```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MedFinder - Smart Search</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link href="https://fonts.googleapis.com/css2?
family=Inter:wght@300;400;500;600;700;800&display=swap" rel="stylesheet">
    <script>
        tailwind.config = {
            theme: {
                extend: {
                    colors: {
                        primary: {
                            50: '#f0f9ff',
                            100: '#e0f2fe',
                            200: '#bae6fd',
                            300: '#7dd3fc',
                            400: '#38bdf8',
                            500: '#0ea5e9',
                            600: '#0284c7',
                            700: '#0369a1',
                            800: '#075985',
                            900: '#0c4a6e'
                        },
                        secondary: {
                            50: '#f0fdf4',
                            100: '#dcfce7',
                            200: '#bbf7d0',
                            300: '#86efac',
                            400: '#4ade80',
                            500: '#22c55e',
                            600: '#16a34a',
                            700: '#15803d',
                            800: '#166534',
                            900: '#14532d'
                        },
                        accent: {
                            50: '#fef3c7',
                            100: '#fde68a',
                            200: '#fcd34d',
                            300: '#fbbf24',
                            400: '#f59e0b',
                            500: '#d97706',
                            600: '#b45309',
                            700: '#92400e',
                            800: '#78350f',
                            900: '#451a03'
                        }
                    },
                    fontFamily: {
                        'inter': ['Inter', 'sans-serif']
                    },
                    backdropBlur: {
                        xs: '2px'
                    }
                }
            }
        }
```

```
</script>
<style>
    body {
        background: linear-gradient(135deg, #0f0f23 0%, #1a1a2e 50%, #16213e 100%);
        font-family: 'Inter', sans-serif;
        min-height: 100vh;
    }

    .glass-card {
        background: rgba(30, 30, 60, 0.95);
        border: 1px solid rgba(139, 92, 246, 0.3);
        box-shadow: 0 4px 16px rgba(139, 92, 246, 0.2);
        backdrop-filter: blur(10px);
    }

    .glass-surface {
        background: rgba(15, 15, 35, 0.95);
        border: 1px solid rgba(139, 92, 246, 0.2);
        backdrop-filter: blur(15px);
    }

    .card-3d {
        background: linear-gradient(145deg, #2d2d5f, #1e1e3f);
        box-shadow:
            20px 20px 60px rgba(0, 0, 0, 0.5),
            -20px -20px 60px rgba(139, 92, 246, 0.1),
            inset 0 1px 0 rgba(139, 92, 246, 0.3);
        transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
    }

    .card-3d:hover {
        transform: translateY(-8px) rotateX(5deg);
        box-shadow:
            25px 25px 80px rgba(0, 0, 0, 0.6),
            -25px -25px 80px rgba(139, 92, 246, 0.15),
            0 20px 40px rgba(139, 92, 246, 0.2);
    }

    .button-3d {
        background: linear-gradient(145deg, #8b5cf6, #7c3aed);
        box-shadow: 0 4px 12px rgba(139, 92, 246, 0.4);
        transition: all 0.2s cubic-bezier(0.4, 0, 0.2, 1);
    }

    .button-3d:hover {
        transform: translateY(-2px);
        box-shadow: 0 6px 20px rgba(139, 92, 246, 0.5);
        background: linear-gradient(145deg, #9333ea, #8b5cf6);
    }

    .button-3d:active {
        transform: translateY(0);
        box-shadow: 0 2px 8px rgba(139, 92, 246, 0.4);
    }

    .search-glow {
        background: rgba(45, 45, 95, 0.9);
        border: 2px solid rgba(139, 92, 246, 0.4);
        box-shadow: 0 4px 16px rgba(139, 92, 246, 0.2);
        transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
    backdrop-filter: blur(10px);
}

.search-glow:focus-within {
    border-color: #8b5cf6;
    box-shadow: 0 4px 16px rgba(139, 92, 246, 0.3), 0 0 0 3px rgba(139, 92, 246, 0.2);
    background: rgba(45, 45, 95, 0.95);
}

.pill-tag {
    background: linear-gradient(135deg, #8b5cf6, #7c3aed);
    box-shadow: 0 4px 15px rgba(139, 92, 246, 0.4);
    animation: slideInUp 0.3s cubic-bezier(0.4, 0, 0.2, 1);
}

.suggestion-item {
    background: rgba(45, 45, 95, 0.95);
    border: 1px solid rgba(139, 92, 246, 0.3);
    transition: all 0.2s cubic-bezier(0.4, 0, 0.2, 1);
    animation: fadeInDown 0.15s cubic-bezier(0.4, 0, 0.2, 1);
}

.suggestion-item:hover {
    background: rgba(139, 92, 246, 0.2);
    transform: translateX(4px);
    border-color: rgba(139, 92, 246, 0.5);
}

.suggestion-item:active {
    background: rgba(139, 92, 246, 0.3);
    transform: translateX(2px);
    transition: all 0.1s ease;
}

.suggestion-item:focus {
    outline: none;
}

.result-card {
    background: white;
    box-shadow:
        0 4px 6px -1px rgba(0, 0, 0, 0.1),
        0 2px 4px -1px rgba(0, 0, 0, 0.06);
    transition: all 0.4s cubic-bezier(0.4, 0, 0.2, 1);
    animation: slideInUp 0.4s cubic-bezier(0.4, 0, 0.2, 1);
    border: 1px solid #e5e7eb;
}

.result-card:hover {
    transform: translateY(-4px);
    box-shadow:
        0 20px 25px -5px rgba(0, 0, 0, 0.1),
        0 10px 10px -5px rgba(0, 0, 0, 0.04);
    border-color: #d1d5db;
}

.toggle-switch {
    background: #4b5563;
    border: 2px solid rgba(139, 92, 246, 0.3);
    transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
```

```
}

.toggle-switch.active {
   background: #8b5cf6;
   border: 2px solid rgba(139, 92, 246, 0.5);
   box-shadow: 0 0 20px rgba(139, 92, 246, 0.3);
}

.toggle-knob {
   background: white;
   box-shadow: 0 2px 8px rgba(0, 0, 0, 0.2);
   border: 1px solid rgba(0, 0, 0, 0.1);
   transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
}

.profile-modal {
   background: white;
   border: 1px solid rgba(0, 0, 0, 0.1);
   box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);
}

.chat-bubble-sent {
   background: linear-gradient(135deg, #8b5cf6, #7c3aed);
   box-shadow: 0 4px 15px rgba(139, 92, 246, 0.4);
}

.chat-bubble-received {
   background: rgba(45, 45, 95, 0.9);
   border: 1px solid rgba(139, 92, 246, 0.2);
   box-shadow: 0 4px 15px rgba(0, 0, 0, 0.3);
}

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

@keyframes fadeInDown {
   from {
      opacity: 0;
      transform: translateY(-5px);
   }
   to {
      opacity: 1;
      transform: translateY(0);
   }
}

@keyframes fadeIn {
   from { opacity: 0; }
   to { opacity: 1; }
}

.screen-transition {
    transition: all 0.4s cubic-bezier(0.25, 0.46, 0.45, 0.94);
}

.slide-out-left {
    transform: translateX(-100%);
    opacity: 0;
}

.slide-in-right {
    transform: translateX(100%);
    opacity: 0;
}

.slide-in-active {
    transform: translateX(0);
    opacity: 1;
}

.drug-tab {
    background: #f3f4f6;
    border: 1px solid #d1d5db;
    transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
    animation: slideInUp 0.3s cubic-bezier(0.4, 0, 0.2, 1);
}

.drug-tab:hover {
    transform: translateY(-2px);
    background: #e5e7eb;
    border-color: #9ca3af;
}

.drug-tab.active {
    background: linear-gradient(135deg, #3b82f6, #2563eb);
    box-shadow: 0 4px 15px rgba(59, 130, 246, 0.3);
    border-color: #3b82f6;
}

.pharmacy-card {
    background: linear-gradient(145deg, #2d2d5f, #1e1e3f);
    box-shadow:
        0 20px 25px -5px rgba(0, 0, 0, 0.4),
        0 10px 10px -5px rgba(0, 0, 0, 0.2);
    transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
    border: 1px solid rgba(139, 92, 246, 0.2);
}

.pharmacy-card:hover {
    transform: translateY(-4px);
    box-shadow:
        0 25px 50px -12px rgba(139, 92, 246, 0.3),
        0 20px 20px -5px rgba(0, 0, 0, 0.3);
    border-color: rgba(139, 92, 246, 0.4);
}

.loading-spinner {
    animation: spin 1s linear infinite;
}

@keyframes spin {
    from { transform: rotate(0deg); }
```

```
            to { transform: rotate(360deg); }
        }

        .logo-container {
            position: relative;
            display: inline-block;
            overflow: hidden;
        }

        .medicine-rain {
            position: absolute;
            top: -30px;
            left: 0;
            right: 0;
            height: 30px;
            pointer-events: none;
            z-index: 10;
