

Platform: **Mobile App.**

Tool to use: **Figma**

Description:

Build UI/UX solution for mobile application to control the home appliance like Fan, Air

conditioner, TV, Lights & Bulbs, Lights with adjustable brightness.

Direct Output Link: <https://snuff-flex-67532926.figma.site/>

Code:

App.tsx

```
import React, { useState } from 'react';
import { DeviceCard } from './components/DeviceCard';
import { StatusBadge } from './components/StatusBadge';
import { Fan, Thermometer, Tv, Lightbulb, Home, Zap, Grid2X2, List } from
' lucide-react';
import { motion } from 'motion/react';
import { Button } from './components/ui/button';

import backgroundImage from
'figma:asset/11deb0a61998a7fb9215c208452b925466b28ffe.png';

interface Appliance {
  id: string;
  name: string;
  type: 'fan' | 'ac' | 'tv' | 'light';
  icon: React.ReactNode;
```

```
isOn: boolean;
settings: {
  speed?: number;
  temperature?: number;
  brightness?: number;
  color?: string;
};
powerConsumption: number; // Watts
}
```

```
export default function App() {
  const [viewMode, setViewMode] = useState<'grid' | 'list'>('list');
  const [appliances, setAppliances] = useState<Appliance[]>([
    {
      id: '1',
      name: 'Living Room Fan',
      type: 'fan',
      icon: <Fan className="w-6 h-6 text-white" />,
      isOn: true,
      settings: { speed: 3 },
      powerConsumption: 75
    },
    {
      id: '2',
      name: 'Bedroom AC',
      type: 'ac',
      icon: <Thermometer className="w-6 h-6 text-white" />,
      isOn: false,
      settings: { temperature: 24 },
      powerConsumption: 1500
    },
    {
      id: '3',
      name: 'Living Room TV',
      type: 'tv',
```

```
    icon: <Tv className="w-6 h-6 text-white" />,
    isOn: false,
    settings: {},
    powerConsumption: 120
  },
  {
    id: '4',
    name: 'Bedroom Light',
    type: 'light',
    icon: <Lightbulb className="w-6 h-6 text-white" />,
    isOn: false,
    settings: { brightness: 80, color: '#ffffff' },
    powerConsumption: 60
  },
  {
    id: '5',
    name: 'Kitchen Light',
    type: 'light',
    icon: <Lightbulb className="w-6 h-6 text-white" />,
    isOn: false,
    settings: { brightness: 60, color: '#fff4e6' },
    powerConsumption: 40
  },
  {
    id: '6',
    name: 'Study Room Fan',
    type: 'fan',
    icon: <Fan className="w-6 h-6 text-white" />,
    isOn: false,
    settings: { speed: 2 },
    powerConsumption: 65
  }
]);
```

```
const toggleAppliance = (id: string) => {
```

```

    setAppliances(prev =>
      prev.map(appliance =>
        appliance.id === id
          ? { ...appliance, isOn: !appliance.isOn }
          : appliance
      )
    );
  };

```

```

const updateSettings = (id: string, newSettings: Partial<Appliance['settings']>)
=> {
  setAppliances(prev =>
    prev.map(appliance =>
      appliance.id === id
        ? { ...appliance, settings: { ...appliance.settings, ...newSettings } }
        : appliance
    )
  );
};

```

```

// Calculate statistics
const activeCount = appliances.filter(a => a.isOn).length;
const totalPowerConsumption = appliances
  .filter(a => a.isOn)
  .reduce((sum, a) => sum + a.powerConsumption, 0);

```

```

const getCategoryCounts = (type: string) => {
  const devices = appliances.filter(a => a.type === type);
  return {
    active: devices.filter(d => d.isOn).length,
    total: devices.length
  };
};

```

```

const fanCounts = getCategoryCounts('fan');

