Theme System Documentation

1. Implementation Details

1.1 Architecture Overview

The theme system is built using a modular architecture with the following key components:

```
src/
 ├── @types/theme.ts             # Theme type definitions
 ├── assets/styles/themes.css     # CSS variables for themes
 ├── components/
|    ├── shared/ThemeSelector.tsx # Theme switching UI
|   └── template/ThemeProvider.tsx # Theme application logic
 ├── configs/theme.config.ts      # Theme configurations
 ├── store/themeStore.ts         # Theme state management
└── views/Home/themes/          # Theme-specific components
```

1.2 Core Components

1.2.1 Theme State Management

```
// themeStore.ts
type ThemeState = Theme & {
  specialty: 'default' | 'theme1' | 'theme2';
}

const useThemeStore = create<ThemeState & ThemeAction>()(
  persist(
    (set) => ({
      specialty: 'default',
      setSpecialty: (payload) => set(() => ({ specialty: payload })),
    }),
    { name: 'theme' }
  )
)
```

1.2.2 Theme Provider

```
// ThemeProvider.tsx
const ThemeProvider: React.FC<{ children: React.ReactNode }> = ({ children }) => {
  const { specialty } = useThemeStore()

  useEffect(() => {
    document.documentElement.className = `theme-${specialty}`
  }, [specialty])

  return <>{children}</>
}
```

1.2.3 Theme Configuration

```
// theme.config.ts
export const themeConfigs = {
  base: {
    colors: { /* ... */ },
    typography: { /* ... */ },
    specialtyConfig: { /* ... */ }
  },
  theme1: { /* ... */ },
  theme2: { /* ... */ }
}
```

1.3 Theme Application Flow

1. User selects a theme through ThemeSelector
2. ThemeStore updates the specialty state
3. ThemeProvider applies the new theme class
4. CSS variables update through themes.css
5. Components re-render with new theme styles

2. Theme Customization Guide

2.1 Creating a New Theme

Step 1: Define Theme Types

```
// @types/theme.ts
export interface Theme {
  id: string;
  name: string;
  colors: {
    primary: {
      main: string;
    light: string;
    dark: string;
   };
   // ... other color properties
  };
  typography: {
   fontFamily: string;
   // ... other typography properties
  };
  specialtyConfig: {
   // ... specialty-specific configurations
  };
}
```

Step 2: Add Theme Configuration

```
// configs/theme.config.ts
export const themeConfigs = {
  newTheme: {
   id: 'new-theme',
   name: 'New Theme',
   colors: {
    primary: {
      main: '#your-color',
      light: '#lighter-color',
      dark: '#darker-color'
    }
    },
    typography: {
      fontFamily: 'Your-Font'
    },
    specialtyConfig: {
      // ... specialty configurations
    }
  }
}
```

Step 3: Update Theme Store

```
// store/themeStore.ts
type ThemeState = Theme & {
  specialty: 'default' | 'theme1' | 'theme2' | 'new-theme';
}
```

2.2 Customizing Existing Themes

2.2.1 Color Customization

```
// src/views/Home/themes/[theme-name]/colors.ts
export const colors = {
  primary: {
    main: '#your-color',
    light: '#lighter-color',
    dark: '#darker-color'
  },
  background: {
    default: '#background-color',
    paper: '#paper-color'
  }
}
```

2.2.2 Typography Customization

```
// src/views/Home/themes/[theme-name]/typography.ts
export const typography = {
  fontFamily: 'Your-Font',
  h1: {
    fontSize: '2.5rem',
    fontWeight: 700
  },
  // ... other typography styles
}
```

2.3 Theme-Specific Components

2.3.1 Creating Theme-Specific Components

```
// src/views/Home/themes/[theme-name]/components/YourComponent.tsx
const YourComponent: React.FC = () => {
  const { specialty } = useThemeStore();

  return (
    <div className={`theme-${specialty}-component`}>
```

```
      {/* Component content */}
    </div>
  );
};
```