            overflow: visible;
        }

        .medicine-item {
            position: absolute;
            font-size: 12px;
            opacity: 0.8;
            animation: fall linear infinite;
            z-index: 10;
        }

        @keyframes fall {
            0% {
                transform: translateY(-30px);
                opacity: 0.8;
            }
            100% {
                transform: translateY(100px);
                opacity: 0;
            }
        }
    </style>
</head>
<body class="min-h-screen font-inter overflow-x-hidden">
    <!-- Status Bar -->
    <div class="flex justify-between items-center px-6 py-3 text-sm font-medium glass-surface">
        <span class="text-white font-semibold">9:41</span>
        <div class="flex items-center space-x-1">
            <div class="flex space-x-1">
                <div class="w-1 h-3 bg-white rounded-full"></div>
                <div class="w-1 h-3 bg-white rounded-full"></div>
                <div class="w-1 h-3 bg-white opacity-60 rounded-full"></div>
                <div class="w-1 h-3 bg-white opacity-30 rounded-full"></div>
            </div>
            <svg class="w-6 h-6 text-white ml-2" fill="currentColor" viewBox="0 0 24 24">
                <path d="M15.67 4H14V2c0-1.1-.9-2-2-2s-2 .9-2 2v2H8.33C7.6 4 7 4.6 7
5.33v15.33C7 21.4 7.6 22 8.33 22h7.33c.74 0 1.34-.6 1.34-1.33V5.33C17 4.6 16.4 4 15.67 4z"/>
            </svg>
        </div>
    </div>

    <!-- Main Container -->
    <div class="search-container">
```

```
        <!-- Search Screen -->
        <div id="searchScreen" class="screen screen-transition">
            <div class="flex flex-col items-center px-6 py-4">
                <!-- Header with Profile -->
                <div class="w-full max-w-md flex justify-between items-center mb-6">
                    <div></div>
                    <div class="relative">
                        <button onclick="toggleProfileDropdown()" class="w-12 h-12 bg-white/20
hover:bg-white/30 backdrop-blur-sm border border-white/30 rounded-full flex items-center
justify-center transition-all duration-200" id="profileButton">
                            <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0
24 24">
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M16 7a4 4 0 11-8 0 4 4 0 018 0zM12 14a7 7 0 00-7 7h14a7 7 0 00-7-7z"/>
                            </svg>
                        </button>

                        <!-- Profile Dropdown -->
                        <div id="profileDropdown" class="absolute right-0 top-14 w-48 bg-white/95
backdrop-blur-sm border border-white/30 rounded-xl shadow-lg hidden z-30">
                            <div class="py-2">
                                <button onclick="showProfile()" class="w-full px-4 py-3 text-left text-gray-800
hover:bg-gray-100 transition-colors flex items-center space-x-3 border-b border-gray-200">
                                    <svg class="w-5 h-5" fill="none" stroke="currentColor" viewBox="0 0 24
24">
                                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M16 7a4 4 0 11-8 0 4 4 0 018 0zM12 14a7 7 0 00-7 7h14a7 7 0 00-7-7z"/>
                                    </svg>
                                    <span class="font-medium">Profile</span>
                                </button>
                                <button onclick="showChat()" class="w-full px-4 py-3 text-left text-gray-800
hover:bg-gray-100 transition-colors flex items-center justify-between">
                                    <div class="flex items-center space-x-3">
                                        <svg class="w-5 h-5" fill="none" stroke="currentColor" viewBox="0 0 24
24">
                                            <path stroke-linecap="round" stroke-linejoin="round" stroke-
width="2" d="M8 12h.01M12 12h.01M16 12h.01M21 12c0 4.418-4.03 8-9 8a9.863 9.863 0
01-4.255-.949L3 20l1.395-3.72C3.512 15.042 3 13.574 3 12c0-4.418 4.03-8 9-8s9 3.582 9 8z"/>
                                        </svg>
                                        <span class="font-medium">Chat</span>
                                    </div>
                                    <span class="w-5 h-5 bg-red-500 rounded-full flex items-center justify-
center text-xs text-white">3</span>
                                </button>
                            </div>
                        </div>
                    </div>
                </div>

                <!-- App Logo and Title -->
                <div class="mb-4 text-center">
                    <div class="logo-container">
                        <div class="medicine-rain" id="medicineRain"></div>
                        <h1 class="text-5xl font-bold text-white tracking-tight mb-1 relative z-10"
style="text-shadow: 0 4px 20px rgba(0,0,0,0.3);">MedFinder</h1>
                    </div>
                    <!-- Tagline -->
                    <p class="text-white/80 text-sm font-medium">Find medicines instantly</p>
                </div>
```

```
<!-- Search Container -->
<div class="w-full max-w-lg relative">
    <!-- Search Bar -->
    <div class="relative search-glow rounded-2xl">
        <input
            type="text"
            placeholder="Type drug name…"
            class="w-full py-3 pl-6 pr-20 bg-transparent text-white placeholder-gray-300 rounded-2xl focus:outline-none text-base font-normal"
            id="searchInput"
            autocomplete="off"
        >
        <div class="absolute right-3 top-1/2 transform -translate-y-1/2 flex space-x-2">
            <!-- Camera Icon -->
            <button onclick="openCamera()" class="p-2 hover:bg-gray-100 rounded-full transition-colors">
                <svg class="w-5 h-5 text-orange-400" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M3 9a2 2 0 012-2h.93a2 2 0 001.664-.89l.812-1.22A2 2 0 0110.07 4h3.86a2 2 0 011.664.89l.812 1.22A2 2 0 0018.07 7H19a2 2 0 012 2v9a2 2 0 01-2 2H5a2 2 0 01-2-2V9z"/>
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15 13a3 3 0 11-6 0 3 3 0 016 0z"/>
                </svg>
            </button>
            <!-- Upload Icon -->
            <button onclick="uploadFile()" class="p-2 hover:bg-gray-100 rounded-full transition-colors">
                <svg class="w-5 h-5 text-cyan-400" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M7 16a4 4 0 01-.88-7.903A5 5 0 1115.9 6L16 6a5 5 0 011 9.9M15 13l-3-3m0 0l-3 3m3-3v12"/>
                </svg>
            </button>
        </div>
    </div>

    <!-- Autocomplete Dropdown -->
    <div id="autocompleteDropdown" class="absolute top-full left-0 right-0 mt-2 glass-card rounded-xl hidden z-20 overflow-hidden">
        <div id="suggestionsList" class="max-h-64 overflow-y-auto">
            <!-- Suggestions will be populated here -->
        </div>
    </div>
</div>

<!-- Multiple Search Toggle - positioned below search bar -->
<div class="flex justify-center mt-6">
    <div class="flex items-center space-x-3">
        <span class="text-white/90 font-medium text-sm">Multiple drug search</span>
        <button onclick="toggleMultipleSearch()" class="toggle-switch w-12 h-6 rounded-full p-1 flex items-center" id="multipleToggle">
            <div class="toggle-knob w-4 h-4 rounded-full transition-transform duration-300" id="toggleKnob"></div>
        </button>
    </div>
</div>

<!-- Selected Drugs Container -->
```

```
                <div id="selectedDrugs" class="w-full max-w-lg mt-6 space-y-2 hidden">
                    <div class="text-white/80 text-sm font-medium mb-2">Selected medicines:</div>
                    <div id="drugsList" class="space-y-2">
                        <!-- Selected drugs will be populated here -->
                    </div>
                    <button onclick="performSearch()" id="searchButton" class="w-full bg-blue-800
hover:bg-blue-900 text-white py-3 rounded-2xl font-semibold text-lg mt-4 flex items-center
justify-center space-x-2 shadow-lg transition-all duration-200">
                        <span id="searchButtonText">Search Medicines</span>
                        <div id="searchSpinner" class="loading-spinner w-5 h-5 border-2 border-white
border-t-transparent rounded-full hidden"></div>
                    </button>
                </div>
            </div>
        </div>

        <!-- Results Screen -->
        <div id="resultsScreen" class="screen screen-transition slide-in-right hidden bg-gray-50
min-h-screen">
            <!-- Header -->