```

```

const acCounts = getCategoryCounts('ac');
const tvCounts = getCategoryCounts('tv');
const lightCounts = getCategoryCounts('light');

return (
  <div className="min-h-screen relative overflow-hidden">
    {/* Background Image */}
    <div
      className="fixed inset-0 bg-cover bg-center bg-no-repeat"
      style={{
        backgroundImage: `url(${backgroundImage})`,
      }}
    />

    {/* Overlay for better contrast */}
    <div className="absolute inset-0 bg-black/40" />

    {/* Glassmorphism Overlay */}
    <div className="absolute inset-0 bg-gradient-to-br from-white/5 via-transparent to-black/10" />

    {/* Content */}
    <div className="relative z-10 max-w-6xl mx-auto p-6">
      {/* Header */}
      <motion.div
        className="text-center mb-8"
        initial={{ opacity: 0, y: -20 }}
        animate={{ opacity: 1, y: 0 }}
        transition={{ duration: 0.5 }}
      >
        <div className="inline-flex items-center justify-center w-16 h-16 bg-cyan-500/20 backdrop-blur-xl rounded-2xl mb-4 border border-cyan-400/30">
          <Home className="w-8 h-8 text-cyan-400" />
        </div>

```

```
<h1 className="text-3xl font-medium text-white mb-6">Smart Home  
Command Center</h1>
```

```
  { /* Status Indicators */ }  
  <div className="flex items-center justify-center gap-6 mb-8">  
    <div className="flex items-center gap-2">  
      <div className="w-2 h-2 bg-green-400 rounded-full animate-  
pulse"></div>  
      <span className="text-green-400 font-medium">{activeCount}  
Active</span>  
    </div>  
    <div className="flex items-center gap-2">  
      <Zap className="w-4 h-4 text-yellow-400" />  
      <span className="text-yellow-400 font-  
medium">{totalPowerConsumption}W</span>  
    </div>  
  </div>
```

```
  { /* Category Summary Badges */ }  
  <div className="flex flex-wrap items-center justify-center gap-4">  
    <StatusBadge  
      icon={<Fan className="w-4 h-4 text-white" />}  
      label="Fans"  
      active={fanCounts.active}  
      total={fanCounts.total}  
      color="bg-cyan-500/30"  
    />  
    <StatusBadge  
      icon={<Thermometer className="w-4 h-4 text-white" />}  
      label="AC Units"  
      active={acCounts.active}  
      total={acCounts.total}  
      color="bg-blue-500/30"  
    />  
    <StatusBadge
```

```

    icon={<Tv className="w-4 h-4 text-white" />}
    label="TVs"
    active={tvCounts.active}
    total={tvCounts.total}
    color="bg-purple-500/30"
  />
  <StatusBadge
    icon={<Lightbulb className="w-4 h-4 text-white" />}
    label="Lights"
    active={lightCounts.active}
    total={lightCounts.total}
    color="bg-yellow-500/30"
  />
</div>
</motion.div>

{/* Connected Devices Section */}
<motion.div
  className="bg-white/10 backdrop-blur-xl rounded-3xl border border-
white/20 shadow-xl p-6 mb-6"
  initial={{ opacity: 0, y: 20 }}
  animate={{ opacity: 1, y: 0 }}
  transition={{ duration: 0.5, delay: 0.2 }}
>
  <div className="flex items-center justify-between mb-6">
    <div>
      <h2 className="text-xl font-medium text-white">Connected
Devices</h2>
      <p className="text-white/70">
        {appliances.length} devices • {activeCount} active •
        {totalPowerConsumption}W consumption
      </p>
    </div>