2.3.2 Styling Theme Components

```
/* src/assets/styles/themes.css */
.theme-new-theme-component {
  /* Theme-specific styles */
}
```

2.4 Best Practices

1. Color Usage
   - Use CSS variables for all theme colors
   - Maintain consistent color naming across themes
   - Ensure sufficient contrast for accessibility

2. Typography
   - Define font families in theme configuration
   - Use relative units for font sizes
   - Maintain consistent typography scale

3. Component Styling
   - Use theme-specific CSS classes
   - Implement responsive design

- Consider dark/light mode variations

4. Performance
   - Minimize theme-specific CSS
   - Use CSS variables efficiently
   - Implement smooth theme transitions

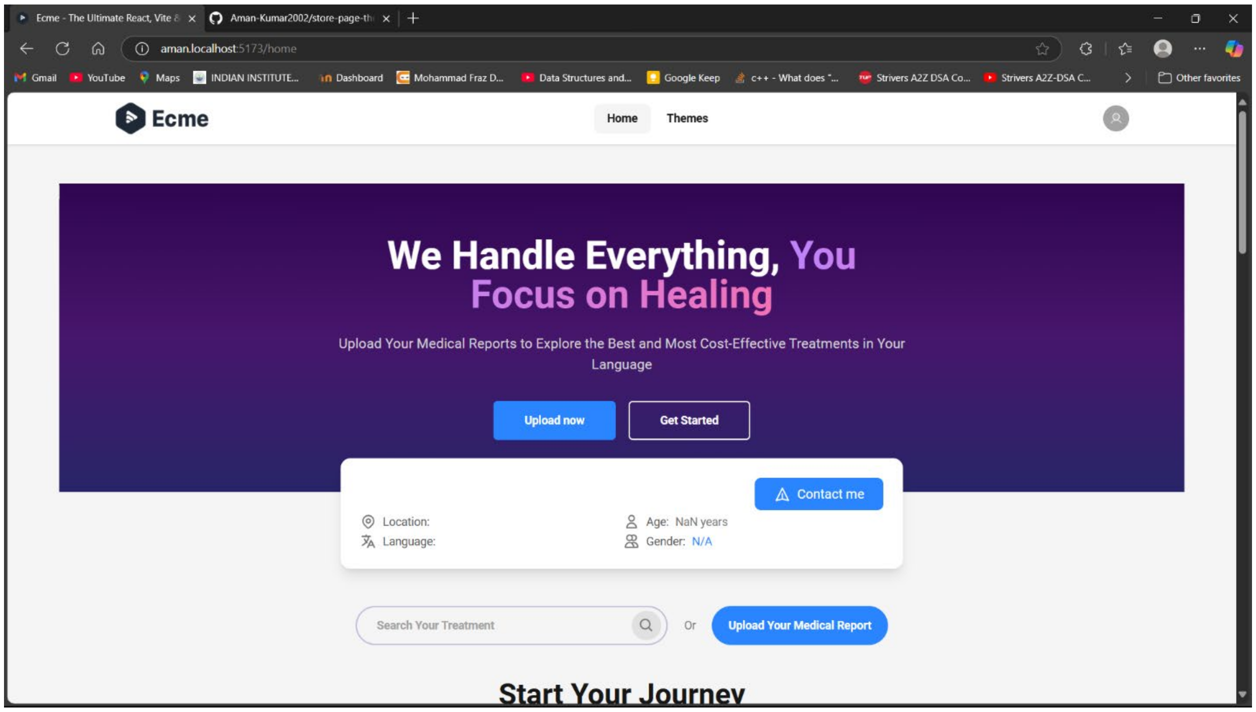
