

Experiment 1

1. Aim:

Build Responsive and Interactive UIs using Tailwind CSS

2. Theory:

a) Prerequisites of the Project:

Before starting this project, some prerequisites were necessary to set up the development environment:

1. **Basic knowledge of HTML, CSS, and JavaScript** – for structuring, styling, and adding interactivity.
1. **Tailwind CSS CDN/Setup** – Tailwind was included directly in the HTML using its CDN link.
2. **Code Editor (VS Code)** – used for writing and organizing project files.
3. **Web Browser** – for running and testing responsiveness.
4. **Basic understanding of responsive design** – so that the UI adapts across mobile, tablet, and desktop devices.

b) About Tailwind CSS:

Tailwind CSS is a modern utility-first CSS framework that allows developers to build responsive and customizable user interfaces rapidly. Unlike traditional CSS frameworks that provide predefined components, Tailwind offers low-level utility classes such as `flex`, `p-4`, `bg-blue-500`, and `text-center`. These classes can be combined to create unique, scalable designs without writing extra CSS. Tailwind also supports responsive design through built-in breakpoints and includes features like hover states, dark mode, and custom configuration, making it a powerful tool for building modern web applications.

c) Tailwind CSS in this Project:

In this project, Tailwind CSS was used extensively to design a personal portfolio website. Sections like **navigation**, **hero banner**, **about**, **projects**, and **contact form** were styled using Tailwind utility classes. For example, the **hero section buttons** were styled with Tailwind's ready-to-use classes to add colors, padding, rounded corners, and hover effects. This avoided writing long CSS rules and directly applied modern UI styling in the HTML itself.

Example code used in project (Hero Section Button):

```
<button class="bg-white text-blue-600 px-8 py-4 rounded-lg text-lg font-bold  
  hover:bg-blue-50 transition-all duration-300 transform hover:scale-105">  
  View Projects  
</button>
```

This single line of Tailwind classes makes the button **white with blue text**, adds **padding**, **rounded corners**, **font styling**, and even a **hover animation that scales the button smoothly**.

d) Additional 30% Extra Features Implemented:

Beyond the basic requirements, about **30% extra functionality** was added to enhance interactivity. JavaScript ([script.js](#)) introduced features such as a **chatbot**, **voice commands**, **scroll progress indicator**, **dark mode toggle**, **animated skill bars**, **floating action button**, and even a small **progress game** where users can level up by earning experience points. Extra UI components such as **project filters**, **testimonials slider**, **quick settings panel**, and **scrollspy navigation dots** were also integrated. These features go beyond static UI, making the portfolio more engaging and interactive while showcasing the flexibility of Tailwind CSS when combined with JavaScript.

e) Detailed Explanation of Extra Features:

The extra features added to the project made it much more advanced and interactive compared to a simple static portfolio. The **Chatbot** allows visitors to interact and get details about skills, projects, and contact info. **Voice commands** enable hands-free navigation, such as saying “scroll to top” or “dark mode.” The **scroll progress indicator** visually shows how much of the page has been read. A **dark mode toggle** was implemented to improve accessibility and user preference. **Animated skill bars and counters** bring data to life by showing skills and achievements dynamically. The **floating action button** provides quick navigation to the top of the page. Gamification was introduced through the **Progress Game**, where users can gain XP and level up. Additionally, **project filters**, **testimonials slider**, **quick settings panel (with themes and print option)**, and **scrollspy navigation dots** were integrated to make the portfolio look more professional and interactive.

3. CODE of VSCODE:-

a) Tailwind config (custom colors, animations, keyframes):

```
<script>
tailwind.config = {
  theme: {
    extend: {
      colors: {
        primary: '#3B82F6',
        secondary: '#1E40AF',
        accent: '#F59E0B',
        dark: '#1F2937',
        light: '#F8FAFC'
      },
      animation: {
        'pulse-slow': 'pulse 3s cubic-bezier(0.4, 0, 0.6, 1) infinite',
        'bounce-slow': 'bounce 2s infinite',
        'spin-slow': 'spin 8s linear infinite',
        'float': 'float 6s ease-in-out infinite',
        'wiggle': 'wiggle 1s ease-in-out infinite',
        'slide-in': 'slideIn 0.5s ease-out',
        'rotate-3d': 'rotate3d 20s linear infinite',
        'shimmer': 'shimmer 2s linear infinite',
        'typewriter': 'typewriter 3s steps(40) 1s forwards',
        'fade-in-up': 'fadeInUp 0.8s ease-out'
      },
      keyframes: {
        float: {
          '0%, 100%': { transform: 'translateY(0px)' },
          '50%': { transform: 'translateY(-20px)' }
        },
        wiggle: {
          '0%, 100%': { transform: 'rotate(-3deg)' },
          '50%': { transform: 'rotate(3deg)' }
        },
        slideIn: {
          '0%': { transform: 'translateX(-100%)', opacity: '0' },
          '100%': { transform: 'translateX(0)', opacity: '1' }
        },
        rotate3d: {
          '0%': { transform: 'rotateY(0deg)' },
          '100%': { transform: 'rotateY(360deg)' }
        },
        shimmer: {
          '0%': { backgroundImage: '-200% 0' },
          '100%': { backgroundImage: '200% 0' }
        },
        typewriter: {
          '0%': { width: '0' },
          '100%': { width: '100%' }
        },
        fadeInUp: {
          '0%': { transform: 'translateY(30px)', opacity: '0' },
          '100%': { transform: 'translateY(0)', opacity: '1' }
        }
      }
    }
  }
}
</script>
<style>
```