```

```
<div className="flex items-center gap-2 bg-white/10 backdrop-blur-xl
p-1 rounded-xl border border-white/20">
```

```
<Button
  variant={viewMode === 'grid' ? 'default' : 'ghost'}
  size="sm"
  onClick={() => setViewMode('grid')}
  className={` ${
    viewMode === 'grid'
      ? 'bg-cyan-500 text-white shadow-lg'
      : 'text-white/70 hover:text-white hover:bg-white/10'
  } rounded-lg transition-all duration-200`}
>
```

```
<Grid2X2 className="w-4 h-4" />
```

```
</Button>
```

```
<Button
  variant={viewMode === 'list' ? 'default' : 'ghost'}
  size="sm"
  onClick={() => setViewMode('list')}
  className={` ${
    viewMode === 'list'
      ? 'bg-cyan-500 text-white shadow-lg'
      : 'text-white/70 hover:text-white hover:bg-white/10'
  } rounded-lg transition-all duration-200`}
>
```

```
<List className="w-4 h-4" />
```

```
</Button>
```

```
</div>
```

```
</div>
```

```
{/* Device Cards */}
```

```
<div className={` ${
  viewMode === 'grid'
    ? 'grid grid-cols-1 md:grid-cols-2 xl:grid-cols-3 gap-4'
    : 'space-y-4'
  } `}>
```



```

    {appliances.map((appliance, index) => (
      <motion.div
        key={appliance.id}
        initial={{ opacity: 0, y: 20 }}
        animate={{ opacity: 1, y: 0 }}
        transition={{ duration: 0.3, delay: index * 0.1 }}
      >
        <DeviceCard
          appliance={appliance}
          onToggle={() => toggleAppliance(appliance.id)}
          onUpdateSettings={(settings) => updateSettings(appliance.id,
settings)}
        />
      </motion.div>
    ))}
  </div>
</motion.div>

{/* System Control */}
<motion.div
  className="bg-white/10 backdrop-blur-xl rounded-3xl border border-
white/20 shadow-xl p-6"
  initial={{ opacity: 0, y: 20 }}
  animate={{ opacity: 1, y: 0 }}
  transition={{ duration: 0.5, delay: 0.4 }}
  >
    <h3 className="text-lg font-medium text-white mb-4">System
Control</h3>
    <div className="flex gap-3 flex-wrap">
      <Button
        className="bg-green-500/20 hover:bg-green-500/30 text-green-400
border border-green-400/30"
        onClick={() => setAppliances(prev => prev.map(a => ({ ...a, isOn: true
}}))}
      >

```

```

      Turn All On
    </Button>
    <Button
      className="bg-red-500/20 hover:bg-red-500/30 text-red-400 border
border-red-400/30"
      onClick={() => setAppliances(prev => prev.map(a => ({ ...a, isOn: false
})))}
    >
      Turn All Off
    </Button>
    <Button
      className="bg-yellow-500/20 hover:bg-yellow-500/30 text-yellow-400
border border-yellow-400/30"
      onClick={() => {
        const lightIds = appliances.filter(a => a.type === 'light').map(a => a.id);
        setAppliances(prev => prev.map(a =>
          lightIds.includes(a.id) ? { ...a, isOn: true } : a
        ));
      }}
    >
      Lights Only
    </Button>
  </div>
</motion.div>
</div>
</div>
);
}