            <div class="flex items-center justify-between px-6 py-4 glass-surface">
                <div class="flex items-center space-x-4">
                    <button onclick="goBackToSearch()" class="p-2 hover:bg-white/20 rounded-full
transition-colors">
                        <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0
24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15
19l-7-7 7-7"/>
                        </svg>
                    </button>
                    <div>
                        <h1 class="text-xl font-bold text-white">Search Results</h1>
                        <p class="text-sm text-white/80 mt-1">Tap on a medicine below to see results</
p>
                    </div>
                </div>
                <button onclick="toggleMapView()" class="p-2 hover:bg-white/20 rounded-full
transition-colors">
                    <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0 24
24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9
20l-5.447-2.724A1 1 0 013 16.382V5.618a1 1 0 011.447-.894L9 7m0 13l6-3m-6 3V7m6 10l4.553
2.276A1 1 0 0021 18.382V7.618a1 1 0 00-1.447-.894L15 4m0 13V4m0 0L9 7"/>
                    </svg>
                </button>
            </div>

            <!-- Drug Tabs -->
            <div class="px-6 py-4 bg-blue-50 border-b border-blue-100">
                <div id="drugTabs" class="flex space-x-2 overflow-x-auto pb-2">
                    <!-- Drug tabs will be populated here -->
                </div>
            </div>

            <!-- Results Container -->
            <div class="px-6 py-4">
                <div id="resultsContainer" class="space-y-4">
                    <!-- Results will be populated here -->
                </div>
                <!-- Load More Button -->
                <div class="flex justify-center mt-8">
                    <button onclick="loadMore()" id="loadMoreBtn" class="bg-blue-600 hover:bg-blue-700 text-white px-6 py-3 rounded-full font-medium transition-colors">
                        Load More Results
                    </button>
                </div>

                <!-- Pagination Dots -->
                <div class="flex justify-center mt-6 space-x-2">
                    <div class="w-2 h-2 bg-blue-600 rounded-full"></div>
                    <div class="w-2 h-2 bg-gray-300 rounded-full"></div>
                    <div class="w-2 h-2 bg-gray-300 rounded-full"></div>
                    <div class="w-2 h-2 bg-gray-300 rounded-full"></div>
                </div>
            </div>
        </div>
    </div>

    <!-- Profile Modal -->
    <div id="profileModal" class="fixed inset-0 bg-black/50 backdrop-blur-sm hidden z-50 flex items-center justify-center p-4">
        <div class="profile-modal rounded-3xl p-6 w-full max-w-md max-h-[80vh] overflow-y-auto">
            <!-- Profile Header -->
            <div class="flex items-center justify-between mb-6">
                <h3 class="text-2xl font-bold text-gray-800">Profile</h3>
                <button onclick="closeProfile()" class="p-2 hover:bg-gray-100 rounded-full transition-colors">
                    <svg class="w-6 h-6 text-gray-600" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12"/>
                    </svg>
                </button>
            </div>



            <!-- Profile Tab Content -->
            <div id="profileTabContent">
                <!-- User Status -->
                <div class="mb-6 p-4 bg-gray-50 rounded-xl border border-gray-200">
                    <div class="flex items-center space-x-3 mb-3">
                        <div class="w-12 h-12 bg-blue-100 rounded-full flex items-center justify-center">
                            <svg class="w-6 h-6 text-blue-600" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M16 7a4 4 0 11-8 0 4 4 0 018 0zM12 14a7 7 0 00-7 7h14a7 7 0 00-7-7z"/>
                            </svg>
                        </div>
                        <div>
                            <h4 class="font-semibold text-gray-800" id="userStatus">Anonymous User</h4>
                            <p class="text-sm text-gray-600" id="chatLimit">4 chats remaining this week</p>
                        </div>
                    </div>
                </div>

                <!-- Profile Form -->
                <form onsubmit="saveProfile(event)" class="space-y-4">
                    <div class="grid grid-cols-2 gap-4">
                        <div>
                            <label class="block text-sm font-medium text-gray-700 mb-2">First Name</label>
                            <input type="text" id="firstName" class="w-full py-3 px-4 bg-white border border-gray-300 rounded-xl text-gray-800 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-primary-500" placeholder="John">
                        </div>
                        <div>
                            <label class="block text-sm font-medium text-gray-700 mb-2">Last Name</label>
                            <input type="text" id="lastName" class="w-full py-3 px-4 bg-white border border-gray-300 rounded-xl text-gray-800 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-primary-500" placeholder="Doe">
                        </div>
                    </div>

                    <div>
                        <label class="block text-sm font-medium text-gray-700 mb-2">Email</label>
                        <input type="email" id="email" class="w-full py-3 px-4 bg-white border border-gray-300 rounded-xl text-gray-800 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-primary-500" placeholder="john@example.com">
                    </div>

                    <div>
                        <label class="block text-sm font-medium text-gray-700 mb-2">Contact</label>
                        <input type="tel" id="contact" class="w-full py-3 px-4 bg-white border border-gray-300 rounded-xl text-gray-800 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-primary-500" placeholder="+233 123 456 789">
                    </div>

                    <button type="submit" class="w-full bg-blue-600 hover:bg-blue-700 text-white py-3 rounded-xl font-semibold mt-6 transition-colors">
                        Save Profile
                    </button>
                </form>
            </div>


        </div>
    </div>

    <!-- Chat List Screen -->
    <div id="chatListScreen" class="fixed inset-0 bg-black/50 backdrop-blur-sm hidden z-50">
        <div class="h-full flex flex-col bg-white">
            <!-- Chat List Header -->
            <div class="flex items-center justify-between px-4 py-3 glass-surface">
                <div class="flex items-center space-x-3">
                    <button onclick="closeChatListScreen()" class="p-2 hover:bg-white/20 rounded-full transition-colors">
                        <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15 19l-7-7 7-7"/>
                        </svg>
                    </button>
                    <div>
                        <h1 class="text-xl font-bold text-white">Chats</h1>
                        <p class="text-sm text-white/80">Your conversations</p>
```

```
          </div>
        </div>
        <button class="p-2 hover:bg-white/20 rounded-full transition-colors">
          <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0 24
24">
            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12
6v6m0 0v6m0-6h6m-6 0H6"/>
          </svg>
        </button>
      </div>

      <!-- Chat List -->
      <div class="flex-1 overflow-y-auto" style="background: linear-gradient(135deg, #f8fafc
0%, #e2e8f0 100%);">
        <div class="p-4 space-y-3">
          <!-- Chat Item 1 -->
          <div onclick="openChat('CareFirst Pharmacy', 'CF')" class="result-card rounded-2xl
p-4 cursor-pointer hover:shadow-lg transition-all">
            <div class="flex items-center space-x-4">
              <div class="w-12 h-12 card-3d rounded-full flex items-center justify-center flex-
shrink-0">
                <span class="text-primary-600 font-medium text-sm">CF</span>
              </div>
              <div class="flex-1 min-w-0">
                <div class="flex items-center justify-between mb-1">
                  <h3 class="font-semibold text-gray-800 truncate">CareFirst Pharmacy</
h3>
                  <span class="text-xs text-gray-500">2:33 PM</span>
                </div>
                <div class="flex items-center justify-between">
                  <p class="text-sm text-gray-600 truncate">I need a pack of 20 tablets.
