

```
<!DOCTYPE html>
<html lang="en" class="h-full">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>AgriFuture - Smart Farm Dashboard</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;50
0;600;700&display=swap" rel="stylesheet">
  <style>
    html {
      scroll-behavior: smooth;
      height: 100%;
      /* Massala background: warm layered gradients, subtle
grain, and animated saffron streak */
      background-color: #fff7ed; /* warm base */
      background-image:
        radial-gradient(1200px 600px at 8% 12%, rgba(255,
205, 65, 0.12), transparent 18%),
        radial-gradient(900px 450px at 92% 86%, rgba(245,
158, 11, 0.09), transparent 18%),
        linear-gradient(180deg, #fff7ed 0%, #ffefd5 35%,
#fff9e6 100%),
        radial-gradient(6px 6px at 10% 80%, rgba(194, 65,
12, 0.12), transparent 20%),
        radial-gradient(4px 4px at 23% 27%, rgba(212,
121, 42, 0.10), transparent 20%),
```

```
        radial-gradient(3px 3px at 76% 40%, rgba(184, 28,
23, 0.08), transparent 20%),
        radial-gradient(5px 5px at 56% 66%, rgba(245,
158, 11, 0.10), transparent 20%);
    background-blend-mode: normal, normal, normal,
overlay, overlay, overlay, overlay;
    background-size: cover, cover, cover, 1200px 800px,
900px 600px, 600px 400px, 800px 500px;
    position: relative;
    animation: bg-drift 20s linear infinite;
}

body::before {
    content: '';
    position: fixed;
    inset: 0;
    pointer-events: none;
    background-image: url("data:image/svg+xml;utf8,<svg
xmlns='http://www.w3.org/2000/svg' width='200'
height='200'><filter id='noise'><feTurbulence
baseFrequency='0.8' numOctaves='1' stitchTiles='stitch'
seed='4'/><feColorMatrix type='satuate'
values='0'/></filter><rect width='100%25' height='100%25'
filter='url(%23noise)' opacity='0.03' fill='white'/></svg>");
    opacity: 0.9;
    mix-blend-mode: multiply;
    z-index: -1;
}

body::after {
    content: '';
    position: fixed;
```

```
        inset: 0;
        pointer-events: none;
        background-image: linear-gradient(120deg, rgba(255,
184, 28, 0.06), rgba(255, 138, 0, 0.04));
        opacity: 0.95;
        transform: translate3d(0,0,0);
        animation: saffron-sweep 14s ease-in-out infinite;
        z-index: -1;
    }

    @keyframes bg-drift {
        0% { background-position: 0% 0%, 100% 100%, 0% 0%, 0%
0%, 0% 0%, 0% 0%, 0% 0%; }
        50% { background-position: 5% 2%, 95% 98%, 0% 3%, 2%
1%, 1% 2%, 4% 3%, 3% 5%; }
        100% { background-position: 0% 0%, 100% 100%, 0% 0%,
0% 0%, 0% 0%, 0% 0%, 0% 0%; }
    }

    @keyframes saffron-sweep {
        0% { transform: translateX(-10%) translateY(0)
rotate(-2deg); opacity: 0.9; }
        50% { transform: translateX(6%) translateY(1%)
rotate(1deg); opacity: 1; }
        100% { transform: translateX(-10%) translateY(0)
rotate(-2deg); opacity: 0.9; }
    }

    body {
        min-height: 100vh;
        overflow-y: auto;
        font-family: 'Inter', sans-serif;
```

```
        -webkit-font-smoothing: antialiased;
        -moz-osx-font-smoothing: grayscale;
    }

    button,
    input[type="checkbox"],
    #lang-toggle {
        cursor: pointer;
    }

    #farm-grid {
        display: grid;
        grid-template-columns: repeat(10, 1fr);
        grid-template-rows: repeat(10, 1fr);
        border: 2px solid rgba(34, 197, 94, 0.08);
        aspect-ratio: 1 / 1;
        width: 100%;
        max-width: 640px;
        margin: 0 auto;
        cursor: crosshair;
        background-image: linear-gradient(rgba(0,0,0,0.02),
        rgba(0,0,0,0.03)), radial-gradient(600px 400px at 50% 10%,
        rgba(255,255,255,0.03), transparent 40%),
        linear-gradient(180deg, rgba(255,255,255,0.02),
        rgba(0,0,0,0.02));
        background-blend-mode: overlay, overlay, normal;
        box-shadow: 0 10px 30px rgba(16,24,40,0.06), inset 0
        2px 8px rgba(255,255,255,0.02);
        border-radius: 12px;
        overflow: hidden;
        position: relative;
    }
```

```
.grid-cell {
    border: 0.5px solid rgba(0,0,0,0.06);
    transition: background-color 0.35s ease, transform
0.18s ease, box-shadow 0.18s ease;
    cursor: cell;
    background-repeat: no-repeat;
    background-position: center;
    background-size: 120% 120%;
    box-shadow: inset 0 0 0 1px rgba(255,255,255,0.02);
    background-image:
linear-gradient(rgba(255,255,255,0.03), rgba(0,0,0,0.01)),
radial-gradient(circle at 30% 20%, rgba(255,255,255,0.03),
transparent 20%);
    background-blend-mode: overlay, overlay, normal;
    position: relative;
    overflow: hidden;
}

.grid-cell:hover {
    transform: translateY(-3px) scale(1.02);
    z-index: 10;
    box-shadow: 0 8px 20px rgba(2,6,23,0.08), inset 0 0 0
1px rgba(255,255,255,0.03);
}

.grid-cell::before {
    content: "";
    position: absolute;
    inset: 0;
    pointer-events: none;
    background-image: repeating-linear-gradient(135deg,
rgba(255,255,255,0.02) 0 2px, rgba(0,0,0,0.02) 2px 6px);
```

```
        opacity: 0.35;
        mix-blend-mode: overlay;
    }

    .cb-ndvi-0 { background-color: #9f1239; }
    .cb-ndvi-1 { background-color: #dc2626; }
    .cb-ndvi-2 { background-color: #ea580c; }
    .cb-ndvi-3 { background-color: #f97316; }
    .cb-ndvi-4 { background-color: #facc15; }
    .cb-ndvi-5 { background-color: #a3e635; }
    .cb-ndvi-6 { background-color: #84cc16; }
    .cb-ndvi-7 { background-color: #22c55e; }
    .cb-ndvi-8 { background-color: #16a34a; }
    .cb-ndvi-9 { background-color: #15803d; }
    .cb-ndvi-10 { background-color: #166534; }

    .pulse-warn {
        animation: pulse-warn 2s cubic-bezier(0.4, 0, 0.6, 1)
infinite;
    }

    @keyframes pulse-warn {
        0%, 100% { opacity: 1; }
        50% { opacity: .7; }
    }

    .listening {
        animation: pulse-listen 1.5s infinite;
    }

    @keyframes pulse-listen {
        0% { transform: scale(1); box-shadow: 0 0 0 0
rgba(245, 158, 11, 0.7); }
```

```
        70% { transform: scale(1.05); box-shadow: 0 0 0 10px
rgba(245, 158, 11, 0); }
        100% { transform: scale(1); box-shadow: 0 0 0 0
rgba(245, 158, 11, 0); }
    }

#offline-banner { display: none; }

.spinner {
    border: 4px solid rgba(255, 255, 255, 0.3);
    border-left-color: #ffffff;
    border-radius: 50%;
    width: 20px;
    height: 20px;
    animation: spin 1s linear infinite;
}

@keyframes spin { 0% { transform: rotate(0deg); } 100% {
transform: rotate(360deg); } }

input[type="checkbox"]#offline-toggle {
-webkit-appearance: none; appearance: none; width: 3.5rem;
height: 2rem; background-color: #d1d5db; border-radius: 9999px;
position: relative; display: inline-flex; align-items: center;
transition: background-color 0.2s ease-in-out; }

    input[type="checkbox"]#offline-toggle::before { content:
''; height: 1.5rem; width: 1.5rem; background-color: white;
border-radius: 9999px; box-shadow: 0 1px 3px rgba(0,0,0,0.1);
transition: transform 0.2s ease-in-out; transform:
translateX(0.25rem); }

    input[type="checkbox"]:checked#offline-toggle {
background-color: #d97706; }
```

```

        input[type="checkbox"]:checked#offline-toggle::before {
transform: translateX(1.75rem); }

    /* floating clouds for a little motion */
    .cloud { position: fixed; top: -10%; left: -10%; width:
420px; height: 140px; background: linear-gradient(180deg,
rgba(255,255,255,0.9), rgba(250,250,250,0.75)); filter:
blur(12px) saturate(0.9); border-radius: 50%; opacity: 0.18;
transform: translate3d(0,0,0); z-index: 0; animation:
floatClouds 30s linear infinite; }

    .cloud.cloud-2 { width: 600px; height: 200px; top: 10%;
left: 50%; opacity: 0.12; animation-duration: 45s; }

    @keyframes floatClouds { 0% { transform:
translateX(-20vw) translateY(0) rotate(0deg); } 50% { transform:
translateX(60vw) translateY(6vh) rotate(1deg); } 100% {
transform: translateX(-20vw) translateY(0) rotate(0deg); } }

    /* ensure main has correct top padding to account for nav
height (nav uses h-20) */
    main.flex-1.pb-8.pt-16 {
        padding-top: 5rem; /* 80px = 5rem; override existing
pt-16 (64px) */
    }

    /* Make tooltip z-index above everything */
    #cell-tooltip { z-index: 60; }

    /* small satellite texture to layer on the farm grid */
    #farm-grid::after {
        content: "";
        position: absolute;
        inset: 0;