```

Globals.css

```
@custom-variant dark (&:is(.dark *));
```

```
:root {
  --font-size: 14px;
  --background: #ffffff;
  --foreground: oklch(0.145 0 0);
  --card: #ffffff;
  --card-foreground: oklch(0.145 0 0);
  --popover: oklch(1 0 0);
  --popover-foreground: oklch(0.145 0 0);
  --primary: #030213;
  --primary-foreground: oklch(1 0 0);
  --secondary: oklch(0.95 0.0058 264.53);
  --secondary-foreground: #030213;
  --muted: #ecec0;
  --muted-foreground: #717182;
  --accent: #e9ebef;
  --accent-foreground: #030213;
  --destructive: #d4183d;
  --destructive-foreground: #ffffff;
  --border: rgba(0, 0, 0, 0.1);
  --input: transparent;
  --input-background: #f3f3f5;
  --switch-background: #cbced4;
  --font-weight-medium: 500;
  --font-weight-normal: 400;
  --ring: oklch(0.708 0 0);
  --chart-1: oklch(0.646 0.222 41.116);
  --chart-2: oklch(0.6 0.118 184.704);
  --chart-3: oklch(0.398 0.07 227.392);
  --chart-4: oklch(0.828 0.189 84.429);
  --chart-5: oklch(0.769 0.188 70.08);
  --radius: 0.625rem;
  --sidebar: oklch(0.985 0 0);
  --sidebar-foreground: oklch(0.145 0 0);
  --sidebar-primary: #030213;
  --sidebar-primary-foreground: oklch(0.985 0 0);
}
```

```
--sidebar-accent: oklch(0.97 0 0);
--sidebar-accent-foreground: oklch(0.205 0 0);
--sidebar-border: oklch(0.922 0 0);
--sidebar-ring: oklch(0.708 0 0);
}

.dark {
  --background: oklch(0.145 0 0);
  --foreground: oklch(0.985 0 0);
  --card: oklch(0.145 0 0);
  --card-foreground: oklch(0.985 0 0);
  --popover: oklch(0.145 0 0);
  --popover-foreground: oklch(0.985 0 0);
  --primary: oklch(0.985 0 0);
  --primary-foreground: oklch(0.205 0 0);
  --secondary: oklch(0.269 0 0);
  --secondary-foreground: oklch(0.985 0 0);
  --muted: oklch(0.269 0 0);
  --muted-foreground: oklch(0.708 0 0);
  --accent: oklch(0.269 0 0);
  --accent-foreground: oklch(0.985 0 0);
  --destructive: oklch(0.396 0.141 25.723);
  --destructive-foreground: oklch(0.637 0.237 25.331);
  --border: oklch(0.269 0 0);
  --input: oklch(0.269 0 0);
  --ring: oklch(0.439 0 0);
  --font-weight-medium: 500;
  --font-weight-normal: 400;
  --chart-1: oklch(0.488 0.243 264.376);
  --chart-2: oklch(0.696 0.17 162.48);
  --chart-3: oklch(0.769 0.188 70.08);
  --chart-4: oklch(0.627 0.265 303.9);
  --chart-5: oklch(0.645 0.246 16.439);
  --sidebar: oklch(0.205 0 0);
  --sidebar-foreground: oklch(0.985 0 0);
```

```
--sidebar-primary: oklch(0.488 0.243 264.376);
--sidebar-primary-foreground: oklch(0.985 0 0);
--sidebar-accent: oklch(0.269 0 0);
--sidebar-accent-foreground: oklch(0.985 0 0);
--sidebar-border: oklch(0.269 0 0);
--sidebar-ring: oklch(0.439 0 0);
}
```

```
@theme inline {
  --color-background: var(--background);
  --color-foreground: var(--foreground);
  --color-card: var(--card);
  --color-card-foreground: var(--card-foreground);
  --color-popover: var(--popover);
  --color-popover-foreground: var(--popover-foreground);
  --color-primary: var(--primary);
  --color-primary-foreground: var(--primary-foreground);
  --color-secondary: var(--secondary);
  --color-secondary-foreground: var(--secondary-foreground);
  --color-muted: var(--muted);
  --color-muted-foreground: var(--muted-foreground);
  --color-accent: var(--accent);
  --color-accent-foreground: var(--accent-foreground);
  --color-destructive: var(--destructive);
  --color-destructive-foreground: var(--destructive-foreground);
  --color-border: var(--border);
  --color-input: var(--input);
  --color-input-background: var(--input-background);
  --color-switch-background: var(--switch-background);
  --color-ring: var(--ring);
  --color-chart-1: var(--chart-1);
  --color-chart-2: var(--chart-2);
  --color-chart-3: var(--chart-3);
  --color-chart-4: var(--chart-4);
  --color-chart-5: var(--chart-5);
}
```

```

--radius-sm: calc(var(--radius) - 4px);
--radius-md: calc(var(--radius) - 2px);
--radius-lg: var(--radius);
--radius-xl: calc(var(--radius) + 4px);
--color-sidebar: var(--sidebar);
--color-sidebar-foreground: var(--sidebar-foreground);
--color-sidebar-primary: var(--sidebar-primary);
--color-sidebar-primary-foreground: var(--sidebar-primary-foreground);
--color-sidebar-accent: var(--sidebar-accent);
--color-sidebar-accent-foreground: var(--sidebar-accent-foreground);
--color-sidebar-border: var(--sidebar-border);
--color-sidebar-ring: var(--sidebar-ring);
}