b) Core inline styles (gradient, glass, timeline, skill-bar, project hover, indicators)

```
<style>
.gradient-bg {
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
}
.glass-card {
  background: rgba(255, 255, 255, 0.1);
  backdrop-filter: blur(10px);
  border: 1px solid rgba(255, 255, 255, 0.2);
}
.card-hover {
  transition: all 0.3s ease;
}
.card-hover:hover {
  transform: translateY(-10px);
  box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);
}
.timeline-item {
  position: relative;
  padding-left: 30px;
}
.timeline-item:before {
  content: '';
  position: absolute;
  left: 0;
  top: 0;
  width: 12px;
  height: 12px;
  background: #3B82F6;
  border-radius: 50%;
  box-shadow: 0 0 10px #3B82F6;
}
.timeline-item:after {
  content: '';
  position: absolute;
  left: 5px;
  top: 0;
  width: 12px;
  height: 12px;
  background: #3B82F6;
  border-radius: 50%;
  box-shadow: 0 0 10px #3B82F6;
}
.timeline-item:after {
  content: '';
  position: absolute;
  left: 5px;
  top: 12px;
  width: 2px;
  height: calc(100% + 20px);
  background: linear-gradient(to bottom, #3B82F6, transparent);
}
.skill-bar {
  background: linear-gradient(90deg, #3B82F6, #1E40AF);
  background-size: 200% 100%;
  animation: shimmer 2s linear infinite;
}
.project-card {
  transform-style: preserve-3d;
  transition: transform 0.6s;
}
.project-card:hover {
  transform: rotateY(10deg) rotateX(5deg);
}
.floating-btn {
  position: fixed;
  bottom: 30px;
  right: 30px;
  z-index: 1000;
}
```

```

65     <style>
123     }
124     .scroll-indicator {
125         position: fixed;
126         top: 0;
127         left: 0;
128         width: 0%;
129         height: 3px;
130         background: linear-gradient(90deg, #3B82F6, #1E40AF);
131         z-index: 1001;
132         transition: width 0.1s ease;
133     }
134     .notification {
135         position: fixed;
136         top: 20px;
137         right: 20px;
138         z-index: 1002;
139         animation: slideIn 0.5s ease-out;
140     }
141     .interactive-bg {
142         background:
143             radial-gradient(circle at 20% 80%, rgba(59, 130, 246, 0.1) 0%, transparent 50%),
144             radial-gradient(circle at 80% 20%, rgba(30, 64, 175, 0.1) 0%, transparent 50%),
145             radial-gradient(circle at 40% 40%, rgba(245, 158, 11, 0.1) 0%, transparent 50%);
146     }
147     .counter {
148         transition: all 0.3s ease;
149         display: inline-block;
150     }
151     .counter.animate {
152         transform: scale(1.1);
153         color: #3B82F6;
154     }
155     </style>
156 </head>

```

c) Hero section

```

<!-- Hero Section -->
<section id="home" class="gradient-bg relative py-20">
  <div class="max-w-6xl mx-auto px-4 text-center text-white">
    <div class="mb-8">
      <h1 class="text-6xl md:text-8xl font-bold mb-6 animate-pulse-slow">
        ATHARVA
      </h1>
      <h2 class="text-4xl md:text-6xl font-bold mb-6 animate-pulse-slow" style="animation-delay: 1s;">
        CHAUDHARI
      </h2>
    </div>
    <p class="text-xl md:text-2xl mb-12 max-w-3xl mx-auto animate-fade-in-up">
      Full Stack Developer | AI/ML Enthusiast | Problem Solver
    </p>
    <div class="flex flex-col sm:flex-row gap-6 justify-center">
      <button class="bg-white text-blue-600 px-8 py-4 rounded-lg text-lg font-bold hover:bg-blue-50 transition-all duration-300">
        View Projects
      </button>
      <button class="border-2 border-white text-white px-8 py-4 rounded-lg text-lg font-bold hover:bg-white hover:text-blue-600">
        Download Resume
      </button>
    </div>
  </div>
</section>

```

d) Stats counters (animated numbers)

```
<!-- Stats Section -->
<section class="py-20 gradient-bg">
  <div class="max-w-6xl mx-auto px-4">
    <div class="grid grid-cols-2 md:grid-cols-4 gap-8 text-center">
      <div class="glass-card p-6 rounded-xl animate-fade-in-up">
        <div class="text-4xl font-bold text-white mb-2 counter" data-target="3">0</div>
        <div class="text-gray-200">Internships</div>
      </div>
      <div class="glass-card p-6 rounded-xl animate-fade-in-up" style="animation-delay: 0.1s;">
        <div class="text-4xl font-bold text-white mb-2 counter" data-target="3">0</div>
        <div class="text-gray-200">Projects</div>
      </div>
      <div class="glass-card p-6 rounded-xl animate-fade-in-up" style="animation-delay: 0.2s;">
        <div class="text-4xl font-bold text-white mb-2 counter" data-target="1">0</div>
        <div class="text-gray-200">Hackathon</div>
      </div>
      <div class="glass-card p-6 rounded-xl animate-fade-in-up" style="animation-delay: 0.3s;">
        <div class="text-4xl font-bold text-white mb-2 counter" data-target="98">0</div>
        <div class="text-gray-200">Success Rate %</div>
      </div>
    </div>
  </div>
</section>
```