What's the price?</p>
                  <div class="flex items-center space-x-2">
                    <span class="w-2 h-2 bg-secondary-500 rounded-full"></span>
                    <span class="text-xs bg-red-500 text-white rounded-full px-2 py-1">2</
span>
                  </div>
                </div>
              </div>
            </div>
          </div>

          <!-- Chat Item 2 -->
          <div onclick="openChat('HealthPlus Pharmacy', 'HP')" class="result-card
rounded-2xl p-4 cursor-pointer hover:shadow-lg transition-all">
            <div class="flex items-center space-x-4">
              <div class="w-12 h-12 card-3d rounded-full flex items-center justify-center flex-
shrink-0">
                <span class="text-primary-600 font-medium text-sm">HP</span>
              </div>
              <div class="flex-1 min-w-0">
                <div class="flex items-center justify-between mb-1">
                  <h3 class="font-semibold text-gray-800 truncate">HealthPlus Pharmacy</
h3>
                  <span class="text-xs text-gray-500">Yesterday</span>
                </div>
                <div class="flex items-center justify-between">
                  <p class="text-sm text-gray-600 truncate">Your order is ready for pickup</
p>
```

```
                                    <span class="text-xs bg-red-500 text-white rounded-full px-2 py-1">1</
span>
                                </div>
                            </div>
                        </div>
                    </div>

                    <!-- Chat Item 3 -->
                    <div onclick="openChat('MediCare Center', 'MC')" class="result-card rounded-2xl
p-4 cursor-pointer hover:shadow-lg transition-all">
                        <div class="flex items-center space-x-4">
                            <div class="w-12 h-12 card-3d rounded-full flex items-center justify-center flex-
shrink-0">
                                <span class="text-primary-600 font-medium text-sm">MC</span>
                            </div>
                            <div class="flex-1 min-w-0">
                                <div class="flex items-center justify-between mb-1">
                                    <h3 class="font-semibold text-gray-800 truncate">MediCare Center</h3>
                                    <span class="text-xs text-gray-500">Monday</span>
                                </div>
                                <div class="flex items-center justify-between">
                                    <p class="text-sm text-gray-500 truncate">Thank you for your purchase!</
p>
                                    <svg class="w-4 h-4 text-primary-500" fill="currentColor" viewBox="0 0 20
20">
                                        <path fill-rule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0
01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clip-
rule="evenodd"/>
                                    </svg>
                                </div>
                            </div>
                        </div>
                    </div>

                    <!-- Chat Item 4 -->
                    <div onclick="openChat('Wellness Pharmacy', 'WP')" class="result-card rounded-2xl
p-4 cursor-pointer hover:shadow-lg transition-all">
                        <div class="flex items-center space-x-4">
                            <div class="w-12 h-12 card-3d rounded-full flex items-center justify-center flex-
shrink-0">
                                <span class="text-primary-600 font-medium text-sm">WP</span>
                            </div>
                            <div class="flex-1 min-w-0">
                                <div class="flex items-center justify-between mb-1">
                                    <h3 class="font-semibold text-gray-800 truncate">Wellness Pharmacy</
h3>
                                    <span class="text-xs text-gray-500">Sunday</span>
                                </div>
                                <div class="flex items-center justify-between">
                                    <p class="text-sm text-gray-500 truncate">We're open 24/7 for
emergencies</p>
                                    <svg class="w-4 h-4 text-primary-500" fill="currentColor" viewBox="0 0 20
20">
                                        <path fill-rule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0
01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clip-
rule="evenodd"/>
                                    </svg>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- Individual Chat Screen -->
    <div id="chatScreen" class="fixed inset-0 bg-black/50 backdrop-blur-sm hidden z-50">
        <div class="h-full flex flex-col bg-white">
            <!-- Chat Header -->
            <div class="flex items-center justify-between px-4 py-3 glass-surface">
                <div class="flex items-center space-x-3">
                    <button onclick="backToChatList()" class="p-2 hover:bg-white/20 rounded-full transition-colors">
                        <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15 19l-7-7 7-7"/>
                        </svg>
                    </button>
                    <div class="flex items-center space-x-3">
                        <div class="w-10 h-10 card-3d rounded-full flex items-center justify-center">
                            <span class="text-primary-600 font-medium text-sm" id="chatInitials">CF</span>
                        </div>
                        <div>
                            <h3 class="font-semibold text-white" id="chatPharmacyName">CareFirst Pharmacy</h3>
                            <p class="text-xs text-white/80">Online</p>
                        </div>
                    </div>
                </div>
                <div class="flex items-center space-x-2">
                    <button class="p-2 hover:bg-white/20 rounded-full transition-colors">
                        <svg class="w-5 h-5 text-white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M3 5a2 2 0 012-2h3.28a1 1 0 01.948.684l1.498 4.493a1 1 0 01-.502 1.21l-2.257 1.13a11.042 11.042 0 005.516 5.516l1.13-2.257a1 1 0 011.21-.502l4.493 1.498a1 1 0 01.684.949V19a2 2 0 01-2 2h-1C9.716 21 3 14.284 3 6V5z"/>
                        </svg>
                    </button>
                    <button class="p-2 hover:bg-white/20 rounded-full transition-colors">
                        <svg class="w-5 h-5 text-white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15 10l4.553-2.276A1 1 0 0121 8.618v6.764a1 1 0 01-1.447.894L15 14M5 18h8a2 2 0 002-2V8a2 2 0 00-2-2H5a2 2 0 00-2 2v8a2 2 0 002 2z"/>
                        </svg>
                    </button>
                </div>
            </div>

            <!-- Chat Messages -->
            <div class="flex-1 overflow-y-auto p-4 space-y-4" id="chatMessages" style="background: linear-gradient(135deg, #f8fafc 0%, #e2e8f0 100%);">
                <!-- Messages will be populated here -->
            </div>

            <!-- Chat Input -->
```

```html
            <div class="p-4 bg-white border-t border-gray-200">
                <div class="flex items-center space-x-3">
                    <button class="p-2 hover:bg-gray-100 rounded-full transition-colors">
                        <svg class="w-6 h-6 text-gray-500" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15.172 7l-6.586 6.586a2 2 0 102.828 2.828l6.414-6.586a4 4 0 00-5.656-5.656l-6.415 6.585a6 6 0 108.486 8.486L20.5 13"/>
                        </svg>
                    </button>
                    <div class="flex-1 relative">
                        <input
                            type="text"
                            placeholder="Type a message..."
                            class="w-full py-3 px-4 pr-12 bg-gray-100 rounded-full text-gray-800 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-primary-500"
                            id="chatInput"
                        >
                        <button class="absolute right-3 top-1/2 transform -translate-y-1/2 p-1 hover:bg-gray-200 rounded-full transition-colors">
                            <svg class="w-5 h-5 text-gray-500" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M14.828 14.828a4 4 0 01-5.656 0M9 10h1m4 0h1m-6 4h8m-9 5a9 9 0 1118 0 9 9 0 01-18 0z"/>
                            </svg>
                        </button>
                    </div>
                    <button onclick="sendMessage()" class="button-3d text-white p-3 rounded-full">
                        <svg class="w-5 h-5" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 19l9 2-9-18-9 18 9-2zm0 0v-8"/>
                        </svg>
                    </button>
                </div>
            </div>
        </div>
    </div>

    <!-- Camera Modal -->
    <div id="cameraModal" class="fixed inset-0 bg-black/95 hidden z-50">
        <div class="flex flex-col h-full">
            <div class="flex items-center justify-between p-4">
                <button onclick="closeCameraModal()" class="p-2 hover:bg-gray-700 rounded-full">
                    <svg class="w-6 h-6 text-white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12"/>
                    </svg>
                </button>
                <h2 class="text-lg font-medium text-white">Scan Prescription</h2>
                <div class="w-10"></div>
            </div>

            <div class="flex-1 flex items-center justify-center px-6">
                <div class="w-64 h-48 border-2 border-dashed border-primary-400 rounded-lg flex items-center justify-center">
                    <div class="text-center">
                        <svg class="w-12 h-12 text-primary-400 mx-auto mb-3" fill="none" stroke="currentColor" viewBox="0 0 24 24">
```

```
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M3
9a2 2 0 012-2h.93a2 2 0 001.664-.89l.812-1.22A2 2 0 0110.07 4h3.86a2 2 0 011.664.89l.812
1.22A2 2 0 0018.07 7H19a2 2 0 012 2v9a2 2 0 01-2 2H5a2 2 0 01-2-2V9z"/>
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15
13a3 3 0 11-6 0 3 3 0 016 0z"/>
                        </svg>
                        <p class="text-white text-base mb-1">Position prescription here</p>
                        <p class="text-gray-400 text-xs">Camera will activate automatically</p>
                    </div>
                </div>
            </div>

            <div class="p-4">
                <button onclick="captureImage()" class="w-full button-3d text-white py-3 rounded-xl
font-medium">
                    Capture Image
                </button>
            </div>
        </div>
    </div>

    <script>
        // Sample medicine data for autocomplete
        const medicines = [
            'Paracetamol 500mg', 'Paracetamol 650mg', 'Panadol', 'Panadol Extra',
            'Amoxicillin 250mg', 'Amoxicillin 500mg', 'Amoxil', 'Augmentin',
            'Ibuprofen 200mg', 'Ibuprofen 400mg', 'Advil', 'Brufen',
            'Aspirin 75mg', 'Aspirin 325mg', 'Disprin', 'Ecosprin',
            'Cetirizine 10mg', 'Cetrizet', 'Zyrtec', 'Alerid',
            'Omeprazole 20mg', 'Omez', 'Prilosec', 'Losec',
            'Metformin 500mg', 'Glucophage', 'Glycomet', 'Diabex',
            'Atorvastatin 10mg', 'Lipitor', 'Atorlip', 'Storvas',
            'Cough Syrup', 'Benadryl', 'Robitussin', 'Dextromethorphan',
            'Vitamin D3', 'Calcium', 'Iron tablets', 'Multivitamin'
        ];

        // Sample search results data
        const searchResults = [
            {
                id: 1,
                drugName: "Paracetamol 500mg",
                price: "GHS 8.50",
                pharmacy: "CareFirst Pharmacy",
                location: "1.2 km away",
                availability: "In Stock",
                rating: 4.8
            },
            {
                id: 2,
                drugName: "Amoxicillin 250mg",
                price: "GHS 15.75",
                pharmacy: "HealthPlus Pharmacy",
                location: "0.8 km away",
                availability: "In Stock",
                rating: 4.6