```



```

        pointer-events: none;
        background-image: url("data:image/svg+xml;utf8,\
          <svg xmlns='http://www.w3.org/2000/svg' width='200'
height='200'>\
            <filter id='grain'><feTurbulence
baseFrequency='0.9' numOctaves='1' stitchTiles='stitch'
seed='5'/><feColorMatrix type='saturate' values='0'/></filter>\
            <rect width='100%' height='100%'
filter='url(%23grain)' opacity='0.04' fill='white'/>\
            </svg>");
        mix-blend-mode: overlay;
        opacity: 0.9;
    }

</style>
</head>
<body class="h-full antialiased text-gray-800">
    <div id="offline-banner" class="fixed top-0 left-0 right-0
z-50 bg-yellow-500 text-center p-2 font-semibold
text-yellow-900">
        <span data-id="offline_banner">You are in Offline Mode.
Showing last saved data.</span>
    </div>

    <div class="min-h-full flex flex-col">
        <nav class="fixed top-0 left-0 right-0 z-40">
            <div class="backdrop-blur-sm bg-gradient-to-r
from-amber-50/70 via-white/60 to-amber-50/50 border-b
border-gray-200">
                <div class="max-w-7xl mx-auto px-4 sm:px-6
lg:px-8">

```

```

        <div class="flex items-center justify-between
h-20">

            <div class="flex items-center space-x-4">

                <div class="flex items-center
bg-white/60 backdrop-blur rounded-xl px-3 py-1 shadow-sm">

                    <svg class="h-8 w-8
text-amber-600" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">

                        <path d="M17.8182
10.6364C17.8182 10.6364 15.0909 13.3636 12 19C8.90909 13.3636
6.18182 10.6364 6.18182 10.6364C3.81818 8.27273 3.81818 4.72727
6.18182 2.36364C8.54545 0 12 3.81818 12 3.81818C12 3.81818
15.4545 0 17.8182 2.36364C20.1818 4.72727 20.1818 8.27273
17.8182 10.6364Z" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round"/>

                        <path d="M12 21V19"
stroke="currentColor" stroke-width="2" stroke-linecap="round"
stroke-linejoin="round"/>

                    </svg>

                    <div class="ml-3">

                        <h1 class="text-xl
font-extrabold text-gray-800 leading-tight"
data-il8n="app_title">AgriFuture</h1>

                        <div class="text-xs
text-gray-500">Smart Farm Dashboard</div>

                    </div>

                </div>

            </div>

            <div class="hidden sm:flex
sm:items-center sm:space-x-3 ml-4">

                <div class="px-3 py-2 bg-white/70
backdrop-blur rounded-lg shadow-sm text-xs">

```

```

                <div class="text-xxs
text-gray-500">Avg NDVI</div>
                <div id="nav-avg-ndvi"
class="font-semibold text-gray-800">--</div>
            </div>
            <div class="px-3 py-2 bg-white/70
backdrop-blur rounded-lg shadow-sm text-xs">
                <div class="text-xxs
text-gray-500">Moisture</div>
                <div id="nav-avg-moist"
class="font-semibold text-gray-800">--</div>
            </div>
            <div class="px-3 py-2 bg-white/70
backdrop-blur rounded-lg shadow-sm text-xs">
                <div class="text-xxs
text-gray-500">Temp</div>
                <div id="nav-current-temp"
class="font-semibold text-gray-800">--</div>
            </div>
        </div>
        <div class="flex items-center space-x-4">
            <button id="lang-toggle"
class="text-sm font-medium text-amber-700 hover:text-amber-800
transition-colors">
                <span
data-i18n="lang_toggle">हिन्दी</span>
            </button>

            <div class="flex items-center">