e) Live ChatBot Section

```
<!-- Live Chat Bot Section -->
<section class="py-20 px-4 bg-gradient-to-r from-purple-600 to-blue-600">
  <div class="max-w-4xl mx-auto">
    <h2 class="text-4xl font-bold text-center mb-16 animate-fade-in-up text-white">Chat with AtharvaBot 🤖</h2>
    <div class="glass-card p-8 rounded-xl card-hover animate-fade-in-up">
      <div id="chatContainer" class="h-96 bg-white rounded-lg p-4 mb-4 overflow-y-auto">
        <div class="flex items-start space-x-3 mb-4">
          <div class="w-8 h-8 bg-blue-500 rounded-full flex items-center justify-center text-white text-sm">🤖</div>
          <div class="bg-gray-100 p-3 rounded-lg max-w-xs">
            <p class="text-gray-800">Hi! I'm AtharvaBot. Ask me anything about Atharva's skills, projects, or experience!</p>
          </div>
        </div>
      </div>
      <div class="flex space-x-2">
        <input type="text" id="chatInput" placeholder="Type your message..." class="flex-1 p-3 border border-gray-300 rounded-lg focus">
        <button id="sendChat" class="bg-blue-600 text-white px-6 py-3 rounded-lg hover:bg-blue-700 transition-colors">Send</button>
      </div>
    </div>
  </div>
</section>
```

f) 3D cube styles and animation

```
/* 3D Cube Styles */
CodeMate
.cube-container {
  perspective: 1000px;
  width: 100px;
  height: 100px;
}

CodeMate
.cube {
  width: 100%;
  height: 100%;
  position: relative;
  transform-style: preserve-3d;
  animation: rotate3d 20s linear infinite;
  cursor: pointer;
}

CodeMate
.cube:hover {
  animation-play-state: paused;
}

CodeMate
.face {
  position: absolute;
  width: 100px;
  height: 100px;
  background: linear-gradient(45deg, #3B82F6, #1E40AF);
  border: 2px solid #000000;
  display: flex;
  align-items: center;
  justify-content: center;
  font-weight: bold;
  color: white;
  font-size: 12px;
  text-align: center;
  backdrop-filter: blur(10px);
  transition: all 0.3s ease;
}

CodeMate
.face:hover {
  background: linear-gradient(45deg, #F59E0B, #D97706);
  transform: scale(1.1);
}

CodeMate
.front { transform: rotateY(0deg) translateZ(50px); }
CodeMate
.back { transform: rotateY(180deg) translateZ(50px); }
CodeMate
.right { transform: rotateY(90deg) translateZ(50px); }
CodeMate
.left { transform: rotateY(-90deg) translateZ(50px); }
CodeMate
.top { transform: rotateX(90deg) translateZ(50px); }
CodeMate
.bottom { transform: rotateX(-90deg) translateZ(50px); }

CodeMate
@keyframes rotate3d {
  0% { transform: rotateX(0deg) rotateY(0deg); }
  100% { transform: rotateX(360deg) rotateY(360deg); }
}
```

g) Notification and particle helpers

```
const notify = (msg) => {
  const n = document.createElement('div');
  n.className = 'notification glass-card p-4 rounded-lg shadow-lg text-white';
  n.innerHTML = `<div class="flex items-center space-x-3"><span>${msg}</span></div>`;
  document.body.appendChild(n);
  setTimeout(() => { n.style.opacity = '0'; setTimeout(() => n.parentNode?.removeChild(n), 300); }, 3000);
};

CodeMate
const createParticle = (x, y, size = Math.random() * 10 + 5, color = 'radial-gradient(circle, #3882f6 0%, transparent 70%)') => {
  const p = document.createElement('div');
  p.className = 'particle';
  Object.assign(p.style, { left: x + 'px', top: y + 'px', width: size + 'px', height: size + 'px', background: color });
  document.body.appendChild(p);
  setTimeout(() => p.parentNode?.removeChild(p), 3000);
};
```

h) Counter animation logic

```
const animateCounter = (c) => {
  const target = parseFloat(c.getAttribute('data-target'));
  const step = target / 125; // 2000ms / 16ms = 125 steps
  let current = 0;
  const timer = setInterval(() => {
    current += step;
    if (current >= target) { current = target; clearInterval(timer); }
    c.textContent = target >= 100 ? Math.floor(current).toLocaleString() : Math.floor(current);
    c.classList.add('animate');
    setTimeout(() => c.classList.remove('animate'), 300);
  }, 16);
};
```

i) Wrapper + utilities + minimal CSS injection

```
// Extra 30% Features 🚀
// =====
Tabnine | Edit | Test | Fix | Explain | Document | CodeMate
(function () {
  // Utilities
  const create = (tag, props = {}, children = []) => {
    const el = document.createElement(tag);
    Object.entries(props).forEach(([k, v]) => {
      if (k === 'className') el.className = v;
      else if (k === 'innerHTML') el.innerHTML = v;
      else if (k.startsWith('on') && typeof v === 'function') el.addEventListener(k.substring(2).toLowerCase(), v);
      else el.setAttribute(k, v);
    });
    children.forEach(c => el.appendChild(c));
    return el;
  };

  const injectStyles = () => {
    const css = `:root{--primary:#3882f6;--primaryDark:#1e40af}.reduce-motion *{animation:none!important;transition:none!important}.settings-panel{position
    const style = create('style', { innerHTML: css });
    document.head.appendChild(style);
  };
  injectStyles();
});
```


