects';

/**
 * Get project by slug
 * @param {string} slug - Project slug
 * @returns {Object|null} - Project object or null
 */
export function getProjectBySlug(slug) {
  return projects.find(project => project.slug === slug) || null;
}

/**
 * Get next and previous projects for navigation
```

```

* @param {string} currentSlug - Current project slug
* @returns {Object} - { previous: Project, next: Project }
*/
export function getProjectNavigation(currentSlug) {
  const currentIndex = projects.findIndex(p => p.slug === currentSlug);

  return {
    previous: projects[currentIndex - 1] || null,
    next: projects[currentIndex + 1] || null
  };
}

```

lib/utils/index.js - Barrel export

```

export * from './classNames';
export * from './formatDate';
export * from './scrollToSection';
export * from './getProjectBySlug';
// Export other utils...

```

lib/index.js - Main barrel export

```

export * from './data';
export * from './constants';
export * from './utils';

```

Styles Directory

Base Styles - Foundation styles

styles/base/variables.css

```

:root {
  /* Colors */
  --color-primary: 59 130 246; /* rgb(59 130 246) */
  --color-secondary: 107 114 128; /* rgb(107 114 128) */
  --color-accent: 139 92 246; /* rgb(139 92 246) */

  /* Background Colors */
  --color-bg-primary: 255 255 255;
  --color-bg-secondary: 249 250 251;

  /* Text Colors */
  --color-text-primary: 17 24 39;
  --color-text-secondary: 107 114 128;

  /* Border Colors */

```

```

--color-border: 229 231 235;

/* Spacing */
--spacing-xs: 0.25rem;
--spacing-sm: 0.5rem;
--spacing-md: 1rem;
--spacing-lg: 1.5rem;
--spacing-xl: 2rem;

/* Border Radius */
--radius-sm: 0.25rem;
--radius-md: 0.5rem;
--radius-lg: 0.75rem;
--radius-xl: 1rem;

/* Shadows */
--shadow-sm: 0 1px 2px 0 rgb(0 0 0 / 0.05);
--shadow-md: 0 4px 6px -1px rgb(0 0 0 / 0.1);
--shadow-lg: 0 10px 15px -3px rgb(0 0 0 / 0.1);
}

/* Dark theme variables */
[data-theme='dark'] {
  --color-bg-primary: 17 24 39;
  --color-bg-secondary: 31 41 55;
  --color-text-primary: 249 250 251;
  --color-text-secondary: 156 163 175;
  --color-border: 55 65 81;
}

```

styles/base/typography.css

```

@import url('https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap');

:root {
  --font-family-sans: 'Inter', system-ui, -apple-system, sans-serif;
  --font-family-mono: 'Fira Code', monospace;
}

body {
  font-family: var(--font-family-sans);
  line-height: 1.6;
  font-weight: 400;
}

h1, h2, h3, h4, h5, h6 {
  font-weight: 600;
  line-height: 1.2;
  margin-bottom: var(--spacing-md);
}

```

```
h1 {
  font-size: 2.5rem;
  font-weight: 700;
}

h2 {
  font-size: 2rem;
}

h3 {
  font-size: 1.75rem;
}

h4 {
  font-size: 1.5rem;
}

p {
  margin-bottom: var(--spacing-md);
}

a {
  color: inherit;
  text-decoration: none;
  transition: color 0.2s ease;
}

a:hover {
  color: rgb(var(--color-primary));
}

code {
  font-family: var(--font-family-mono);
  background: rgb(var(--color-bg-secondary));
  padding: var(--spacing-xs) var(--spacing-sm);
  border-radius: var(--radius-sm);
  font-size: 0.875em;
}
```

styles/base/animations.css

```
@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}
```

```
@keyframes slideInLeft {
  from {
    opacity: 0;
    transform: translateX(-50px);
  }
  to {
    opacity: 1;
    transform: translateX(0);
  }
}

@keyframes slideInRight {
  from {
    opacity: 0;
    transform: translateX(50px);
  }
  to {
    opacity: 1;
    transform: translateX(0);
  }
}

@keyframes bounce {
  0%, 20%, 53%, 80%, 100% {
    transform: translate3d(0, 0, 0);
  }
  40%, 43% {
    transform: translate3d(0, -30px, 0);
  }
  70% {
    transform: translate3d(0, -15px, 0);
  }
  90% {
    transform: translate3d(0, -4px, 0);
  }
}