2.5 Testing Themes

1. Visual Testing
   - Test all components with each theme
   - Verify color contrast
   - Check typography rendering

2. Functionality Testing
   - Test theme switching
   - Verify theme persistence
   - Check responsive behavior

3. Accessibility Testing
   - Verify color contrast ratios
   - Check text readability

- Ensure keyboard navigation

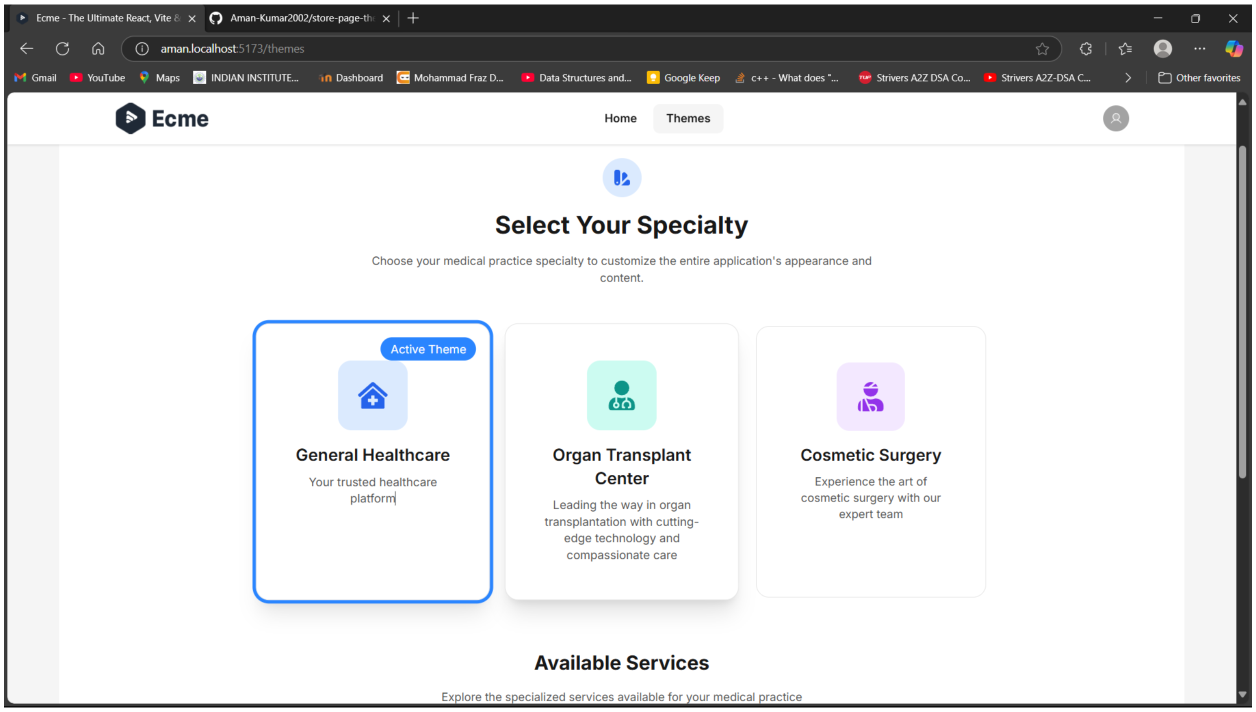


We Handle Everything, You Focus on Healing

Upload Your Medical Reports to Explore the Best and Most Cost-Effective Treatments in Your Language

Location:
Language:
Age: NaN years
Gender: N/A

Start Your Journey



Select Your Specialty

Choose your medical practice specialty to customize the entire application's appearance and content.