j) Quick Settings panel (theme, reduce motion, print)

```
// Quick Settings Panel
const settingsToggle = create('button', { id: 'settingsToggle', className: 'fixed top-4 left-4 z-50 w-12 h-12 bg-gradient-to-r from-blue-500 to-blue-700 text-white rounded-full shadow-lg hover:shadow-xl' });
const settingsPanel = create('div', { id: 'settingsPanel', className: 'settings-panel hidden' });
settingsPanel.innerHTML = `<div class="glass-card p-4 rounded-xl shadow-lg">
  <h3 class="font-bold mb-3 text-white">Quick Settings</h3>
  <div class="space-y-3 text-sm">
    <div>
      <div class="text-white/90 mb-2">Theme</div>
      <div class="flex gap-2">
        <button class="theme-btn w-7 h-7 rounded-full border-2 border-white/40 data-colors="#3882F6,#1E40AF" style="background: linear-gradient(45deg,#3882F6,#1E40AF)"></button>
        <button class="theme-btn w-7 h-7 rounded-full border-2 border-white/40 data-colors="#10B981,#065F46" style="background: linear-gradient(45deg,#10B981,#065F46)"></button>
        <button class="theme-btn w-7 h-7 rounded-full border-2 border-white/40 data-colors="#F59E08,#D97706" style="background: linear-gradient(45deg,#F59E08,#D97706)"></button>
        <button class="theme-btn w-7 h-7 rounded-full border-2 border-white/40 data-colors="#8B5CF6,#4C1095" style="background: linear-gradient(45deg,#8B5CF6,#4C1095)"></button>
      </div>
    </div>
    <label class="flex items-center gap-2 text-white/90">
      <input id="reduceMotionToggle" type="checkbox"> Reduce motion
    </label>
    <button id="printBtn" class="w-full bg-white text-blue-700 px-3 py-2 rounded-lg font-semibold hover:bg-blue-50">🖨️ Print this page</button>
  </div>
</div>`;
document.body.appendChild(settingsToggle);
document.body.appendChild(settingsPanel);
settingsToggle.addEventListener('click', () => settingsPanel.classList.toggle('hidden'));
settingsPanel.addEventListener('click', (e) => e.stopPropagation());
document.addEventListener('click', (e) => {
  if (!settingsPanel.classList.contains('hidden') && !settingsPanel.contains(e.target) && e.target !== settingsToggle) settingsPanel.classList.add('hidden');
});
settingsPanel.querySelectorAll('.theme-btn').forEach(btn => btn.addEventListener('click', () => {
  const [primary, primaryDark] = btn.getAttribute('data-colors').split(',');
  document.documentElement.style.setProperty('--primary', primary);
  document.documentElement.style.setProperty('--primaryDark', primaryDark);
  notify('🎨 Theme updated');
}));
settingsPanel.querySelector('#reduceMotionToggle').addEventListener('change', (e) => {
  document.body.classList.toggle('reduce-motion', e.target.checked);
  notify(e.target.checked ? '🔊 Reduced motion on' : '🔊 Reduced motion off');
});
settingsPanel.querySelector('#printBtn').addEventListener('click', () => window.print());
```

k) Project filters (All, AI/ML, Web, Mobile)

```
// Project Filters
const projectsSection = document.querySelector('#projects .max-w-6xl');
const grid = projectsSection?.querySelector('.grid');
if (projectsSection && grid) {
  const filterBar = create('div', { id: 'projectFilters', className: 'flex flex-wrap gap-2 justify-center mb-10' });
  const filters = [
    { label: 'All', value: 'all', cls: 'bg-white/90 text-blue-700 px-4 py-2 rounded-full hover:bg-white font-semibold' },
    { label: 'AI/ML', value: 'ai', cls: 'bg-white/20 text-white px-4 py-2 rounded-full hover:bg-white/30' },
    { label: 'Web', value: 'web', cls: 'bg-white/20 text-white px-4 py-2 rounded-full hover:bg-white/30' },
    { label: 'Mobile', value: 'mobile', cls: 'bg-white/20 text-white px-4 py-2 rounded-full hover:bg-white/30' }
  ];
  filters.forEach(f => filterBar.appendChild(create('button', { className: f.cls + ' filter-btn', 'data-filter': f.value, innerHTML: f.label })));
  projectsSection.insertBefore(filterBar, grid);

  const cards = Array.from(grid.querySelectorAll('.project-card'));
  cards.forEach(card => {
    const title = card.querySelector('h3')?.textContent || '';
    if (title.includes('Fake Logo')) card.setAttribute('data-tags', 'ai ml web');
    else if (title.includes('QuizSync')) card.setAttribute('data-tags', 'web realtime');
    else if (title.includes('Maharashtra Police')) card.setAttribute('data-tags', 'mobile flutter');
  });

  const applyFilter = (value) => {
    cards.forEach(c => {
      const tags = (c.getAttribute('data-tags') || '').toLowerCase();
      c.style.display = value === 'all' || tags.includes(value) ? '' : 'none';
    });
  };
  applyFilter('all');
  filterBar.addEventListener('click', (e) => {
    const btn = e.target.closest('.filter-btn');
    if (!btn) return;
    filterBar.querySelectorAll('.filter-btn').forEach(b => b.classList.remove('ring-2', 'ring-white'));
    btn.classList.add('ring-2', 'ring-white');
    applyFilter(btn.getAttribute('data-filter'));
  });
}
```