```

```

                <span class="text-sm font-medium
text-gray-700 mr-3" data-il8n="offline_toggle">Offline
Mode</span>

                <input id="offline-toggle"
type="checkbox" class="focus:outline-none focus:ring-2
focus:ring-amber-500 focus:ring-offset-2">

                </input>
            </div>

            <div class="flex items-center
bg-white/60 backdrop-blur rounded-full px-3 py-1 shadow-sm">

                

                <div class="text-sm font-medium
text-gray-800">Niel</div>

            </div>

        </div>

    </div>

</nav>

<main class="flex-1 pb-8 pt-16">

    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8
py-8">

        <div class="mb-6">

            <h2 class="text-2xl font-semibold
text-gray-900" data-il8n="dashboard_title">Farmer's
Dashboard</h2>

```

```

        <p class="mt-1 text-md text-gray-600"
data-i18n="dashboard_subtitle">Clear, actionable insights to
protect your yield.</p>
    </div>

    <div class="grid grid-cols-1 lg:grid-cols-3
gap-6">

        <div class="lg:col-span-2 space-y-6">
            <div class="bg-white shadow-lg rounded-lg
overflow-hidden">
                <div class="p-5 border-b
border-gray-200">
                    <div class="flex items-center">
                        <svg class="h-6 w-6
text-green-700" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">
                            <path
stroke-linecap="round" stroke-linejoin="round" d="M13.5
6H5.25A2.25 2.25 0 03 8.25v10.5A2.25 2.25 0 05.25 21h10.5A2.25
2.25 0 018 18.75V10.5m-10.5 6L21 3m0 0h-6.75M21 3v6.75" />
                        </svg>
                        <h3 class="ml-3 text-lg
font-medium leading-6 text-gray-900" data-i18n="map_title">Field
Health Heatmap (NDVI)</h3>
                    </div>
                    <p class="mt-1 text-sm
text-gray-500 ml-9" data-i18n="map_subtitle">Green is healthy,
Yellow/Orange is stressed.</p>
                </div>
                <div class="p-5">
                    <div id="farm-grid"
class="rounded-lg overflow-hidden"></div>

```

```

<div class="mt-4 text-sm
text-gray-600">

    <div class="flex-1 h-2
rounded-full mx-auto bg-gradient-to-r from-orange-500
via-yellow-400 to-green-700"></div>

    <div class="flex
justify-between mt-1">

        <span
data-il8n="stressed">Stressed</span>

        <span
data-il8n="healthy">Healthy</span>

    </div>
</div>
</div>
</div>

<div class="bg-white shadow-lg
rounded-lg">

    <div class="p-5 border-b
border-gray-200">

        <div class="flex items-center">

            <svg class="h-6 w-6
text-gray-700" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">

                <path
stroke-linecap="round" stroke-linejoin="round" d="M12.75
15l3-3m0 0l-3-3m3 3h-7.5M21 12a9 9 0 11-18 0 9 9 0 0118 0z" />

            </svg>

            <h3 class="ml-3 text-lg
font-medium leading-6 text-gray-900"
data-il8n="actions_title">Field Management</h3>

        </div>

```

```

        <p class="mt-1 text-sm
text-gray-500 ml-9" data-i18n="actions_subtitle">Apply
interventions or use voice commands.</p>
    </div>
    <div class="p-5 grid grid-cols-1
sm:grid-cols-4 gap-4">
        <button id="action-irrigate"
class="flex flex-col items-center justify-center p-4 bg-blue-600
text-white rounded-lg shadow-md hover:bg-blue-700
focus:outline-none focus:ring-2 focus:ring-offset-2
focus:ring-blue-600 transition-all transform hover:scale-105">
            <svg class="h-10 w-10"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24
24" stroke-width="1.5" stroke="currentColor">
                <path
stroke-linecap="round" stroke-linejoin="round" d="M15.375
17.55A.75.75 0 0114.625 18H9.375a.75.75 0
01-.75-.75V13.19l-4.26-4.576a.75.75 0 01.53-1.28h13.714a.75.75 0
01.53 1.28L15.375 13.19v4.36z" />
            </svg>
            <span class="mt-2 font-medium
text-sm" data-i18n="action_irrigate">Irrigate</span>
        </button>
        <button id="action-fertilize"
class="flex flex-col items-center justify-center p-4
bg-green-600 text-white rounded-lg shadow-md hover:bg-green-700
focus:outline-none focus:ring-2 focus:ring-offset-2
focus:ring-green-600 transition-all transform hover:scale-105">
            <svg class="h-10 w-10"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24
24" stroke-width="1.5" stroke="currentColor">

```

```

<path
stroke-linecap="round" stroke-linejoin="round" d="M20.25
7.5l-.625 10.632a2.25 2.25 0 01-2.247 2.118H6.622a2.25 2.25 0
01-2.247-2.118L3.75 7.5M10.5 11.25V18M13.5 11.25V18" />

<path
stroke-linecap="round" stroke-linejoin="round" d="M3.75
7.5h16.5v-1.5A1.5 1.5 0 0018.75 4.5H5.25A1.5 1.5 0 003.75
6v1.5z" />

</svg>
<span class="mt-2 font-medium
text-sm" data-il8n="action_fertilize">Fertilize</span>

</button>

<button id="action-pesticide"
class="flex flex-col items-center justify-center p-4 bg-red-600
text-white rounded-lg shadow-md hover:bg-red-700
focus:outline-none focus:ring-2 focus:ring-offset-2
focus:ring-red-600 transition-all transform hover:scale-105">
<svg class="h-10 w-10"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24
24" stroke-width="1.5" stroke="currentColor">
<path
stroke-linecap="round" stroke-linejoin="round" d="M19.5 14.25c0
1.313-1.313 2.625-2.625 2.625H7.125c-1.313
0-2.625-1.313-2.625-2.625S5.812 11.625 7.125 11.625h9.75c1.313 0
2.625 1.313 2.625 2.625z" />

<path
stroke-linecap="round" stroke-linejoin="round" d="M12.75
17.25v-2.25M10.5 17.25v-2.25M15 17.25v-2.25M12 7.5a2.25 2.25 0
012.25 2.25v1.5H9.75v-1.5A2.25 2.25 0 0112 7.5zM15 3.75l-1.5
1.5M9 3.75l1.5 1.5M19.5 7.5l-1.5 1.5M4.5 7.5l1.5 1.5" />

</svg>