.animate-fade-in {
  animation: fadeIn 0.6s ease-out;
}

.animate-slide-in-left {
  animation: slideInLeft 0.6s ease-out;
}

.animate-slide-in-right {
  animation: slideInRight 0.6s ease-out;
}

.animate-bounce {
  animation: bounce 1s ease-in-out;
}

/* Reduced motion support */
```

```
@media (prefers-reduced-motion: reduce) {  
  * {  
    animation-duration: 0.01ms !important;  
    animation-iteration-count: 1 !important;  
    transition-duration: 0.01ms !important;  
  }  
}
```

Component Styles - Component-specific CSS

styles/components/buttons.css

```
.btn {  
  display: inline-flex;  
  align-items: center;  
  justify-content: center;  
  gap: var(--spacing-sm);  
  padding: var(--spacing-sm) var(--spacing-lg);  
  border-radius: var(--radius-md);  
  font-weight: 500;  
  font-size: 0.875rem;  
  line-height: 1.25rem;  
  border: 1px solid transparent;  
  transition: all 0.2s ease;  
  cursor: pointer;  
  text-decoration: none;  
}  
  
.btn:focus {  
  outline: 2px solid rgb(var(--color-primary));  
  outline-offset: 2px;  
}  
  
.btn:disabled {  
  opacity: 0.5;  
  cursor: not-allowed;  
}  
  
.btn-primary {  
  background: rgb(var(--color-primary));  
  color: white;  
  border-color: rgb(var(--color-primary));  
}  
  
.btn-primary:hover:not(:disabled) {  
  background: rgb(37 99 235);  
  border-color: rgb(37 99 235);  
}  
  
.btn-secondary {  
  background: rgb(var(--color-bg-secondary));
```

```
    color: rgb(var(--color-text-primary));
    border-color: rgb(var(--color-border));
}

.btn-secondary:hover:not(:disabled) {
    background: rgb(243 244 246);
    border-color: rgb(209 213 219);
}

.btn-outline {
    background: transparent;
    color: rgb(var(--color-primary));
    border-color: rgb(var(--color-primary));
}

.btn-outline:hover:not(:disabled) {
    background: rgb(var(--color-primary));
    color: white;
}
```

styles/components/cards.css

```
.card {
    background: rgb(var(--color-bg-primary));
    border: 1px solid rgb(var(--color-border));
    border-radius: var(--radius-lg);
    overflow: hidden;
    transition: all 0.3s ease;
}

.card:hover {
    transform: translateY(-4px);
    box-shadow: var(--shadow-lg);
}

.card-header {
    padding: var(--spacing-lg);
    border-bottom: 1px solid rgb(var(--color-border));
}

.card-body {
    padding: var(--spacing-lg);
}

.card-footer {
    padding: var(--spacing-lg);
    border-top: 1px solid rgb(var(--color-border));
    background: rgb(var(--color-bg-secondary));
}

.project-card {
    position: relative;
```

```
    overflow: hidden;
  }

.project-card::before {
  content: '';
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  height: 4px;
  background: linear-gradient(
    90deg,
    rgb(var(--color-primary)),
    rgb(var(--color-accent))
  );
}
```

styles/components/navbar.css

```
.navbar {
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  z-index: 50;
  background: rgba(255, 255, 255, 0.8);
  backdrop-filter: blur(8px);
  border-bottom: 1px solid rgb(var(--color-border));
  transition: all 0.3s ease;
}

.navbar-scrolled {
  background: rgba(255, 255, 255, 0.95);
  box-shadow: var(--shadow-sm);
}

.navbar-container {
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: var(--spacing-md) var(--spacing-lg);
  max-width: 1200px;
  margin: 0 auto;
}

.navbar-brand {
  font-size: 1.25rem;
  font-weight: 700;
  color: rgb(var(--color-text-primary));
}

.navbar-nav {
```



```
display: flex;
gap: var(--spacing-lg);
align-items: center;
}

.navbar-link {
color: rgb(var(--color-text-secondary));
font-weight: 500;
padding: var(--spacing-sm) var(--spacing-md);
border-radius: var(--radius-md);
transition: all 0.2s ease;
}

.navbar-link:hover,
.navbar-link.active {
color: rgb(var(--color-primary));
background: rgba(59, 130, 246, 0.1);
}