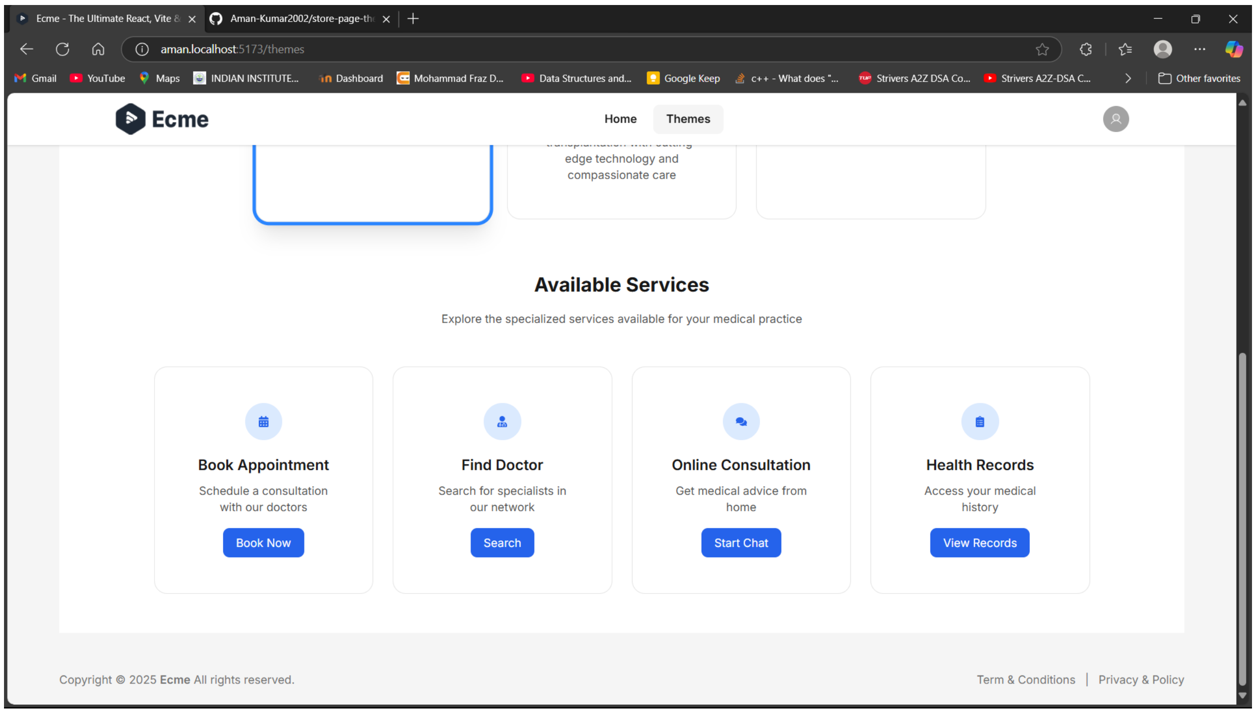
General Healthcare
Your trusted healthcare platform

Organ Transplant Center
Leading the way in organ transplantation with cutting-edge technology and compassionate care

Cosmetic Surgery
Experience the art of cosmetic surgery with our expert team

Available Services

Explore the specialized services available for your medical practice



Ecme Home Themes

edge technology and compassionate care

Available Services

Explore the specialized services available for your medical practice

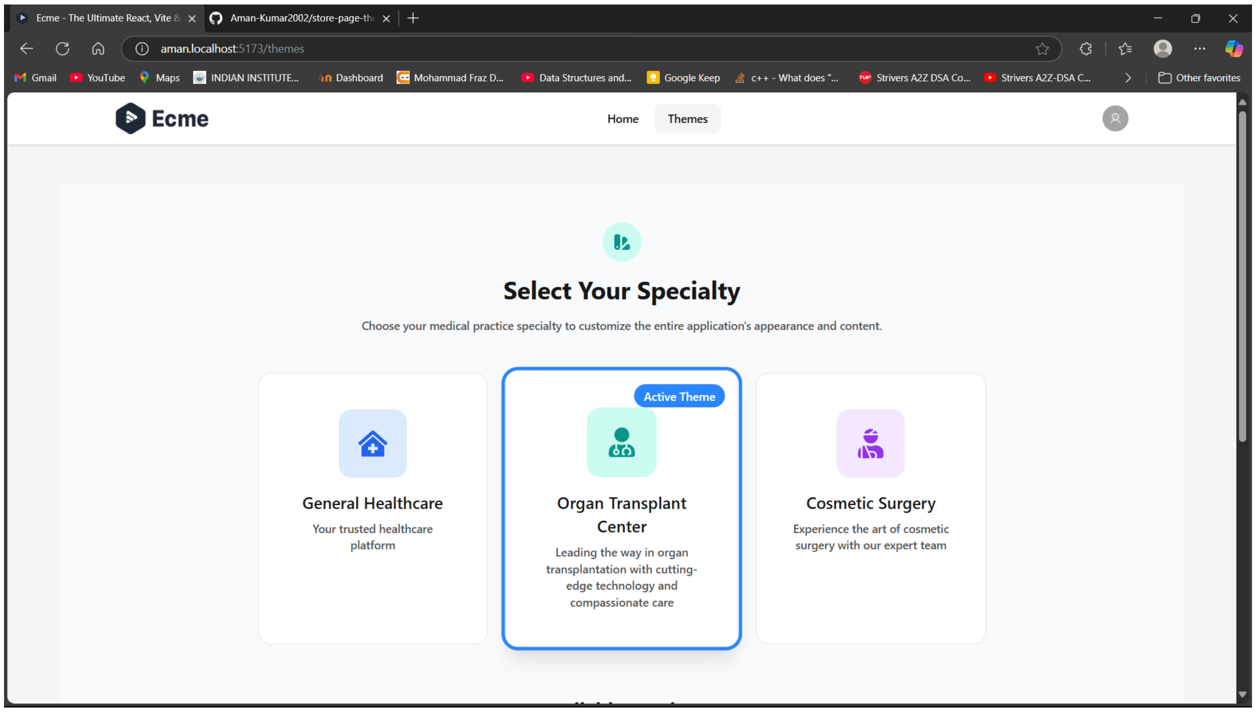
Book Appointment
Schedule a consultation with our doctors
Book Now