            },
            {
                id: 3,
                drugName: "Paracetamol 650mg",
                price: "GHS 12.00",
                pharmacy: "MediCare Center",
                location: "2.1 km away",
                availability: "Low Stock",
                rating: 4.7
            },
            {
                id: 4,
                drugName: "Amoxicillin 500mg",
                price: "GHS 28.90",
                pharmacy: "Wellness Pharmacy",
                location: "1.5 km away",
                availability: "In Stock",
                rating: 4.9
            },
            {
                id: 5,
                drugName: "Paracetamol Syrup",
                price: "GHS 18.25",
                pharmacy: "QuickMed Pharmacy",
                location: "3.2 km away",
                availability: "In Stock",
                rating: 4.5
            },
            {
                id: 6,
                drugName: "Amoxicillin Capsules",
                price: "GHS 22.50",
                pharmacy: "CityMed Pharmacy",
                location: "2.8 km away",
                availability: "In Stock",
                rating: 4.4
            }
        ];

        let selectedDrugs = [];
        let currentPage = 1;
        const itemsPerPage = 4;
        let currentActiveDrug = null;
        let searchedDrugs = [];
        let isMultipleSearch = false;
        let userProfile = {
            firstName: '',
            lastName: '',
            email: '',
            contact: '',
            isRegistered: false,
            chatsUsed: 0
        };

        const searchInput = document.getElementById('searchInput');
        const autocompleteDropdown = document.getElementById('autocompleteDropdown');
        const suggestionsList = document.getElementById('suggestionsList');
        const selectedDrugsContainer = document.getElementById('selectedDrugs');
        const searchScreen = document.getElementById('searchScreen');
        const resultsScreen = document.getElementById('resultsScreen');

        // Search input event listeners
        searchInput.addEventListener('input', function(e) {
            const query = e.target.value.trim();
            if (query.length > 0) {
```

```
                showAutocomplete(query);
            } else {
                hideAutocomplete();
            }
        });

        // Show autocomplete suggestions
        function showAutocomplete(query) {
            const filteredMedicines = medicines.filter(med =>
                med.toLowerCase().includes(query.toLowerCase()) &&
                !selectedDrugs.includes(med)
            ).slice(0, 6);

            if (filteredMedicines.length > 0) {
                suggestionsList.innerHTML = filteredMedicines.map((med, index) => `
                    <div class="suggestion-item px-4 py-3 cursor-pointer transition-all duration-150
hover:bg-primary-50"
                         onclick="selectSuggestion('${med}')"
                         style="animation-delay: ${index * 0.02}s">
                        <div class="flex items-center space-x-3">
                            <svg class="w-4 h-4 text-white/70" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M19.428 15.428a2 2 0 00-1.022-.547l-2.387-.477a6 6 0 00-3.86.517l-.318.158a6 6 0
01-3.86.517L6.05 15.21a2 2 0 00-1.806.547M8 4h8l-1 1v5.172a2 2 0 00.586 1.414l5 5c1.26
1.26.367 3.414-1.415 3.414H4.828c-1.782 0-2.674-2.154-1.414-3.414l5-5A2 2 0 009 7.172V5L8
4z"/>
                            </svg>
                            <span class="text-white font-normal">${med}</span>
                        </div>
                    </div>
                `).join('');
                autocompleteDropdown.classList.remove('hidden');
            } else {
                hideAutocomplete();
            }
        }

        // Hide autocomplete
        function hideAutocomplete() {
            autocompleteDropdown.classList.add('hidden');
        }

        // Select suggestion
        function selectSuggestion(medicine) {
            if (isMultipleSearch) {
                addDrug(medicine);
                searchInput.value = '';
            } else {
                // Single search - execute immediately
                performSingleSearch(medicine);
            }
            hideAutocomplete();
        }

        // Toggle multiple search
        function toggleMultipleSearch() {
            isMultipleSearch = !isMultipleSearch;
            const toggle = document.getElementById('multipleToggle');
            const knob = document.getElementById('toggleKnob');
            if (isMultipleSearch) {
                toggle.classList.add('active');
                knob.style.transform = 'translateX(24px)';
                selectedDrugsContainer.classList.remove('hidden');
            } else {
                toggle.classList.remove('active');
                knob.style.transform = 'translateX(0)';
                selectedDrugsContainer.classList.add('hidden');
                selectedDrugs = [];
                updateDrugsList();
            }
        }

        // Add drug to list (multiple search mode)
        function addDrug(drugName) {
            if (!selectedDrugs.includes(drugName)) {
                selectedDrugs.push(drugName);
                updateDrugsList();
            }
        }

        // Update drugs list display
        function updateDrugsList() {
            const drugsList = document.getElementById('drugsList');
            drugsList.innerHTML = selectedDrugs.map(drug => `
                <div class="pill-tag flex items-center justify-between text-white px-4 py-2 rounded-xl
text-sm">
                    <span class="font-medium">${drug}</span>
                    <button onclick="removeDrug('${drug}')" class="ml-3 p-1 hover:bg-white/20
rounded-full transition-colors">
                        <svg class="w-4 h-4" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6
18L18 6M6 6l12 12"/>
                        </svg>
                    </button>
                </div>
            `).join('');
        }

        // Remove drug from list
        function removeDrug(drugName) {
            selectedDrugs = selectedDrugs.filter(drug => drug !== drugName);
            updateDrugsList();
        }

        // Perform single search
        function performSingleSearch(medicine) {
            searchedDrugs = [medicine];
            showResults([medicine]);
        }

        // Perform multiple search
        function performSearch() {
            if (selectedDrugs.length === 0) return;

            // Show loading state
            const searchButton = document.getElementById('searchButton');
            const searchButtonText = document.getElementById('searchButtonText');
            const searchSpinner = document.getElementById('searchSpinner');
            searchButton.disabled = true;
            searchButtonText.textContent = 'Searching nearby pharmacies...';
            searchSpinner.classList.remove('hidden');

            setTimeout(() => {
                showResults(selectedDrugs);

                // Reset button state
                searchButton.disabled = false;
                searchButtonText.textContent = 'Search Medicines';
                searchSpinner.classList.add('hidden');
            }, 1800);
        }

        // Show results screen
        function showResults(searchTerms) {
            searchedDrugs = [...searchTerms];

            document.getElementById('resultsContainer').innerHTML = '';
            currentPage = 1;

            searchScreen.classList.add('slide-out-left');

            setTimeout(() => {
                searchScreen.classList.add('hidden');
                resultsScreen.classList.remove('hidden', 'slide-in-right');
                resultsScreen.classList.add('slide-in-active');

                setTimeout(() => {
                    createDrugTabs(searchedDrugs);
                    if (searchedDrugs.length > 1) {
                        showAllResults();
                    } else if (searchedDrugs.length > 0) {
                        showDrugResults(searchedDrugs[0]);
                    }
                }, 100);
            }, 200);
        }

        // Create drug tabs
        function createDrugTabs(drugs) {
            const tabsContainer = document.getElementById('drugTabs');
            let tabsHTML = '';

            // Add "All" tab if multiple drugs
            if (drugs.length > 1) {
                tabsHTML += `
                    <button onclick="showAllResults()"
                            class="drug-tab active px-4 py-2 rounded-full font-medium text-sm whitespace-nowrap flex-shrink-0"
                            id="tab-all"
                            style="animation-delay: 0s">
                        <span class="text-white">All</span>
                    </button>
                `;
            }

            // Add individual drug tabs
            tabsHTML += drugs.map((drug, index) => `
                <button onclick="showDrugResults('${drug}')"
                        class="drug-tab ${drugs.length === 1 && index === 0 ? 'active' : ''} px-4 py-2
rounded-full font-medium text-sm whitespace-nowrap flex-shrink-0"
                        id="tab-${drug.replace(/\s+/g, '-')}"
                        style="animation-delay: ${(index + (drugs.length > 1 ? 1 : 0)) * 0.1}s">
                    <span class="${drugs.length === 1 && index === 0 ? 'text-white' : 'text-
gray-700'}">${drug}</span>
                </button>
            `).join('');

            tabsContainer.innerHTML = tabsHTML;
        }

        // Show all results (for multiple drug search)
        function showAllResults() {
            currentActiveDrug = 'all';

            // Update active tab
            document.querySelectorAll('.drug-tab').forEach(tab => {
                tab.classList.remove('active');
                tab.querySelector('span').classList.remove('text-white');
                tab.querySelector('span').classList.add('text-gray-700');
            });

            const allTab = document.getElementById('tab-all');
            if (allTab) {
                allTab.classList.add('active');
                allTab.querySelector('span').classList.remove('text-gray-700');
                allTab.querySelector('span').classList.add('text-white');
            }

            // Get all unique pharmacies and calculate availability
            const pharmacyResults = {};

            searchedDrugs.forEach(drugName => {
                const drugResults = searchResults.filter(result =>
                    result.drugName.toLowerCase().includes(drugName.toLowerCase()) ||
                    drugName.toLowerCase().includes(result.drugName.toLowerCase().split(' ')[0])
                );

                drugResults.forEach(result => {
                    if (!pharmacyResults[result.pharmacy]) {
                        pharmacyResults[result.pharmacy] = {
                            pharmacy: result.pharmacy,
                            location: result.location,
                            rating: result.rating,
                            availableDrugs: [],
                            totalSearched: searchedDrugs.length
                        };
                    }

                    // Add drug if not already added
                    const existingDrug = pharmacyResults[result.pharmacy].availableDrugs.find(d =>
d.drugName === result.drugName);
                    if (!existingDrug) {
                        pharmacyResults[result.pharmacy].availableDrugs.push({
                            drugName: result.drugName,
                            price: result.price,
                            availability: result.availability
                        });
                    }
                });
            });

            // Convert to array and sort by availability count
            const allResults = Object.values(pharmacyResults).sort((a, b) => b.availableDrugs.length - 