```



```

                <span class="mt-2 font-medium
text-sm" data-il8n="action_pesticide">Pesticide</span>
            </button>
            <button id="action-voice"
class="flex flex-col items-center justify-center p-4 bg-gray-700
text-white rounded-lg shadow-md hover:bg-gray-800
focus:outline-none focus:ring-2 focus:ring-offset-2
focus:ring-gray-700 transition-all transform hover:scale-105">
                <svg class="h-10 w-10"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24
24" stroke-width="1.5" stroke="currentColor">
                    <path
stroke-linecap="round" stroke-linejoin="round" d="M12
18.75a.75.75 0 00.75-.75V6.75a.75.75 0 00-1.5 0v11.25a.75.75 0
00.75.75zM12 21a.75.75 0 00.75-.75v-.01a.75.75 0 00-1.5
0v.01A.75.75 0 0012 21z" />
                    <path
stroke-linecap="round" stroke-linejoin="round" d="M12 15a4.5 4.5
0 004.5-4.5V6a4.5 4.5 0 10-9 0v4.5A4.5 4.5 0 0012 15zM6.75
10.5v.75a5.25 5.25 0 0010.5 0v-.75" />
                </svg>
                <span class="mt-2 font-medium
text-sm" data-il8n="action_voice">Ask</span>
            </button>
        </div>
    </div>

    <div class="lg:col-span-1 space-y-6">
        <div class="bg-white shadow-lg
rounded-lg">

```

```

<div class="p-5 border-b
border-gray-200">
    <div class="flex items-center">
        <svg class="h-6 w-6
text-red-600" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">
            <path
stroke-linecap="round" stroke-linejoin="round" d="M12
9v3.75m-9.303 3.376c-.866 1.5.217 3.374 1.948 3.374h14.71c1.73 0
2.813-1.874 1.948-3.374L13.949 3.378c-.866-1.5-3.032-1.5-3.898
0L2.697 16.126z" />
        </svg>
        <h3 class="ml-3 text-lg
font-medium leading-6 text-gray-900"
data-i18n="rec_title">Priority Actions</h3>
    </div>
    <p class="mt-1 text-sm
text-gray-500 ml-9" data-i18n="rec_subtitle">Smart advice based
on your field data.</p>
</div>
<div class="p-5">
    <ul id="recommendations-list"
class="space-y-3">
        <li class="p-3 bg-gray-50
rounded-lg text-gray-500 italic"
data-i18n="rec_loading">Analyzing data...</li>
    </ul>
</div>
</div>

<div class="bg-white shadow-lg
rounded-lg">

```

```
<div class="p-5 border-b
border-gray-200">
    <div class="flex items-center">
        <svg class="h-6 w-6
text-amber-500" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">
            <path
stroke-linecap="round" stroke-linejoin="round" d="M9.813
15.904L9 18.751-.813-2.846a4.5 4.5 0 00-3.09-3.09L2.25
1212.846-.813a4.5 4.5 0 003.09-3.09L9 5.251.813 2.846a4.5 4.5 0
003.09 3.09L15.75 121-2.846.813a4.5 4.5 0 00-3.09 3.09zM18.25
12L17 14.1881-1.25.467L17 14.654 18.25
1711.25-2.346.467-1.25-.467-1.25L18.25 101-1.25 2.346L17
12.4671-1.25-.467L17 11.533 18.25 911.25 2.346L21.75
121-.467.467L21.75 12z" />
        </svg>
        <h3 class="ml-3 text-lg
font-medium leading-6 text-gray-900" data-il8n="gemini_title">✨
Gemini AI Report</h3>
    </div>
    <p class="mt-1 text-sm
text-gray-500 ml-9" data-il8n="gemini_subtitle">Long-term
strategic advice.</p>
</div>
<div class="p-5 space-y-4">
    <div id="gemini-report-content"
class="text-sm text-gray-700 space-y-4 bg-amber-50/50 p-4
rounded-lg">
        <div>
            <h4 class="font-semibold
text-gray-800" data-il8n="gemini_yield">Yield Outlook</h4>
```

```

        <p
id="gemini-yield-outlook" class="mt-1"
data-118n="gemini_yield_placeholder">Click the button to
generate an AI-powered yield forecast.</p>
    </div>
    <div>
        <h4 class="font-semibold
text-gray-800" data-118n="gemini_advice">Strategic Advice</h4>
        <p
id="gemini-strategic-advice" class="mt-1"
data-118n="gemini_advice_placeholder">Click the button to get
long-term strategic advice.</p>
    </div>
    <p id="gemini-error-message"
class="text-red-600 text-xs hidden"
data-118n="gemini_error">Error generating report. Please try
again.</p>
    </div>
    <button
id="gemini-generate-report" class="w-full flex items-center
justify-center px-4 py-2 border border-transparent rounded-lg
shadow-sm text-sm font-medium text-white bg-gradient-to-r
from-amber-600 to-amber-700 hover:from-amber-700
hover:to-amber-800 focus:outline-none focus:ring-2
focus:ring-offset-2 focus:ring-amber-500 transition-all
disabled:opacity-50 transform hover:scale-105">
        <span id="gemini-button-text"
data-118n="gemini_generate">🌟 Generate AI Report</span>
    </div>
id="gemini-loading-spinner" class="spinner ml-2 hidden"></div>
    </button>
</div>

```

```

        </div>

        <div class="bg-white shadow-lg
rounded-lg">

            <div class="p-5 border-b
border-gray-200">

                <div class="flex items-center">

                    <svg class="h-6 w-6
text-amber-600" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">
                        <path
stroke-linecap="round" stroke-linejoin="round" d="M8.25
6.75h12M8.25 12h12M8.25 17.25h12M3.75
6.75h.007v.008H3.75V6.75zm.375 0a.375.375 0 11-.75 0 .375.375 0
01.75 0zM3.75 12h.007v.008H3.75V12zm.375 0a.375.375 0 11-.75 0
.375.375 0 01.75 0zm-.375 5.25h.007v.008H3.75v-.008zm.375
0a.375.375 0 11-.75 0 .375.375 0 01.75 0z" />
                    </svg>

                    <h3 class="ml-3 text-lg
font-medium leading-6 text-gray-900"
data-il8n="growth_title">Crop Growth Stage</h3>

                </div>

                <p class="mt-1 text-sm
text-gray-500 ml-9" data-il8n="growth_subtitle">Tracking
progress to maturity.</p>

            </div>

            <div class="p-5 space-y-3">

                <div class="flex justify-between
text-sm font-medium text-gray-700">

                    <span
data-il8n="growth_stage"
id="growth-stage-label">Germination</span>