Find Doctor
Search for specialists in our network
Search

Online Consultation
Get medical advice from home
Start Chat

Health Records
Access your medical history
View Records

Copyright © 2025 Ecme All rights reserved. Term & Conditions | Privacy & Policy



Ecme Home Themes

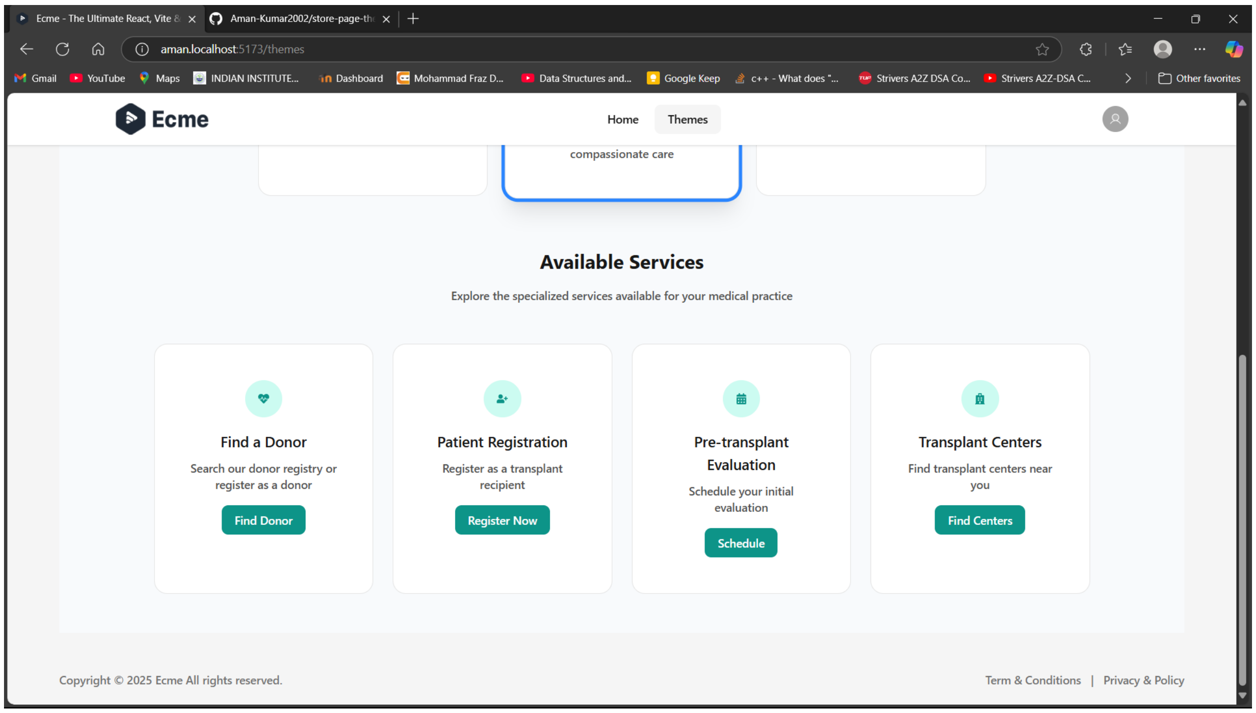
Select Your Specialty

Choose your medical practice specialty to customize the entire application's appearance and content.

General Healthcare