a.availableDrugs.length);

            // Clear and display results
            document.getElementById('resultsContainer').innerHTML = '';
            currentPage = 1;

            if (allResults.length > 0) {
                displayAllResults(allResults.slice(0, itemsPerPage));
                document.getElementById('loadMoreBtn').style.display = allResults.length > 
itemsPerPage ? 'block' : 'none';
            } else {
                // Show no results message
                document.getElementById('resultsContainer').innerHTML = `
                    <div class="text-center py-12">
                        <div class="w-16 h-16 mx-auto mb-4 bg-gray-100 rounded-full flex items-center 
justify-center">
                            <svg class="w-8 h-8 text-gray-500" fill="none" stroke="currentColor" 
viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" 
d="M9.172 16.172a4 4 0 015.656 0M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 
012-2h5.586a1 1 0 01.707.293l5.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"/>
                            </svg>
                        </div>
                        <h3 class="text-lg font-medium text-gray-900 mb-2">No results found</h3>
                        <p class="text-gray-600">No pharmacies have the searched medicines in stock 
nearby.</p>
                    </div>
                `;
                document.getElementById('loadMoreBtn').style.display = 'none';
            }
        }

        // Show results for specific drug
        function showDrugResults(drugName) {
            currentActiveDrug = drugName;

            // Update active tab
            document.querySelectorAll('.drug-tab').forEach(tab => {
                tab.classList.remove('active');
                tab.querySelector('span').classList.remove('text-white');
                tab.querySelector('span').classList.add('text-gray-700');
            });

            const activeTab = document.getElementById(`tab-${drugName.replace(/\s+/g, '-')}`);
            if (activeTab) {
                activeTab.classList.add('active');
                activeTab.querySelector('span').classList.remove('text-gray-700');
                activeTab.querySelector('span').classList.add('text-white');
            }

            // Filter results for this specific drug
            const drugResults = searchResults.filter(result => 
                result.drugName.toLowerCase().includes(drugName.toLowerCase()) ||
                drugName.toLowerCase().includes(result.drugName.toLowerCase().split(' ')[0])
            );

            // Clear and display results
            document.getElementById('resultsContainer').innerHTML = '';
            currentPage = 1;

            if (drugResults.length > 0) {
                displayResults(drugResults.slice(0, itemsPerPage));
                document.getElementById('loadMoreBtn').style.display = drugResults.length >
itemsPerPage ? 'block' : 'none';
            } else {
                // Show no results message
                document.getElementById('resultsContainer').innerHTML = `
                    <div class="text-center py-12">
                        <div class="w-16 h-16 mx-auto mb-4 bg-gray-100 rounded-full flex items-center
justify-center">
                            <svg class="w-8 h-8 text-gray-500" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M9.172 16.172a4 4 0 015.656 0M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0
012-2h5.586a1 1 0 01.707.293l5.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"/>
                            </svg>
                        </div>
                        <h3 class="text-lg font-medium text-gray-900 mb-2">No results found</h3>
                        <p class="text-gray-600">No pharmacies have "${drugName}" in stock nearby.</
p>
                    </div>
                `;
                document.getElementById('loadMoreBtn').style.display = 'none';
            }
        }

        // Display results
        function displayResults(results) {
            const container = document.getElementById('resultsContainer');

            results.forEach((result, index) => {
                const resultCard = createResultCard(result, index);
                container.appendChild(resultCard);
            });
        }

        // Display all results (for multiple drug search)
        function displayAllResults(results) {
            const container = document.getElementById('resultsContainer');

            results.forEach((result, index) => {
                const resultCard = createAllResultCard(result, index);
                container.appendChild(resultCard);
            });
        }

        // Create result card for all results view
        function createAllResultCard(result, index) {
            const card = document.createElement('div');
            card.className = 'result-card rounded-2xl p-4';
            card.style.animationDelay = `${index * 0.1}s`;

            // Calculate availability status
            const availableCount = result.availableDrugs.length;
```

```
    const totalCount = result.totalSearched;
    const availabilityText = `${availableCount}/${totalCount}`;
    const availabilityColor = availableCount === totalCount ? 'text-green-600' : 'text-orange-600';

    // Get price range
    const prices = result.availableDrugs.map(drug => parseFloat(drug.price.replace('GHS ', '')));
    const minPrice = Math.min(...prices);
    const maxPrice = Math.max(...prices);
    const priceRange = minPrice === maxPrice ? `GHS ${minPrice.toFixed(2)}` : `GHS ${minPrice.toFixed(2)} - ${maxPrice.toFixed(2)}`;

    card.innerHTML = `
      <div class="flex items-center justify-between mb-3">
        <div class="flex items-center space-x-3">
          <div class="w-12 h-12 bg-blue-100 rounded-xl flex items-center justify-center flex-shrink-0">
            <span class="text-2xl">🏪</span>
          </div>
          <div class="min-w-0 flex-1">
            <h3 class="font-bold text-gray-900 text-base truncate">${result.pharmacy}</h3>
            <div class="flex items-center space-x-2 mt-1">
              <span class="text-base font-bold text-green-600">${priceRange}</span>
              <span class="px-2 py-1 text-xs rounded-full font-medium ${availableCount === totalCount ? 'bg-green-100 text-green-800' : 'bg-orange-100 text-orange-800'}">${availabilityText}</span>
            </div>
          </div>
        </div>
        <div class="flex items-center space-x-1 flex-shrink-0">
          <svg class="w-4 h-4 text-yellow-400" fill="currentColor" viewBox="0 0 20 20">
            <path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07 3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8 2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54 1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98 8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-3.292z"/>
          </svg>
          <span class="text-sm text-gray-600 font-medium">${result.rating}</span>
        </div>
      </div>

      <div class="flex items-center space-x-2 mb-3">
        <span class="text-base">📍</span>
        <span class="text-gray-600 text-sm">${result.location}</span>
      </div>

      <!-- Available Drugs List -->
      <div class="mb-3">
        <h4 class="text-sm font-medium text-gray-700 mb-2">Available medicines:</h4>
        <div class="space-y-1">
          ${result.availableDrugs.map(drug => `
            <div class="flex items-center justify-between py-2 px-3 bg-gray-50 rounded-lg">
              <div class="flex items-center space-x-2 min-w-0 flex-1">
                <span class="text-base">💊</span>
                <span class="text-sm font-medium text-gray-800 truncate">${drug.drugName}</span>
```

```
                                </div>
                                <div class="flex items-center space-x-2 flex-shrink-0">
                                    <span class="text-sm font-bold text-green-600">${drug.price}</span>
                                    <span class="text-xs px-2 py-1 rounded-full ${drug.availability === 'In Stock' ? 'bg-green-100 text-green-800' : 'bg-yellow-100 text-yellow-800'}">${drug.availability}</span>
                                </div>
                            </div>
                        `).join('')}
                    </div>
                </div>

                <div class="flex">
                    <button onclick="startChat('${result.pharmacy}')" class="w-full bg-blue-600 hover:bg-blue-700 text-white py-2.5 px-4 rounded-xl font-semibold flex items-center justify-center space-x-2 transition-colors">
                        <span class="text-base">💬</span>
                        <span>Chat</span>
                    </button>
                </div>
            `;

            return card;
        }

        // Create result card
        function createResultCard(result, index) {
            const card = document.createElement('div');
            card.className = 'result-card rounded-2xl p-4';
            card.style.animationDelay = `${index * 0.1}s`;

            card.innerHTML = `
                <div class="flex items-center justify-between mb-3">
                    <div class="flex items-center space-x-3">
                        <div class="w-12 h-12 bg-blue-100 rounded-xl flex items-center justify-center flex-shrink-0">
                            <span class="text-2xl">💊</span>
                        </div>
                        <div class="min-w-0 flex-1">
                            <h3 class="font-bold text-gray-900 text-base truncate">${result.drugName}</h3>
                            <div class="flex items-center space-x-2 mt-1">
                                <span class="text-base font-bold text-green-600">${result.price}</span>
                                <span class="px-2 py-1 text-xs rounded-full font-medium ${result.availability === 'In Stock' ? 'bg-green-100 text-green-800' : 'bg-yellow-100 text-yellow-800'}">${result.availability}</span>
                            </div>
                        </div>
                    </div>
                    <div class="flex items-center space-x-1 flex-shrink-0">
                        <svg class="w-4 h-4 text-yellow-400" fill="currentColor" viewBox="0 0 20 20">
                            <path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07 3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8 2.034a1 1 0 00-.364 1.118l1.07 3.292c.3.921-.755 1.688-1.54 1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-1.838-.197-1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118L2.98 8.72c-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-3.292z"/>
                        </svg>
                        <span class="text-sm text-gray-600 font-medium">${result.rating}</span>
                    </div>
        </div>

        <div class="flex items-center space-x-2 mb-2">
          <span class="text-base">🏪</span>
          <span class="font-semibold text-gray-800 text-sm">${result.pharmacy}</span>
        </div>

        <div class="flex items-center space-x-2 mb-3">
          <span class="text-base">📍</span>
          <span class="text-gray-600 text-sm">${result.location}</span>
        </div>

        <div class="flex">
          <button onclick="startChat('${result.pharmacy}')" class="w-full bg-blue-600
hover:bg-blue-700 text-white py-2.5 px-4 rounded-xl font-semibold flex items-center justify-
center space-x-2 transition-colors">
            <span class="text-base">💬</span>
            <span>Chat</span>
          </button>
        </div>
      `;

      return card;
    }

    // Load more results
    function loadMore() {
      if (!currentActiveDrug) return;

      if (currentActiveDrug === 'all') {
        // Load more for all results view
        const pharmacyResults = {};

        searchedDrugs.forEach(drugName => {
          const drugResults = searchResults.filter(result =>
            result.drugName.toLowerCase().includes(drugName.toLowerCase()) ||