```

```

@layer base {
  * {
    @apply border-border outline-ring/50;
  }
}

```

```

body {
  @apply bg-background text-foreground;
}
}

```

```

/**
 * Base typography. This is not applied to elements which have an ancestor
 * with a Tailwind text class.
 */

```

```

@layer base {
  :where(:not(:has([class*=" text-"]), :not(:has([class^="text-"])))) {
    h1 {
      font-size: var(--text-2xl);
      font-weight: var(--font-weight-medium);
      line-height: 1.5;
    }
  }
}

```

```
h2 {  
  font-size: var(--text-xl);  
  font-weight: var(--font-weight-medium);  
  line-height: 1.5;  
}
```

```
h3 {  
  font-size: var(--text-lg);  
  font-weight: var(--font-weight-medium);  
  line-height: 1.5;  
}
```

```
h4 {  
  font-size: var(--text-base);  
  font-weight: var(--font-weight-medium);  
  line-height: 1.5;  
}
```

```
p {  
  font-size: var(--text-base);  
  font-weight: var(--font-weight-normal);  
  line-height: 1.5;  
}
```

```
label {  
  font-size: var(--text-base);  
  font-weight: var(--font-weight-medium);  
  line-height: 1.5;  
}
```

```
button {  
  font-size: var(--text-base);  
  font-weight: var(--font-weight-medium);  
  line-height: 1.5;
```

```

}

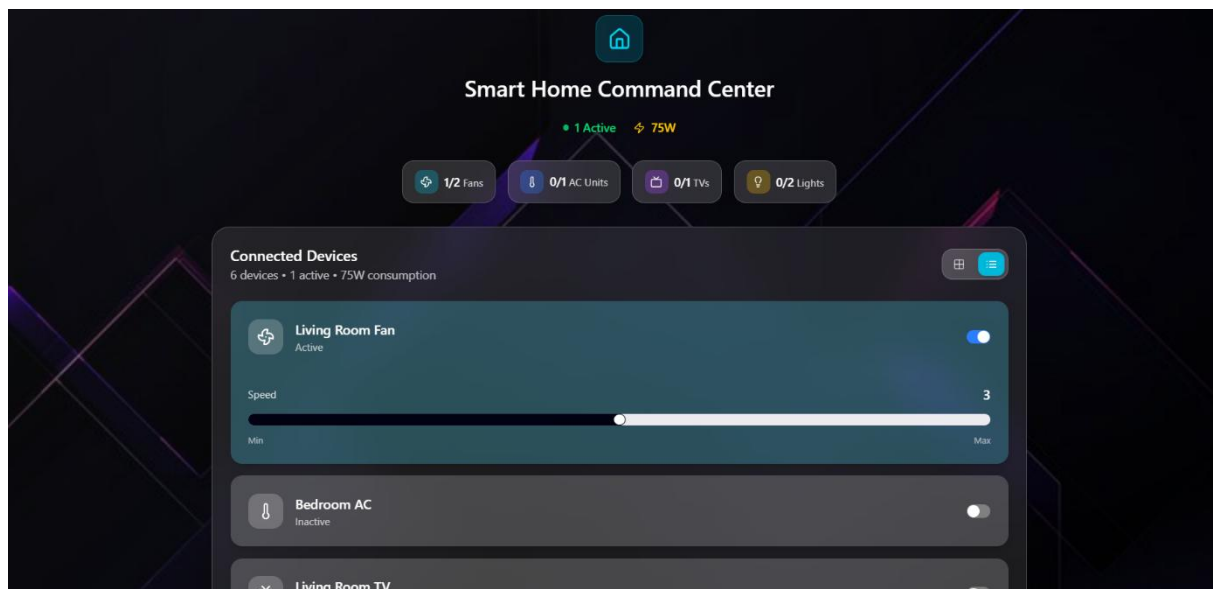
input {
  font-size: var(--text-base);
  font-weight: var(--font-weight-normal);
  line-height: 1.5;
}
}
}

html {
  font-size: var(--font-size);
}

```

OUTPUT screenshots:

1.



2.