l) Testimonials slider (auto-rotating with Prev/Next)

```
// Testimonials Slider
const contactSection = document.getElementById('contact');
if (contactSection) {
  const testimonials = create('section', { id: 'testimonials', className: 'py-20 px-4' });
  testimonials.innerHTML = `<div class="max-w-4xl mx-auto">
    <h2 class="text-4xl font-bold text-center mb-16 animate-fade-in-up">Testimonials</h2>
    <div class="glass-card p-8 rounded-xl card-hover animate-fade-in-up">
      <div id="testimonialSlider" class="relative overflow-hidden">
        <div id="testimonialSlides" class="slides">
          <div class="slide"><div class="p-4"><p class="text-gray-800 text-lg italic">Abhijeet is a quick learner and a reliable teammate. His work on our AI project was exceptional.</p><div class="mt-3 text-gray-600">– Mentor, AI Internship</div></div></div>
          <div class="slide"><div class="p-4"><p class="text-gray-800 text-lg italic">Delivered a polished web app ahead of schedule. Great attention to UX and detail.</p><div class="mt-3 text-gray-600">– Client, QuizSync</div></div></div>
          <div class="slide"><div class="p-4"><p class="text-gray-800 text-lg italic">Strong problem-solving skills and ownership. Highly recommend!</p><div class="mt-3 text-gray-600">– Supervisor, Mobile Project</div></div></div>
        </div>
        <div class="flex items-center justify-between mt-6">
          <button id="prevTestimonial" class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700">Prev</button>
          <button id="nextTestimonial" class="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700">Next </button>
        </div>
      </div>
    </div>
  </div>`;
  contactSection.parentNode.insertBefore(testimonials, contactSection);
  const slidesEl = testimonials.querySelector('#testimonialSlides');
  let idx = 0; const total = slidesEl.children.length;
  const render = () => { slidesEl.style.transform = `translateX(-${idx * 100}%)`; };
  testimonials.querySelector('#prevTestimonial').addEventListener('click', () => { idx = (idx - 1 + total) % total; render(); });
  testimonials.querySelector('#nextTestimonial').addEventListener('click', () => { idx = (idx + 1) % total; render(); });
  setInterval(() => { idx = (idx + 1) % total; render(); }, 5000);
}
```

m) Quick Search overlay (Ctrl+K)

```
// Quick Search Overlay (Ctrl+K)
const overlay = create('div', { id: 'quickSearch', className: 'fixed inset-0 bg-black/50 backdrop-blur-sm hidden items-center justify-center z-50' });
overlay.innerHTML = `<div class="bg-white rounded-lg p-6 w-11/12 md:w-1/2 shadow-xl">
  <input id="quickSearchInput" type="text" placeholder="Type a section (home, about, projects, experience, testimonials, contact) and hit Enter" class="w-full p-3 border border-gray-300 rounded-lg focus:border-blue-500 focus:outline-none">
  <div class="text-sm text-gray-500 mt-2">Press Esc to close</div>
</div>`;
document.body.appendChild(overlay);
const showOverlay = () => { overlay.classList.remove('hidden'); setTimeout(() => overlay.classList.add('flex'), 0); overlay.querySelector('#quickSearchInput').focus(); };
const hideOverlay = () => { overlay.classList.add('hidden'); overlay.classList.remove('flex'); };
document.addEventListener('keydown', (e) => {
  if ((e.ctrlKey || e.metaKey) && e.key.toLowerCase() === 'k') { e.preventDefault(); showOverlay(); }
  if (e.key === 'Escape') hideOverlay();
});
overlay.addEventListener('click', (e) => { if (e.target === overlay) hideOverlay(); });
overlay.querySelector('#quickSearchInput').addEventListener('keydown', (e) => {
  if (e.key === 'Enter') {
    const val = e.target.value.trim().toLowerCase();
    const map = { home: '#home', about: '#about', projects: '#projects', experience: '#experience', testimonials: '#testimonials', contact: '#contact' };
    const id = map[val];
    if (id && document.querySelector(id)) {
      document.querySelector(id).scrollIntoView({ behavior: 'smooth' });
      hideOverlay();
    } else notify('Try: home, about, projects, experience, testimonials, contact');
  }
});
})();
```

4. OUTPUT :-

1.

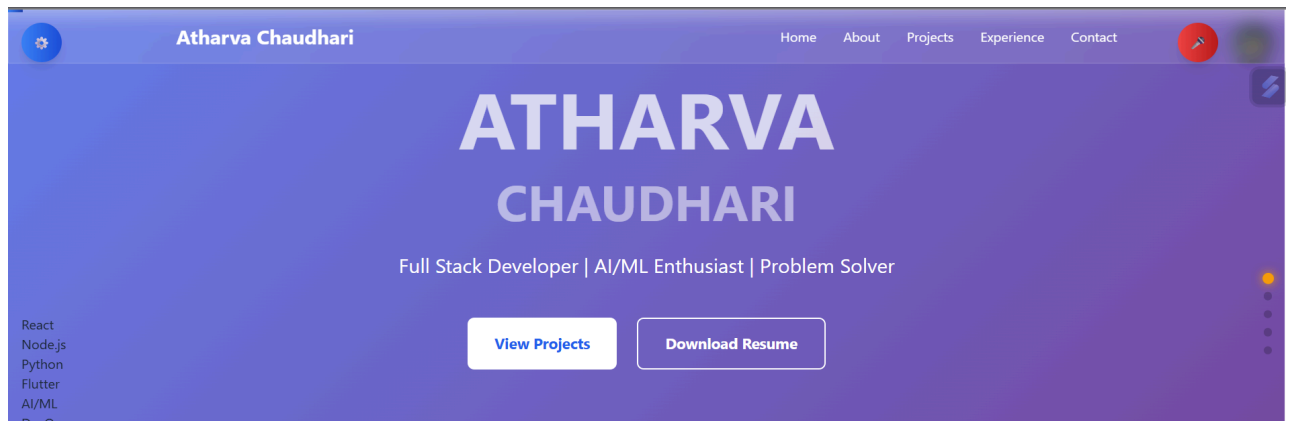


Fig1.