            drugName.toLowerCase().includes(result.drugName.toLowerCase().split(' ')[0])
          );

          drugResults.forEach(result => {
            if (!pharmacyResults[result.pharmacy]) {
              pharmacyResults[result.pharmacy] = {
                pharmacy: result.pharmacy,
                location: result.location,
                rating: result.rating,
                availableDrugs: [],
                totalSearched: searchedDrugs.length
              };
            }

            const existingDrug = pharmacyResults[result.pharmacy].availableDrugs.find(d =>
d.drugName === result.drugName);
            if (!existingDrug) {
              pharmacyResults[result.pharmacy].availableDrugs.push({
                drugName: result.drugName,
                price: result.price,
                availability: result.availability
              });
            }
```

```
                    });
                });

                const allResults = Object.values(pharmacyResults).sort((a, b) => b.availableDrugs.length
- a.availableDrugs.length);
                const startIndex = currentPage * itemsPerPage;
                const endIndex = startIndex + itemsPerPage;
                const nextResults = allResults.slice(startIndex, endIndex);

                if (nextResults.length > 0) {
                    displayAllResults(nextResults);
                    currentPage++;

                    if (endIndex >= allResults.length) {
                        document.getElementById('loadMoreBtn').style.display = 'none';
                    }
                }
            } else {
                // Load more for individual drug results
                const drugResults = searchResults.filter(result =>
                    result.drugName.toLowerCase().includes(currentActiveDrug.toLowerCase()) ||
                    currentActiveDrug.toLowerCase().includes(result.drugName.toLowerCase().split(' ')[0])
                );

                const startIndex = currentPage * itemsPerPage;
                const endIndex = startIndex + itemsPerPage;
                const nextResults = drugResults.slice(startIndex, endIndex);

                if (nextResults.length > 0) {
                    displayResults(nextResults);
                    currentPage++;

                    if (endIndex >= drugResults.length) {
                        document.getElementById('loadMoreBtn').style.display = 'none';
                    }
                }
            }
        }

        // Go back to search
        function goBackToSearch() {
            resultsScreen.classList.remove('slide-in-active');
            resultsScreen.classList.add('slide-in-right');

            setTimeout(() => {
                resultsScreen.classList.add('hidden');
                searchScreen.classList.remove('hidden', 'slide-out-left');
                searchScreen.classList.add('slide-in-active');

                searchInput.value = '';
                hideAutocomplete();

                setTimeout(() => {
                    searchScreen.classList.remove('slide-in-active');
                }, 100);
            }, 200);
        }

        // Profile dropdown functions
        function toggleProfileDropdown() {
            const dropdown = document.getElementById('profileDropdown');
            dropdown.classList.toggle('hidden');
        }

        // Profile functions
        function showProfile() {
            document.getElementById('profileDropdown').classList.add('hidden');
            document.getElementById('profileModal').classList.remove('hidden');
            showProfileTab('profile');
        }

        function closeProfile() {
            document.getElementById('profileModal').classList.add('hidden');
        }

        // Chat data
        const chatData = {
            'CareFirst Pharmacy': {
                initials: 'CF',
                messages: [
                    { type: 'received', text: 'Hello! How can I help you today?', time: '2:30 PM' },
                    { type: 'sent', text: 'Hi! Do you have Paracetamol 500mg in stock?', time: '2:31 PM' },
                    { type: 'received', text: 'Yes, we have Paracetamol 500mg available. How many do
you need?', time: '2:32 PM' },
                    { type: 'sent', text: 'I need a pack of 20 tablets. What\'s the price?', time: '2:33 PM' }
                ]
            },
            'HealthPlus Pharmacy': {
                initials: 'HP',
                messages: [
                    { type: 'received', text: 'Welcome to HealthPlus! How can we assist you?', time:
'Yesterday' },
                    { type: 'sent', text: 'I ordered Amoxicillin yesterday. Is it ready?', time: 'Yesterday' },
                    { type: 'received', text: 'Your order is ready for pickup', time: 'Yesterday' }
                ]
            },
            'MediCare Center': {
                initials: 'MC',
                messages: [
                    { type: 'received', text: 'Thank you for choosing MediCare!', time: 'Monday' },
                    { type: 'sent', text: 'Thank you for the excellent service!', time: 'Monday' },
                    { type: 'received', text: 'Thank you for your purchase!', time: 'Monday' }
                ]
            },
            'Wellness Pharmacy': {
                initials: 'WP',
                messages: [
                    { type: 'received', text: 'Hello! We\'re here to help 24/7', time: 'Sunday' },
                    { type: 'sent', text: 'Do you deliver medications?', time: 'Sunday' },
                    { type: 'received', text: 'We\'re open 24/7 for emergencies', time: 'Sunday' }
                ]
            }
        };

        let currentChatPharmacy = null;

        // Chat functions
        function showChat() {
            document.getElementById('profileDropdown').classList.add('hidden');
            document.getElementById('chatListScreen').classList.remove('hidden');
        }

        function closeChatListScreen() {
            document.getElementById('chatListScreen').classList.add('hidden');
        }

        function openChat(pharmacyName, initials) {
            currentChatPharmacy = pharmacyName;

            // Update chat header
            document.getElementById('chatPharmacyName').textContent = pharmacyName;
            document.getElementById('chatInitials').textContent = initials;

            // Load chat messages
            loadChatMessages(pharmacyName);

            // Show individual chat screen
            document.getElementById('chatListScreen').classList.add('hidden');
            document.getElementById('chatScreen').classList.remove('hidden');
        }

        function backToChatList() {
            document.getElementById('chatScreen').classList.add('hidden');
            document.getElementById('chatListScreen').classList.remove('hidden');
        }

        function loadChatMessages(pharmacyName) {
            const chatMessages = document.getElementById('chatMessages');
            const chat = chatData[pharmacyName];

            if (!chat) return;

            chatMessages.innerHTML = '';

            // Add existing messages
            chat.messages.forEach(message => {
                const messageDiv = document.createElement('div');

                if (message.type === 'sent') {
                    messageDiv.className = 'flex items-end justify-end space-x-2';
                    messageDiv.innerHTML = `
                        <div class="chat-bubble-sent max-w-xs px-4 py-2 rounded-2xl rounded-br-md">
                            <p class="text-white text-sm">${message.text}</p>
                            <p class="text-xs text-white/80 mt-1 text-right">${message.time} ✓✓</p>
                        </div>
                    `;
                } else {
                    messageDiv.className = 'flex items-start space-x-2';
                    messageDiv.innerHTML = `
                        <div class="w-8 h-8 card-3d rounded-full flex items-center justify-center flex-shrink-0">
                            <span class="text-primary-600 font-medium text-xs">${chat.initials}</span>
                        </div>
                        <div class="chat-bubble-received max-w-xs px-4 py-2 rounded-2xl rounded-tl-md">
                            <p class="text-gray-800 text-sm">${message.text}</p>
                            <p class="text-xs text-gray-500 mt-1">${message.time}</p>
                        </div>
                    `;
                }
                chatMessages.appendChild(messageDiv);
            });

            // Add typing indicator
            const typingIndicator = document.createElement('div');
            typingIndicator.id = 'typingIndicator';
            typingIndicator.className = 'flex items-start space-x-2 hidden';
            typingIndicator.innerHTML = `
                <div class="w-8 h-8 card-3d rounded-full flex items-center justify-center flex-shrink-0">
                    <span class="text-primary-600 font-medium text-xs">${chat.initials}</span>
                </div>
                <div class="chat-bubble-received px-4 py-3 rounded-2xl rounded-tl-md">
                    <div class="flex space-x-1">
                        <div class="w-2 h-2 bg-gray-400 rounded-full animate-bounce"></div>
                        <div class="w-2 h-2 bg-gray-400 rounded-full animate-bounce" style="animation-
delay: 0.1s"></div>
                        <div class="w-2 h-2 bg-gray-400 rounded-full animate-bounce" style="animation-
delay: 0.2s"></div>
                    </div>
                </div>
            `;

            chatMessages.appendChild(typingIndicator);

            // Scroll to bottom
            chatMessages.scrollTop = chatMessages.scrollHeight;
        }

        function sendMessage() {
            const input = document.getElementById('chatInput');
            const message = input.value.trim();

            if (message && currentChatPharmacy) {
                const chatMessages = document.getElementById('chatMessages');
                const typingIndicator = document.getElementById('typingIndicator');
                const chat = chatData[currentChatPharmacy];

                // Add sent message to chat data
                chat.messages.push({
                    type: 'sent',
                    text: message,
                    time: new Date().toLocaleTimeString([], {hour: '2-digit', minute:'2-digit'})
                });

                // Add sent message to UI
                const messageDiv = document.createElement('div');
                messageDiv.className = 'flex items-end justify-end space-x-2';
                messageDiv.innerHTML = `
                    <div class="chat-bubble-sent max-w-xs px-4 py-2 rounded-2xl rounded-br-md">
                        <p class="text-white text-sm">${message}</p>
                        <p class="text-xs text-white/80 mt-1 text-right">${new
Date().toLocaleTimeString([], {hour: '2-digit', minute:'2-digit'})} ✓✓</p>
                    </div>
                `;

                chatMessages.insertBefore(messageDiv, typingIndicator);
                input.value = '';

                // Show typing indicator
                typingIndicator.classList.remove('hidden');

                // Scroll to bottom
                chatMessages.scrollTop = chatMessages.scrollHeight;

                // Simulate pharmacy response after 2 seconds
                setTimeout(() => {
                    const responses = [
                        "Thank you for your message. Let me check our inventory.",
                        "That item is available. Would you like to reserve it?",
                        "I'll prepare your order. When would you like to pick it up?",
                        "Great! Your order will be ready in 15 minutes.",
                        "We have that in stock. The price is GHS 12.50.",
                        "I can help you with that. What quantity do you need?"