/* Mobile menu */
@media (max-width: 768px) {
.navbar-nav {
position: fixed;
top: 100%;
left: 0;
right: 0;
background: white;
flex-direction: column;
padding: var(--spacing-lg);
border-top: 1px solid rgb(var(--color-border));
transform: translateY(-100%);
opacity: 0;
visibility: hidden;
transition: all 0.3s ease;
}

.navbar-nav.open {
transform: translateY(0);
opacity: 1;
visibility: visible;
}
}
```

styles/components/project-detail.css

```
.project-detail {
max-width: 1200px;
margin: 0 auto;
padding: var(--spacing-xl);
}

.project-header {
```

```
text-align: center;
margin-bottom: var(--spacing-2xl);
padding: var(--spacing-2xl) 0;
}

.project-title {
  font-size: 3rem;
  font-weight: 700;
  margin-bottom: var(--spacing-md);
  background: linear-gradient(
    135deg,
    rgb(var(--color-primary)),
    rgb(var(--color-accent))
  );
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-clip: text;
}

.project-meta {
  display: flex;
  gap: var(--spacing-lg);
  justify-content: center;
  flex-wrap: wrap;
  margin-bottom: var(--spacing-lg);
}

.project-tag {
  background: rgb(var(--color-primary));
  color: white;
  padding: var(--spacing-xs) var(--spacing-md);
  border-radius: var(--radius-full);
  font-size: 0.875rem;
  font-weight: 500;
}

.project-gallery {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  gap: var(--spacing-lg);
  margin-bottom: var(--spacing-2xl);
}

.project-image {
  border-radius: var(--radius-lg);
  overflow: hidden;
  box-shadow: var(--shadow-md);
}

.project-content {
  display: grid;
  grid-template-columns: 2fr 1fr;
  gap: var(--spacing-2xl);
  align-items: start;
```

```
}

.project-description {
  line-height: 1.8;
  font-size: 1.125rem;
}

.project-sidebar {
  position: sticky;
  top: 100px;
}

.project-links {
  display: flex;
  gap: var(--spacing-md);
  margin-top: var(--spacing-lg);
}

@media (max-width: 768px) {
  .project-content {
    grid-template-columns: 1fr;
  }

  .project-title {
    font-size: 2rem;
  }
}
```

Theme Styles - Theme-specific overrides

styles/themes/light.css

```
/* Light theme specific styles */
[data-theme='light'] {
  color-scheme: light;
}

[data-theme='light'] body {
  background: white;
  color: rgb(17 24 39);
}

[data-theme='light'] .card {
  background: white;
  border-color: rgb(229 231 235);
}
```

styles/themes/dark.css

```
/* Dark theme specific styles */
[data-theme='dark'] {
  color-scheme: dark;
}

[data-theme='dark'] body {
  background: rgb(17 24 39);
  color: rgb(249 250 251);
}

[data-theme='dark'] .card {
  background: rgb(31 41 55);
  border-color: rgb(55 65 81);
}

[data-theme='dark'] .navbar {
  background: rgba(17, 24, 39, 0.8);
  border-bottom-color: rgb(55 65 81);
}
```

Main Style Files

styles/tailwind.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

/* Import custom styles */
@import './base/variables.css';
@import './base/typography.css';
@import './base/animations.css';
@import './themes/light.css';
@import './themes/dark.css';
@import './components/buttons.css';
@import './components/cards.css';
@import './components/navbar.css';
@import './components/project-detail.css';
```

styles/index.css - Main entry point

```
/* This is the main CSS entry point */
@import './tailwind.css';

/* Additional global styles can go here */
```

Purpose Summary

Lib Directory:

- **Data:** Centralized content management for easy updates
- **Constants:** Prevents magic strings, ensures consistency
- **Utils:** Reusable functions to avoid code duplication

Styles Directory:

- **Base:** Foundational styles (variables, typography, animations)
- **Components:** Modular CSS for specific components
- **Themes:** Theme-specific overrides for light/dark modes
- **Organization:** Separates concerns, makes maintenance easier

This structure ensures your portfolio is:

- **Maintainable:** Easy to update content and styles
- **Scalable:** Can grow without becoming messy
- **Consistent:** Uniform styling and data structure
- **Performant:** Organized imports and optimized CSS
- **Themeable:** Easy to switch between light/dark themes