```

```

                <span id="growth-stage-days"
class="font-bold">Day 1 / 56</span>
            </div>
            <div class="w-full bg-gray-200
rounded-full h-4">
                <div id="growth-stage-bar"
class="bg-amber-600 h-4 rounded-full transition-all"
style="width: 2%"></div>
            </div>
            <div class="text-xs text-gray-500
flex justify-between">
                <span
data-i18n="stage_1">Germination</span>
                <span
data-i18n="stage_2">Vegetative</span>
                <span
data-i18n="stage_3">Flowering</span>
                <span
data-i18n="stage_4">Mature</span>
            </div>
        </div>
    </div>

    <div class="bg-white shadow-lg
rounded-lg">
        <div class="p-5 border-b
border-gray-200">
            <div class="flex items-center">
                <svg class="h-6 w-6
text-blue-600" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor">

```

```

<path
stroke-linecap="round" stroke-linejoin="round" d="M20.25 6.375c0
2.278-3.694 4.125-8.25 4.125S3.75 8.653 3.75 6.375m16.5
0c0-2.278-3.694-4.125-8.25-4.125S3.75 4.097 3.75 6.375m16.5
0v11.25c0 2.278-3.694 4.125-8.25
4.125s-8.25-1.847-8.25-4.125V6.375" />
</svg>
<h3 class="ml-3 text-lg
font-medium leading-6 text-gray-900" data-il8n="soil_title">Soil
Nutrient Overview</h3>
</div>
<p class="mt-1 text-sm
text-gray-500 ml-9" data-il8n="soil_subtitle">Current nutrient
levels.</p>
</div>
<div class="p-5 space-y-4">
  <div>
    <div class="flex
justify-between text-sm font-medium text-gray-700">
      <span
data-il8n="soil_n">Nitrogen (N)</span>
      <span
id="soil-nitrogen-value" class="font-bold">60%</span>
    </div>
    <div class="mt-1 w-full
bg-gray-200 rounded-full h-2.5">
      <div
id="soil-nitrogen-bar" class="bg-green-600 h-2.5 rounded-full"
style="width: 60%"></div>
    </div>
  </div>
</div>
<div>

```

```

                <div class="flex
justify-between text-sm font-medium text-gray-700">
                    <span
data-i18n="soil_p">Phosphorus (P)</span>
                    <span
id="soil-phosphorus-value" class="font-bold">50%</span>
                </div>
                <div class="mt-1 w-full
bg-gray-200 rounded-full h-2.5">
                    <div
id="soil-phosphorus-bar" class="bg-purple-600 h-2.5
rounded-full" style="width: 50%"></div>
                </div>
            </div>
            <div>
                <div class="flex
justify-between text-sm font-medium text-gray-700">
                    <span
data-i18n="soil_k">Potassium (K)</span>
                    <span
id="soil-potassium-value" class="font-bold">55%</span>
                </div>
                <div class="mt-1 w-full
bg-gray-200 rounded-full h-2.5">
                    <div
id="soil-potassium-bar" class="bg-yellow-500 h-2.5 rounded-full"
style="width: 55%"></div>
                </div>
            </div>
        </div>
    </div>
</div>

```



```

        </div>
    </div>
</main>
</div>

<!-- Toast (notifications) -->
<div id="toast-message" class="fixed bottom-5 right-5 flex
items-center bg-gray-900 text-white py-3 px-5 rounded-lg
shadow-xl opacity-0 transition-opacity duration-300
pointer-events-none">
    <svg id="toast-icon-success" class="h-5 w-5
text-green-500 mr-2" xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" stroke-width="1.5"
stroke="currentColor">
        <path stroke-linecap="round" stroke-linejoin="round"
d="M9 12.75L11.25 15 15 9.75M21 12a9 9 0 11-18 0 9 9 0 0118 0z"
/>
    </svg>
    <svg id="toast-icon-info" class="h-5 w-5 text-blue-500
mr-2" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0
0 24 24" stroke-width="1.5" stroke="currentColor">
        <path stroke-linecap="round" stroke-linejoin="round"
d="M11.25 11.25l.041-.02a.75.75 0 011.063.852l-.708 2.836a.75.75
0 001.063.853l.041-.021M21 12a9 9 0 11-18 0 9 9 0 0118 0z" />
    </svg>
    <svg id="toast-icon-error" class="h-5 w-5 text-red-500
mr-2" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0
0 24 24" stroke-width="1.5" stroke="currentColor">
        <path stroke-linecap="round" stroke-linejoin="round"
d="M12 9v3.75m-9.303 3.376c-.866 1.5.217 3.374 1.948
3.374h14.71c1.73 0 2.813-1.874 1.948-3.374L13.949
3.378c-.866-1.5-3.032-1.5-3.898 0L2.697 16.126z" />

```

```

    </svg>

    <span id="toast-text">Action successful!</span>
  </div>

  <!-- Tooltip (for per-cell satellite info) -->
  <div id="cell-tooltip" class="hidden fixed z-50
pointer-events-none bg-black text-white text-xs px-2 py-1
rounded-md opacity-90"><span
id="cell-tooltip-content"></span></div>

<script>
  document.addEventListener('DOMContentLoaded', () => {

    // --- i18n and constants ---
    const i18n = {
      en: {
        app_title: 'AgriFuture',
        lang_toggle: 'हिन्दी',
        offline_banner: 'You are in Offline Mode.
Showing last saved data.',
        offline_toggle: 'Offline Mode',
        dashboard_title: "Farmer's Dashboard",
        dashboard_subtitle: 'Clear, actionable
insights to protect your yield.',
        map_title: 'Field Health Heatmap (NDVI)',
        map_subtitle: 'Green is healthy,
Yellow/Orange is stressed.',
        stressed: 'Stressed',
        healthy: 'Healthy',
        actions_title: 'Field Management',
        actions_subtitle: 'Apply interventions or use
voice commands.',