2.

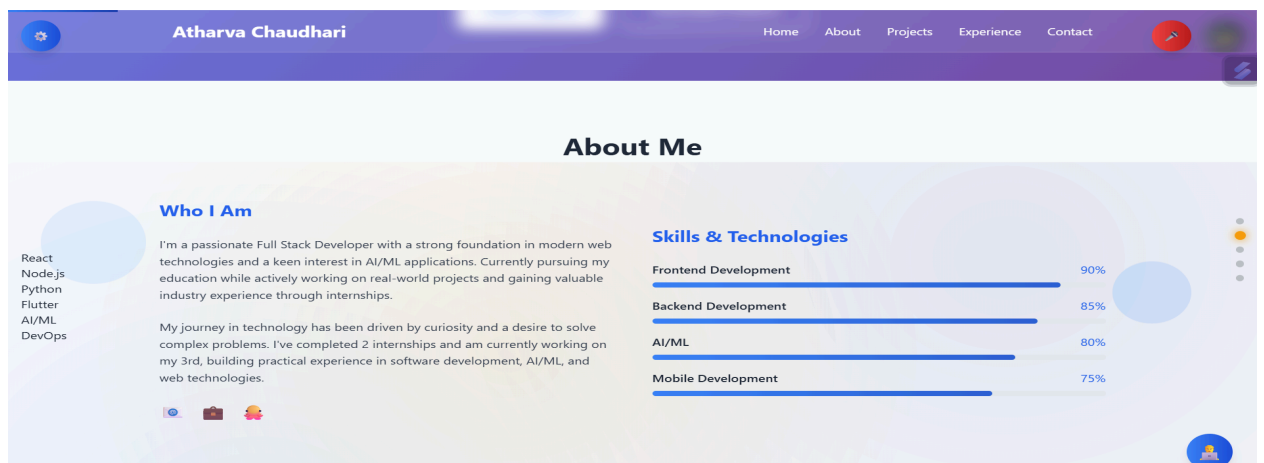
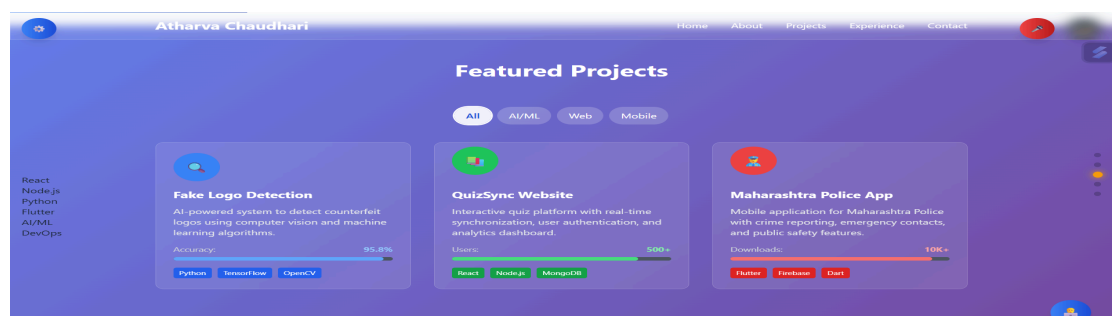


Fig2.

3.

Fig3.



4.

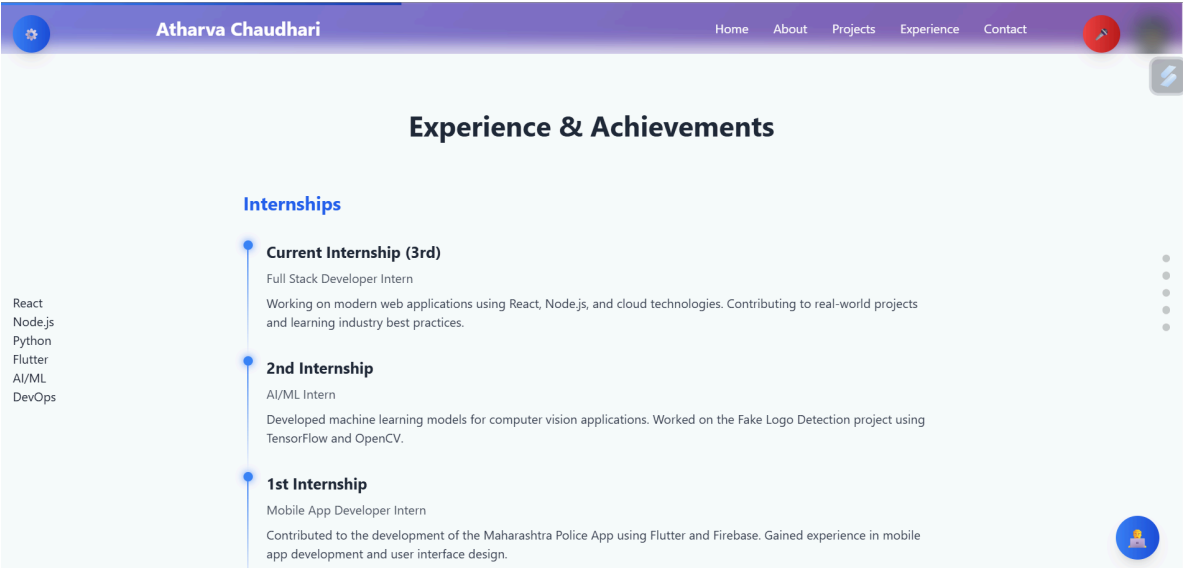


Fig4.

5.