Your trusted healthcare platform

Active Theme
Organ Transplant Center
Leading the way in organ transplantation with cutting-edge technology and compassionate care

Cosmetic Surgery
Experience the art of cosmetic surgery with our expert team



compassionate care

Available Services

Explore the specialized services available for your medical practice

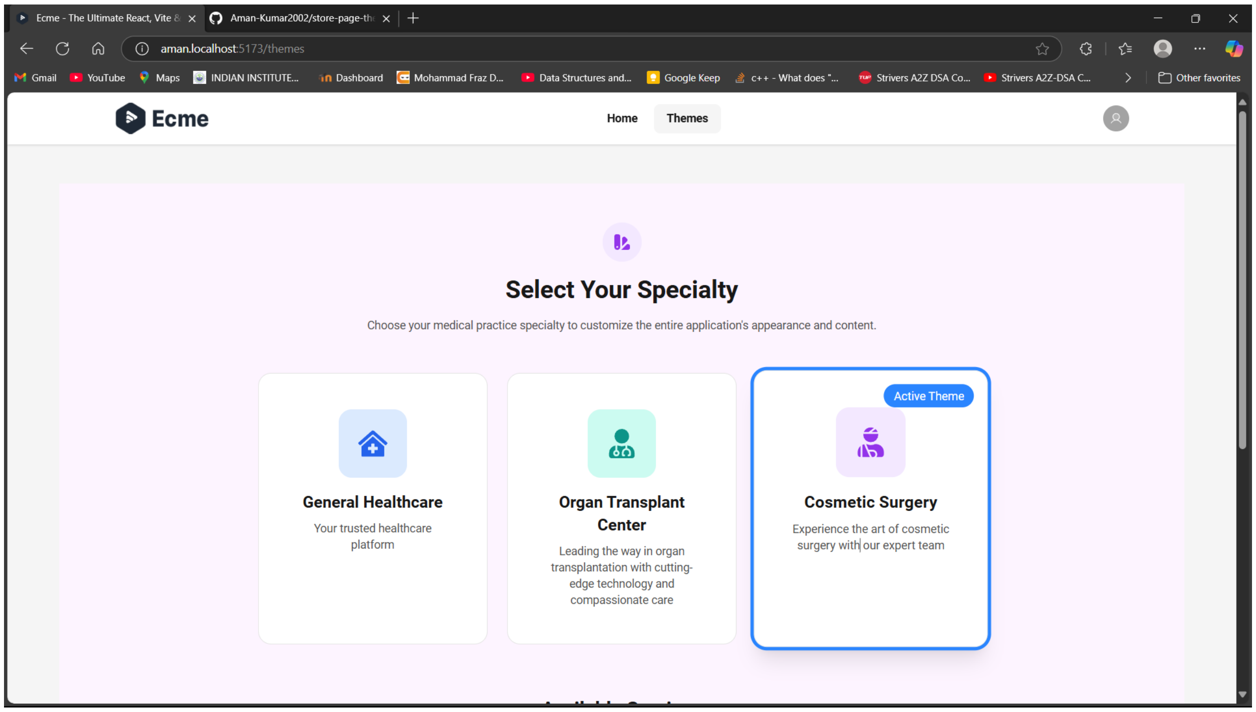
Find a Donor
Search our donor registry or register as a donor
Find Donor