```

```
        action_irrigate: 'Irrigate',
        action_fertilize: 'Fertilize',
        action_pesticide: 'Pesticide',
        action_voice: 'Ask',
        rec_title: 'Priority Actions',
        rec_subtitle: 'Smart advice based on your
field data.',

        rec_loading: 'Analyzing data...',
        gemini_title: '✨ Gemini AI Report',
        gemini_subtitle: 'Long-term strategic
advice.',

        gemini_yield: 'Yield Outlook',
        gemini_yield_placeholder: 'Click the button
to generate an AI-powered yield forecast.',
        gemini_advice: 'Strategic Advice',
        gemini_advice_placeholder: 'Click the button
to get long-term strategic advice.',
        gemini_error: 'Error generating report.
Please try again.',

        gemini_generate: '✨ Generate AI Report',
        growth_title: 'Crop Growth Stage',
        growth_subtitle: 'Tracking progress to
maturity.',

        growth_stage: 'Germination',
        stage_1: 'Germination',
        stage_2: 'Vegetative',
        stage_3: 'Flowering',
        stage_4: 'Mature',
        soil_title: 'Soil Nutrient Overview',
        soil_subtitle: 'Current nutrient levels.',
        soil_n: 'Nitrogen (N)',
        soil_p: 'Phosphorus (P)',
```

```
soil_k: 'Potassium (K) '
},
hi: {
  app_title: 'एग्रीफ्यूचर',
  lang_toggle: 'English',
  offline_banner: 'आप ऑफलाइन मोड में हैं। अंतिम सेव्ड
  डेटा दिखाया जा रहा है।',
  offline_toggle: 'ऑफलाइन मोड',
  dashboard_title: 'किसान डैशबोर्ड',
  dashboard_subtitle: 'आपकी पैदावार की रक्षा के लिए
  स्पष्ट और करेंट सुझाव।',
  map_title: 'फील्ड हेल्थ हीटमैप (NDVI)',
  map_subtitle: 'हरा स्वस्थ है, पीला/नारंगी तनाव दिखाती
  है।',
  stressed: 'तनावग्रस्त',
  healthy: 'स्वस्थ',
  actions_title: 'फील्ड प्रबंधन',
  actions_subtitle: 'हस्तक्षेप लागू करें या वॉइस कमांड का
  उपयोग करें।',
  action_irrigate: 'सिंचाई करें',
  action_fertilize: 'उर्वरक डालें',
  action_pesticide: 'कीट नाशक',
  action_voice: 'पूछें',
  rec_title: 'प्राथमिक कार्य',
  rec_subtitle: 'आपके फील्ड डेटा के आधार पर सुझाव।',
  rec_loading: 'डेटा विश्लेषण हो रहा है...',
  gemini_title: '🌟 जेमिनी AI रिपोर्ट',
  gemini_subtitle: 'दीर्घकालिक रणनीतिक सलाह।',
  gemini_yield: 'उपज का आउटलुक',
  gemini_yield_placeholder: 'AI-पावर्ड उपज पूर्वानुमान
  के लिए बटन दबाएँ।',
  gemini_advice: 'रणनीतिक सलाह',
```

```

        gemini_advice_placeholder: 'दीर्घकालिक रणनीतिक
सलाह के लिए बटन दबाएँ।',
        gemini_error: 'रिपोर्ट जनरेट करने में त्रुटि। कृपया पुनः
प्रयास करें।',

        gemini_generate: '🌟 AI रिपोर्ट बनाएं',
        growth_title: 'फसल विकास चरण',
        growth_subtitle: 'परिपक्वता की प्रगति पर नज़र।',
        growth_stage: 'अंकुरण',
        stage_1: 'अंकुरण',
        stage_2: 'वेजिटेटिव',
        stage_3: 'फ्लॉवरिंग',
        stage_4: 'परिपक्व',
        soil_title: 'मृदा पोषक तत्व ओवरव्यू',
        soil_subtitle: 'वर्तमान पोषक तत्व स्तर।',
        soil_n: 'नाइट्रोजन (N)',
        soil_p: 'फॉस्फोरस (P)',
        soil_k: 'पोटैशियम (K)'
    }
};

function applyLanguage() {
    // replace text for every element with data-i18n

document.querySelectorAll('[data-i18n]').forEach(el => {
        const key = el.getAttribute('data-i18n');
        const txt = (i18n[currentLanguage] &&
i18n[currentLanguage][key]) || (i18n['en'] && i18n['en'][key])
|| '';

        // if element is an input/button, set value
or textContent accordingly
        if (el.tagName.toLowerCase() === 'input' ||
el.tagName.toLowerCase() === 'textarea') {

```

```

        el.value = txt;
    } else {
        el.textContent = txt;
    }
});
// update lang-toggle button label specifically
const langBtn =
document.getElementById('lang-toggle');
if (langBtn) {
    const lbl = (i18n[currentLanguage] &&
i18n[currentLanguage].lang_toggle) || 'Lang';
    langBtn.textContent = lbl;
}
}

```

```

// ensure init applies language after loading cache
or resetting state

```

```

const GRID_SIZE = 10;
const TICK_RATE = 5000;
const CROP_CYCLE_LENGTH = 56;
let farmGrid = [];
let weather = {};
let simulationEvents = {};
let dayOfCycle = 1;
let ndviHistory = [];
let soilMoistureHistory = [];
let currentLanguage = 'en';
let isOffline = false;
let gameLoopInterval;

```

```
        // DOM handles - ensure these are fetched inside
DOMContentLoaded

        const gridContainer =
document.getElementById('farm-grid');

        const recList =
document.getElementById('recommendations-list');

        const toast =
document.getElementById('toast-message');

        const toastText =
document.getElementById('toast-text');

        const toastIconSuccess =
document.getElementById('toast-icon-success');

        const toastIconInfo =
document.getElementById('toast-icon-info');

        const toastIconError =
document.getElementById('toast-icon-error');

        const navAvgNdvi =
document.getElementById('nav-avg-ndvi');

        const navAvgMoist =
document.getElementById('nav-avg-moist');

        const navTemp =
document.getElementById('nav-current-temp');

        const geminiErrorEl =
document.getElementById('gemini-error-message');

        const tooltip =
document.getElementById('cell-tooltip');

        const tooltipContent =
document.getElementById('cell-tooltip-content');


        // --- Utility: safe set text ---
        function safeSetText(el, text) {
            if (!el) return;
```

```

        el.textContent = text;
    }

    // --- Tooltip helpers ---
    function showCellTooltip(x, y, htmlContent, clientX,
clientY) {
        if (!tooltip || !tooltipContent) return;
        tooltipContent.innerHTML = htmlContent;
        tooltip.classList.remove('hidden');
        const padding = 12;
        const vw =
Math.max(document.documentElement.clientWidth || 0,
window.innerWidth || 0);
        const vh =
Math.max(document.documentElement.clientHeight || 0,
window.innerHeight || 0);
        let left = clientX + padding;
        let top = clientY + padding;
        const rect = tooltip.getBoundingClientRect();
        if (left + rect.width + 8 > vw) left = clientX -
rect.width - padding;
        if (top + rect.height + 8 > vh) top = clientY -
rect.height - padding;
        tooltip.style.left = left + 'px';
        tooltip.style.top = top + 'px';
    }

    function hideCellTooltip() {
        if (!tooltip) return;
        tooltip.classList.add('hidden');
    }