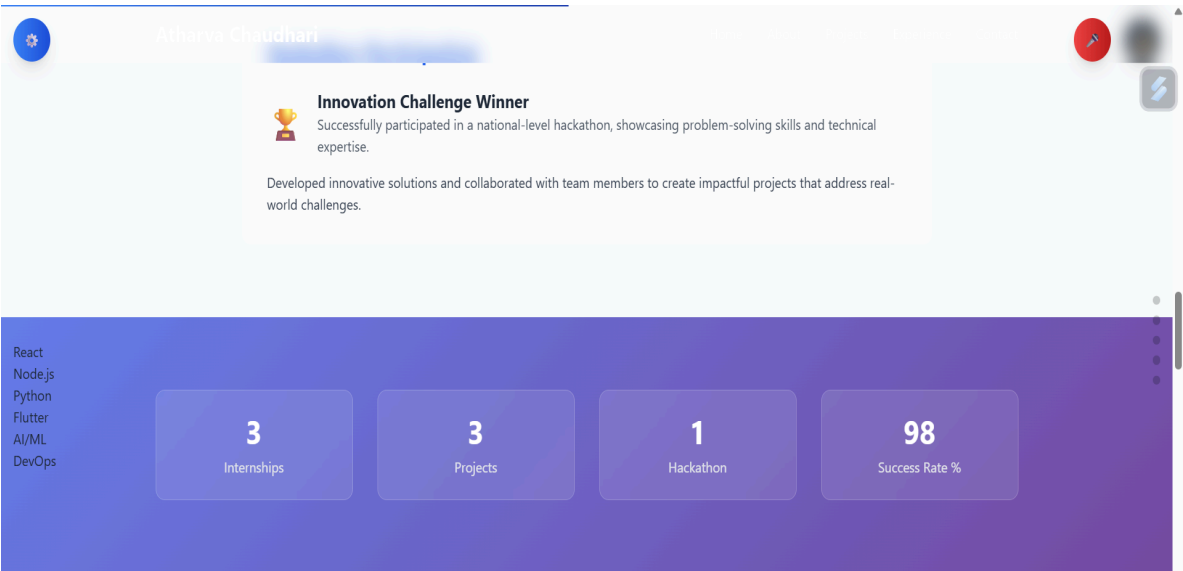


Fig5.

6.

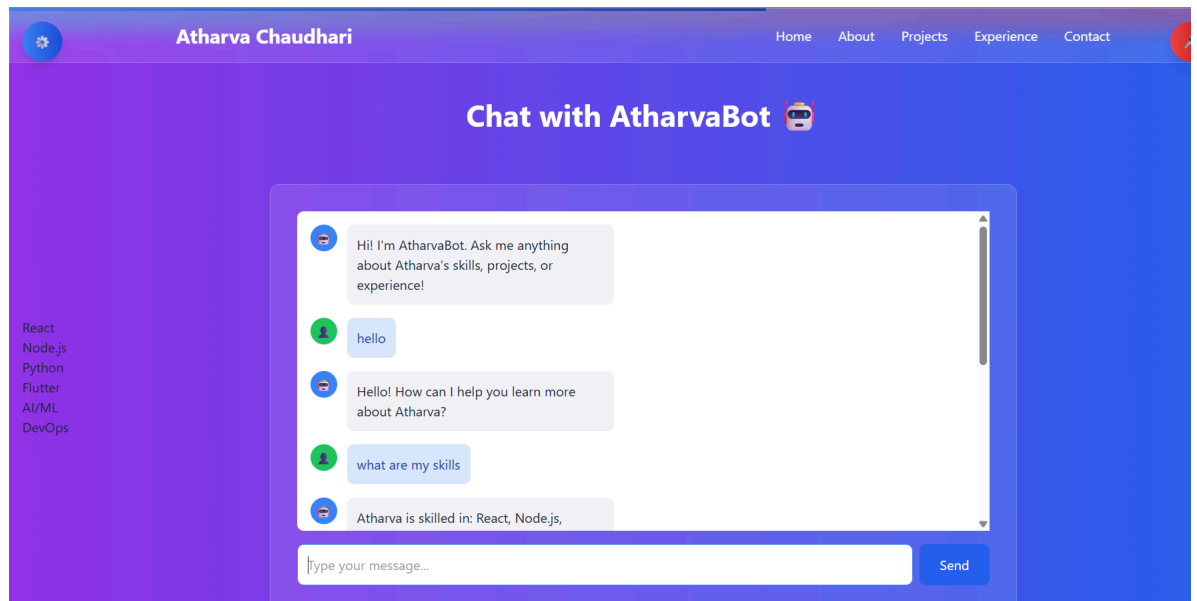


Fig6.

7.