Patient Registration
Register as a transplant recipient
Register Now

Pre-transplant Evaluation
Schedule your initial evaluation
Schedule

Transplant Centers
Find transplant centers near you
Find Centers

Copyright © 2025 Ecme All rights reserved.
Term & Conditions | Privacy & Policy



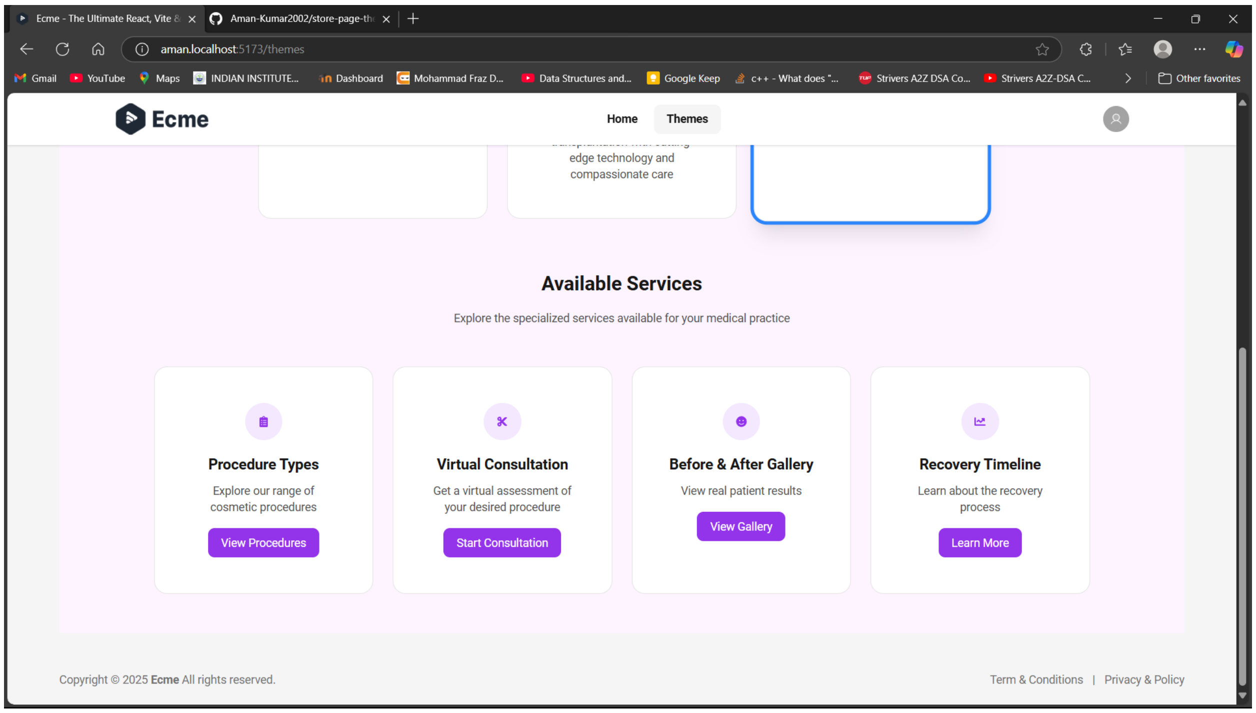
Select Your Specialty

Choose your medical practice specialty to customize the entire application's appearance and content.

General Healthcare
Your trusted healthcare platform

Organ Transplant Center
Leading the way in organ transplantation with cutting-edge technology and compassionate care

Active Theme
Cosmetic Surgery
Experience the art of cosmetic surgery with our expert team



edge technology and compassionate care

Available Services

Explore the specialized services available for your medical practice

Procedure Types
Explore our range of cosmetic procedures
View Procedures

Virtual Consultation
Get a virtual assessment of your desired procedure
Start Consultation

Before & After Gallery
View real patient results
View Gallery

Recovery Timeline
Learn about the recovery process
Learn More

Copyright © 2025 **Ecme** All rights reserved.

Term & Conditions | Privacy & Policy