                    ];

                    const randomResponse = responses[Math.floor(Math.random() * responses.length)];
                    const responseTime = new Date().toLocaleTimeString([], {hour: '2-digit', minute:'2-
digit'});

                    // Add response to chat data
                    chat.messages.push({
                        type: 'received',
                        text: randomResponse,
                        time: responseTime
                    });

                    // Hide typing indicator
                    typingIndicator.classList.add('hidden');

                    // Add response to UI
                    const responseDiv = document.createElement('div');
                    responseDiv.className = 'flex items-start space-x-2';
                    responseDiv.innerHTML = `
                        <div class="w-8 h-8 card-3d rounded-full flex items-center justify-center flex-
shrink-0">
                        <span class="text-primary-600 font-medium text-xs">${chat.initials}</span>
                        </div>
                        <div class="chat-bubble-received max-w-xs px-4 py-2 rounded-2xl rounded-tl-
md">
                        <p class="text-gray-800 text-sm">${randomResponse}</p>
                        <p class="text-xs text-gray-500 mt-1">${responseTime}</p>
                        </div>
                    `;

                    chatMessages.insertBefore(responseDiv, typingIndicator);
                    chatMessages.scrollTop = chatMessages.scrollHeight;
                }, 2000);
            }
        }

        function showProfileTab(tab) {
            // Profile tab is now the only tab, so no switching needed
            const profileContent = document.getElementById('profileTabContent');
            profileContent.classList.remove('hidden');
        }

        function saveProfile(event) {
            event.preventDefault();
```

```
            userProfile.firstName = document.getElementById('firstName').value;
            userProfile.lastName = document.getElementById('lastName').value;
            userProfile.email = document.getElementById('email').value;
            userProfile.contact = document.getElementById('contact').value;

            if (userProfile.firstName && userProfile.lastName && userProfile.email &&
userProfile.contact) {
                userProfile.isRegistered = true;
                document.getElementById('userStatus').textContent = `${userProfile.firstName} $
{userProfile.lastName}`;
                document.getElementById('chatLimit').textContent = 'Unlimited chats';
                showToast('Profile saved successfully!', 'success');
            } else {
                showToast('Please fill in all fields', 'error');
            }
        }

        // Start chat
        function startChat(pharmacy) {
            if (!userProfile.isRegistered && userProfile.chatsUsed >= 4) {
                showToast('Chat limit reached. Please register to continue.', 'error');
                showProfile();
                return;
            }

            if (!userProfile.isRegistered) {
                userProfile.chatsUsed++;
                document.getElementById('chatLimit').textContent = `${4 - userProfile.chatsUsed} chats
remaining this week`;
            }

            showToast(`Starting chat with ${pharmacy}...`, 'info');
        }

        // Camera functions
        function openCamera() {
            document.getElementById('cameraModal').classList.remove('hidden');
        }

        function closeCameraModal() {
            document.getElementById('cameraModal').classList.add('hidden');
        }

        function captureImage() {
            closeCameraModal();
            const detectedDrugs = ['Paracetamol 500mg', 'Amoxicillin 250mg'];
            if (isMultipleSearch) {
                detectedDrugs.forEach(drug => addDrug(drug));
                showToast('Prescription scanned! Drugs added to search.', 'success');
            } else {
                performSingleSearch(detectedDrugs[0]);
            }
        }

        // Upload file function
        function uploadFile() {
            const input = document.createElement('input');
            input.type = 'file';
            input.accept = 'image/*,.pdf';
            input.onchange = function(e) {
                const file = e.target.files[0];
                if (file) {
                    showToast(`Processing ${file.name}...`, 'info');
                    setTimeout(() => {
                        const detectedDrugs = ['Ibuprofen 400mg', 'Cetirizine 10mg'];
                        if (isMultipleSearch) {
                            detectedDrugs.forEach(drug => addDrug(drug));
                            showToast('File processed! Drugs added to search.', 'success');
                        } else {
                            performSingleSearch(detectedDrugs[0]);
                        }
                    }, 2000);
                }
            };
            input.click();
        }

        // Toggle map view
        function toggleMapView() {
            showToast('Map view coming soon!', 'info');
        }

        // Show toast notification
        function showToast(message, type) {
            const toast = document.createElement('div');
            const bgColor = type === 'success' ? 'bg-secondary-500' : type === 'error' ? 'bg-
red-500' : 'bg-primary-500';

            toast.className = `fixed top-20 left-1/2 transform -translate-x-1/2 ${bgColor} text-white
px-6 py-3 rounded-full shadow-lg z-50`;
            toast.style.animation = 'fadeIn 0.3s ease';
            toast.textContent = message;

            document.body.appendChild(toast);

            setTimeout(() => {
                toast.remove();
            }, 3000);
        }

        // Hide autocomplete and dropdown when clicking outside
        document.addEventListener('click', function(e) {
            if (!e.target.closest('.relative')) {
                hideAutocomplete();
            }

            // Close profile dropdown when clicking outside
            if (!e.target.closest('#profileButton') && !e.target.closest('#profileDropdown')) {
                document.getElementById('profileDropdown').classList.add('hidden');
            }
        });

        // Medicine rain animation
        function createMedicineRain() {
            const medicineRain = document.getElementById('medicineRain');
            const medicines = ['💊', '💉', '🧪', '⚕️', '🩹'];

            function createMedicineItem() {
                const item = document.createElement('div');
                item.className = 'medicine-item';
                item.textContent = medicines[Math.floor(Math.random() * medicines.length)];
                item.style.left = Math.random() * 100 + '%';
                item.style.animationDuration = (Math.random() * 2 + 2) + 's';
                item.style.animationDelay = Math.random() * 2 + 's';

                medicineRain.appendChild(item);

                // Remove item after animation
                setTimeout(() => {
                    if (item.parentNode) {
                        item.parentNode.removeChild(item);
                    }
                }, 4000);
            }

            // Create initial items
            for (let i = 0; i < 8; i++) {
                setTimeout(() => createMedicineItem(), i * 300);
            }

            // Continue creating items
            setInterval(createMedicineItem, 800);
        }

        // Initialize the app
        document.addEventListener('DOMContentLoaded', function() {
            // Start medicine rain animation
            createMedicineRain();

            // Add Enter key support for chat input
            document.getElementById('chatInput').addEventListener('keypress', function(e) {
                if (e.key === 'Enter') {
                    sendMessage();
                }
            });

            // Add entrance animation
            document.body.style.opacity = '0';
            setTimeout(() => {
                document.body.style.transition = 'opacity 0.5s ease';
                document.body.style.opacity = '1';
            }, 100);
        });
    </script>
<script>(function(){function c(){var b=a.contentDocument||a.contentWindow.document;if(b){var
d=b.createElement('script');d.innerHTML="window.__CF$cv$params={r:'9660fed903d39339',t:'M
Tc1MzY3MDI1NS4wMDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/
cdn-cgi/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head')
[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(document.body){var
a=document.createElement('iframe');a.height=1;a.width=1;a.style.position='absolute';a.style.top=
0;a.style.left=0;a.style.border='none';a.style.visibility='hidden';document.body.appendChild(a);if('l
oading'!==document.readyState)c();else
if(window.addEventListener)document.addEventListener('DOMContentLoaded',c);else{var
e=document.onreadystatechange||function(){};document.onreadystatechange=function(b)
{e(b);'loading'!==document.readyState&&(document.onreadystatechange=e,c())}}}})();</script></
body>
</html>
```