```



```

// --- Initialization and simulation ---
function init() {
    if (loadCache()) {
        showToast('Loaded cached state', 'info');
    } else {
        resetState();
    }
    // apply language strings to the UI before
rendering

    applyLanguage();
    renderAll();
    addEventListeners();
    startGameLoop();
    initVoice();
}

function resetState() {
    dayOfCycle = 1;
    farmGrid = [];
    for (let y = 0; y < GRID_SIZE; y++) {
        const row = [];
        for (let x = 0; x < GRID_SIZE; x++) {
            row.push({
                x, y,
                soilMoisture: 0.7 + (Math.random() -
0.5) * 0.2,
                nitrogen: 0.6 + (Math.random() - 0.5)
* 0.2,
                phosphorus: 0.5 + (Math.random() -
0.5) * 0.2,
                potassium: 0.55 + (Math.random() -
0.5) * 0.2,
            });
        }
    }
}

```

```

        ndvi: 0
    });
    }
    farmGrid.push(row);
}
weather = { current: { type: 'Sunny', temp: 28,
humidity: 0.6, rainProb: 0.1 }, forecast: generateForecast() };
simulationEvents = { pest: false, pestLocation: {
x: -1, y: -1 } };
ndviHistory = [];
soilMoistureHistory = [];
calculateAllNDVI();
}

function addEventListeners() {
    const elIrr =
document.getElementById('action-irrigate');
    const elFert =
document.getElementById('action-fertilize');
    const elPest =
document.getElementById('action-pesticide');
    const elVoice =
document.getElementById('action-voice');
    const elLang =
document.getElementById('lang-toggle');
    const elOffline =
document.getElementById('offline-toggle');
    const elGemini =
document.getElementById('gemini-generate-report');

    if (elIrr) elIrr.addEventListener('click',
handleIrrigate);

```

```
        if (elFert) elFert.addEventListener('click',
handleFertilize);

        if (elPest) elPest.addEventListener('click',
handlePesticide);

        if (elVoice) elVoice.addEventListener('click',
startVoiceCommand);

        if (elLang) elLang.addEventListener('click',
toggleLanguage);

        if (elOffline)
elOffline.addEventListener('change', toggleOffline);

        if (elGemini) elGemini.addEventListener('click',
handleGeminiReport);


        window.addEventListener('scroll',
hideCellTooltip, true);

        window.addEventListener('resize',
hideCellTooltip);

        document.addEventListener('click', (e) => {
            const isCell = e.target.closest '&&
e.target.closest('.grid-cell');
            if (!isCell) hideCellTooltip();
        });
    }

    function startGameLoop() {
        if (gameLoopInterval)
clearInterval(gameLoopInterval);
        gameLoopInterval = setInterval(gameTick,
TICK_RATE);
    }

    function gameTick() {
```

```

        if (isOffline) return;
        updateWeather();
        updateFarmData();
        spawnEvents();
        calculateAllNDVI();
        const avgs = getAverages();
        ndviHistory.push(avgs.ndvi);
        if (ndviHistory.length > 5) ndviHistory.shift();
        soilMoistureHistory.push(avgs.moisture);
        if (soilMoistureHistory.length > 20)
soilMoistureHistory.shift();
        dayOfCycle = (dayOfCycle % CROP_CYCLE_LENGTH) +
1;

        renderAll();
        saveCache();
    }

    function saveCache() {
        try {
            const cache = { farmGrid, weather,
simulationEvents, dayOfCycle, ndviHistory, soilMoistureHistory,
currentLanguage };
            localStorage.setItem('agriFutureCache',
JSON.stringify(cache));
        } catch (e) { console.warn('Could not save
cache', e); }
    }

    // ----- helpers & simple implementations so
preview works -----

    function generateForecast() {

```

```

        return Array.from({length: 5}).map((_, i) => ({day:
i+1, type:
['Sunny', 'Cloudy', 'Rain'][Math.floor(Math.random()*3)], temp:
25+Math.floor(Math.random()*6)}));
    }

    function calculateAllNDVI() {
        for (let y=0; y<farmGrid.length; y++){
            for (let x=0; x<farmGrid[y].length; x++){
                const cell = farmGrid[y][x];
                // simple NDVI proxy: based on moisture
and nitrogen
                const raw = Math.max(0, Math.min(1,
(cell.soilMoisture*0.6 + cell.nitrogen*0.4) ));
                cell.ndvi = Math.round(raw * 10);
            }
        }
    }

    function updateWeather(){
        // small random drift
        if (!weather.current) weather.current =
{type:'Sunny', temp:28, humidity:0.6, rainProb:0.1};
        weather.current.temp += (Math.random()-0.5)*0.5;
        if (Math.random() < 0.05) weather.current.type =
['Sunny', 'Cloudy', 'Rain'][Math.floor(Math.random()*3)];
    }

    function updateFarmData(){
        // small changes: evaporate moisture
        for (let row of farmGrid){
            for (let cell of row){

```

```

        cell.soilMoisture = Math.max(0.05,
cell.soilMoisture - 0.01*(0.5+Math.random()));
    }
}

function spawnEvents(){
    // occasional pest event
    if (Math.random() < 0.02){
        simulationEvents.pest = true;
        simulationEvents.pestLocation = { x:
Math.floor(Math.random()*GRID_SIZE), y:
Math.floor(Math.random()*GRID_SIZE) };
    } else {
        simulationEvents.pest = false;
    }
}

function getAverages(){
    let total=0, moist=0, count=0;
    for (let r of farmGrid){ for (let c of r){
total+=c.ndvi; moist+=c.soilMoisture; count++; }}
    return { ndvi: count? (total/count):0, moisture:
count? (moist/count):0 };
}

function renderAll(){
    renderGrid();
    const avgs = getAverages();
    safeSetText(navAvgNdvi, avgs.ndvi.toFixed(1));
    safeSetText(navAvgMoist,
(avgs.moisture*100).toFixed(0)+'%');