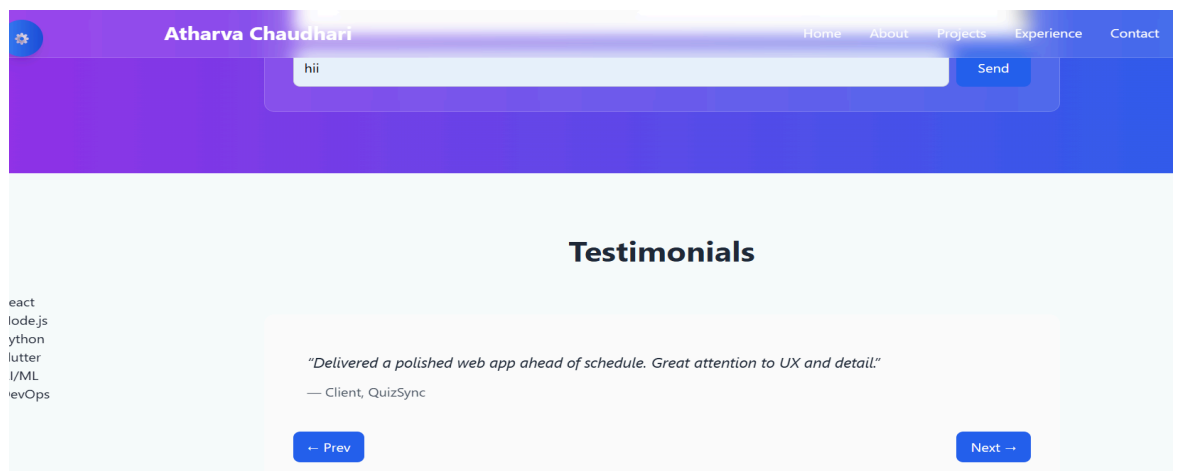


Fig7.

8.

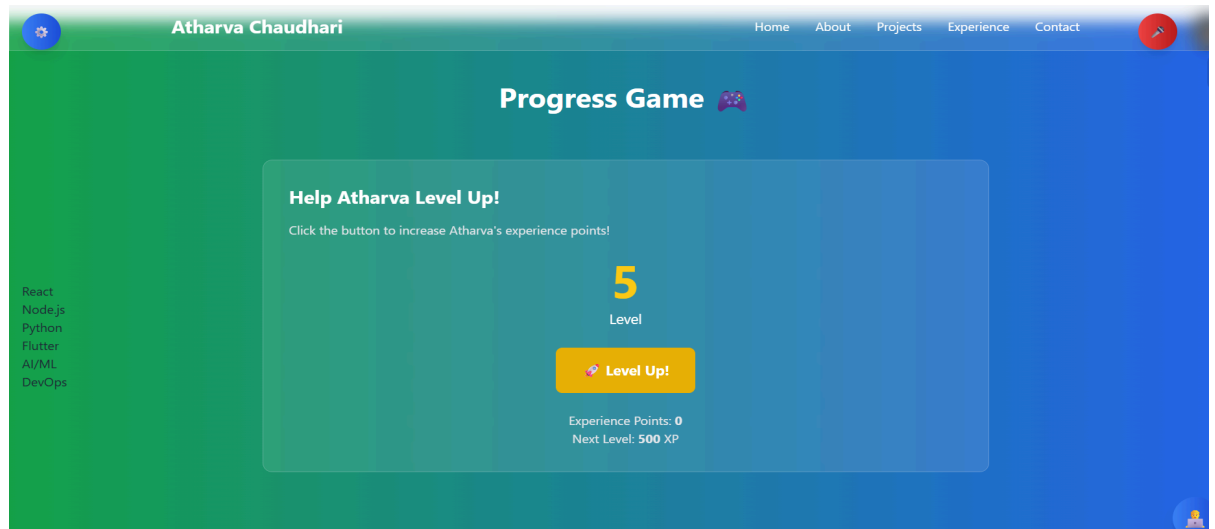


Fig8.

9.

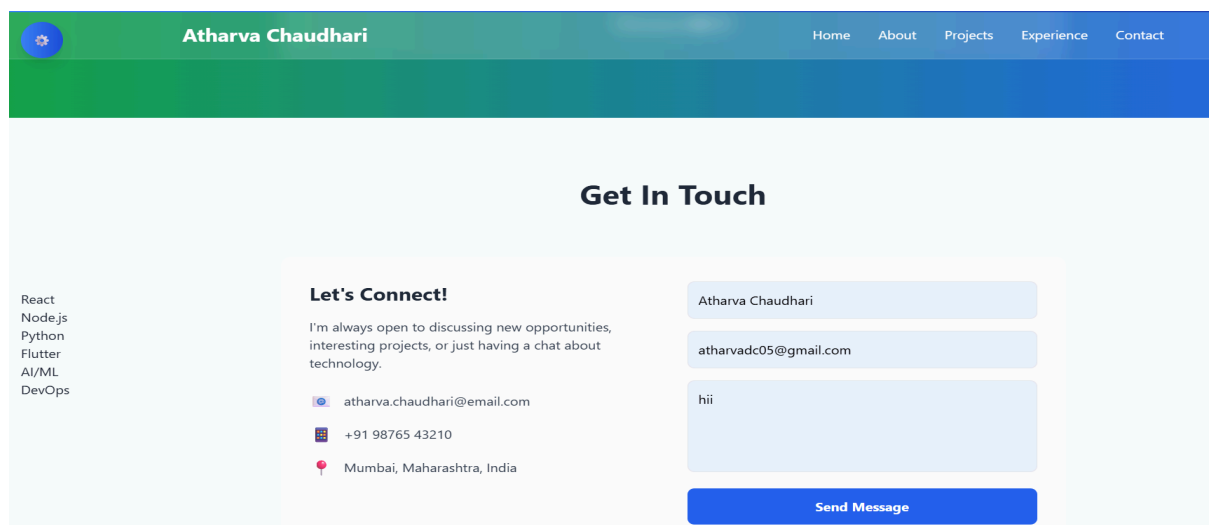


Fig9.

5. Conclusion:

This project successfully demonstrated how Tailwind CSS can be used to build a modern, responsive, and interactive portfolio website. By combining utility-first styling with JavaScript enhancements, the project not only fulfilled the experiment's aim but also went further by adding advanced features like voice commands, dynamic counters, animations, and a chatbot. The result is a professional, user-friendly portfolio that highlights both the power of Tailwind CSS and the ability to create highly interactive web experiences.