```

```

        safeSetText(navTemp, weather.current ?
Math.round(weather.current.temp)+'°C' : '--');
        renderRecommendations();
        renderGrowth();
    }

    function renderGrid(){
        if (!gridContainer) return;
        gridContainer.innerHTML = '';
        for (let y=0;y<GRID_SIZE;y++){
            for (let x=0;x<GRID_SIZE;x++){
                const cell = farmGrid[y][x];
                const el = document.createElement('div');
                el.className = 'grid-cell
cb-ndvi-'+Math.max(0,Math.min(10,cell.ndvi));
                el.dataset.x = x; el.dataset.y = y;
                el.addEventListener('mouseenter', (ev)=>{
                    showCellTooltip(x,y,`NDVI:
${cell.ndvi}<br>Moist: ${ (cell.soilMoisture*100).toFixed(0)}%`,
ev.clientX, ev.clientY);
                });
                el.addEventListener('mousemove', (ev)=>{
                    showCellTooltip(x,y,`NDVI:
${cell.ndvi}<br>Moist: ${ (cell.soilMoisture*100).toFixed(0)}%`,
ev.clientX, ev.clientY);
                });
                el.addEventListener('mouseleave',
hideCellTooltip);
                gridContainer.appendChild(el);
            }
        }
    }
}

```

```

function renderRecommendations(){
    if (!recList) return;
    recList.innerHTML = '';
    // simple rules
    const low = [];
    for (let r of farmGrid){ for (let c of r){ if
(c.ndvi <= 3) low.push(c); }}
    if (low.length){
        const li = document.createElement('li');
        li.className = 'p-3 bg-white rounded-lg
shadow-sm';
        li.textContent = `Low NDVI in ${low.length}
patches – consider targeted irrigation or nutrient
application.`;
        recList.appendChild(li);
    } else {
        const li = document.createElement('li');
        li.className = 'p-3 bg-gray-50 rounded-lg
text-gray-500 italic';
        li.textContent = 'No urgent actions
detected.';
        recList.appendChild(li);
    }
}

function renderGrowth(){
    const pct =
Math.round((dayOfCycle/CROP_CYCLE_LENGTH)*100);
    const bar =
document.getElementById('growth-stage-bar');

```



```

        const lbl =
document.getElementById('growth-stage-days');
        if (bar) bar.style.width = pct+'%';
        if (lbl) lbl.textContent = `Day ${dayOfCycle} /
${CROP_CYCLE_LENGTH}`;
    }

    // simple voice init placeholder
    function initVoice(){
        // placeholder: later you can integrate Web
Speech API
    }

    // action handlers (simple)
    function handleIrrigate(){
        for (let r of farmGrid){ for (let c of r){
c.soilMoisture = Math.min(1, c.soilMoisture + 0.12); }}
        calculateAllNDVI(); renderAll(); saveCache();
showToast('Irrigation applied', 'success');
    }

    function handleFertilize(){
        for (let r of farmGrid){ for (let c of r){
c.nitrogen = Math.min(1, c.nitrogen + 0.1); }}
        calculateAllNDVI(); renderAll(); saveCache();
showToast('Fertilizer applied', 'success');
    }

    function handlePesticide(){
        simulationEvents.pest = false; renderAll();
saveCache(); showToast('Pesticide applied', 'success');
    }

    function startVoiceCommand(){ showToast('Voice
command not enabled in preview', 'info'); }

```

```

        function toggleLanguage(){
            currentLanguage = currentLanguage==='en' ? 'hi' :
'en';

            applyLanguage();
            saveCache();
            // show a small toast in the chosen language
            const label = currentLanguage === 'en' ?
'Language: English' : 'भाषा अब हिन्दी चुनी गई';
            showToast(label, 'info');
        }

        function toggleOffline(e){ isOffline =
e.target.checked;
document.getElementById('offline-banner').style.display =
isOffline ? 'block' : 'none'; showToast(isOffline? 'Offline mode
enabled':'Offline mode disabled', 'info'); }

        function handleGeminiReport(){ const btn =
document.getElementById('gemini-generate-report'); btn.disabled
= true;
document.getElementById('gemini-loading-spinner').classList.remo
ve('hidden'); setTimeout(()=>{
document.getElementById('gemini-loading-spinner').classList.add(
'hidden'); btn.disabled = false;
document.getElementById('gemini-yield-outlook').textContent =
'Yield looks stable – continue current regime.';
document.getElementById('gemini-strategic-advice').textContent =
'Rotate crops next season and monitor soil N.'; showToast('AI
report generated','success'); }, 1200); }

        // ----- cache load/save completion, error
handling, init -----

        function loadCache() {
            try {

```

```

        const cacheString =
localStorage.getItem('agriFutureCache');
        if (!cacheString) return false;
        const cache = JSON.parse(cacheString);
        if (cache && Array.isArray(cache.farmGrid) &&
cache.dayOfCycle) {
            farmGrid = cache.farmGrid;
            weather = cache.weather || weather;
            simulationEvents = cache.simulationEvents
|| simulationEvents;
            dayOfCycle = cache.dayOfCycle ||
dayOfCycle;
            ndviHistory = cache.ndviHistory || [];
            soilMoistureHistory =
cache.soilMoistureHistory || [];
            currentLanguage = cache.currentLanguage
|| currentLanguage;
            if (farmGrid.length !== GRID_SIZE) {
                console.warn('Cached grid size
mismatch - resetting state.');
```

```

                resetState();
                return false;
            }
            return true;
        }
    } catch (e) {
        console.warn('Failed to load cache', e);
    }
    return false;
}

```

```
function showToast(message, type='info',
duration=3000) {
    if (!toast) return console.log('[toast]',
message);

    toast.classList.remove('opacity-0');
    toast.style.pointerEvents = 'auto';
    toastText.textContent = message;
    toastIconSuccess.style.display = (type ===
'success') ? 'inline-block' : 'none';
    toastIconInfo.style.display = (type === 'info') ?
'inline-block' : 'none';
    toastIconError.style.display = (type === 'error')
? 'inline-block' : 'none';
    setTimeout(() => {
        toast.classList.add('opacity-0');
        toast.style.pointerEvents = 'none';
    }, duration);
}

window.addEventListener('error', (event) => {
    console.error('Unhandled error:', event.error ||
event.message);
    showToast('An error occurred: ' + (event.message
|| 'See console'), 'error', 6000);
});

window.addEventListener('unhandledrejection', (ev) =>
{
    console.error('Unhandled promise rejection:',
ev.reason);
```

```
        showToast('Promise rejection: ' + (ev.reason &&
ev.reason.message ? ev.reason.message : String(ev.reason)),
'error', 6000);
    });

    try {
        if (document.readyState === 'complete' ||
document.readyState === 'interactive') {
            init();
        } else {
            document.addEventListener('DOMContentLoaded',
init);
        }
    } catch (e) {
        console.error('Init failed', e);
        showToast('Initialization failed – check
console', 'error', 8000);
    }

});
</script>
</body>
</html>
```