Quickstart Guide: Launch Global Wellness Perspectives in 24 Hours

This guide provides the exact steps to get your application online within 24 hours. Follow these instructions in order.

Hour 1: Project Setup

Create the Next.js Project

bash

Copy

```bash
# Create a new Next.js project with TypeScript and Tailwind
npx create-next-app@latest global-wellness-perspectives --typescript --tailwind --esli
cd global-wellness-perspectives

# Install required dependencies
npm install recharts axios
```

Set Up Version Control

bash

Copy

```bash
git init
git add .
git commit -m "Initial commit with project setup"
```

Create GitHub Repository

1. Go to GitHub and create a new repository named "global-wellness-perspectives"
2. Follow the instructions to push your existing repository:

bash

Copy

```bash
git remote add origin https://github.com/YOUR-USERNAME/global-wellness-perspectives.gi
git branch -M main
git push -u origin main
```

Hour 2-3: Core File Structure and Static Data

Create the API Directory and Data File

Create the file `pages/api/wellness-data.js` with the sample data:

javascript

Copy

```javascript
export default function handler(req, res) {
  const { countries, metrics } = req.query;

  const selectedCountries = countries ? countries.split(',') : ["USA", "Finland", "Japa
  const selectedMetrics = metrics ? metrics.split(',') : ["happiness", "healthcare", "

  // Country profiles with base values and adjustments
  const countryProfiles = {
    'USA': { base: 70, healthcare: -10, education: 0, safety: -15, work_life: -20, inc
    'Finland': { base: 90, happiness: +5, healthcare: +8, education: +5, work_life: +1
    'Sweden': { base: 88, happiness: +4, healthcare: +7, education: +6, work_life: +8,
    'Norway': { base: 89, happiness: +5, healthcare: +8, education: +5, work_life: +9,
    'Denmark': { base: 87, happiness: +8, healthcare: +6, education: +4, work_life: +1
    'Germany': { base: 82, healthcare: +5, education: +3, safety: +2, income_inequalit
    'Japan': { base: 83, healthcare: +6, education: +5, safety: +10, work_life: -15, s
    'Costa Rica': { base: 77, happiness: +10, environmental: +12, social_support: +6,
    'Canada': { base: 82, healthcare: +5, education: +3, social_support: +4, safety: +
  };

  // Calculate the metric values for each country
  const result = {
    countries: selectedCountries,
    metrics: {}
  };

  // Generate data for each requested metric
  selectedMetrics.forEach(metricId => {
    result.metrics[metricId] = {};

    selectedCountries.forEach(country => {
      if (!countryProfiles[country]) {
        // If country not found, use average values
        result.metrics[metricId][country] = 65 + Math.random() * 10;
        return;
      }

      const profile = countryProfiles[country];

      // Base value + specific adjustment for this metric + small random variation
      let value = profile.base;
      // Add metric-specific adjustment if available
      if (profile[metricId]) {
        value += profile[metricId];
      }

      // Add a small random variation (-3 to +3)
      value += (Math.random() * 6) - 3;

      // Ensure value is within 0-100 range
      value = Math.max(0, Math.min(100, value));

      result.metrics[metricId][country] = parseFloat(value.toFixed(1));
    });
  });

  res.status(200).json(result);
}
```

Create Components Directory

bash

Copy

```bash
mkdir -p components
```

Hour 4-6: Implement Core Components

1. Create Main Page (pages/index.js)

Use the code from the global-wellness-comparisons artifact, but initially without the optional components.

2. Create WellnessComparison Component

Copy the code from the component-wellness-comparison artifact to `components/WellnessComparison.js`.

3. Create Selector Components

Copy the code from the component-selectors artifact to:

- `components/CountrySelector.js`
- `components/MetricSelector.js`

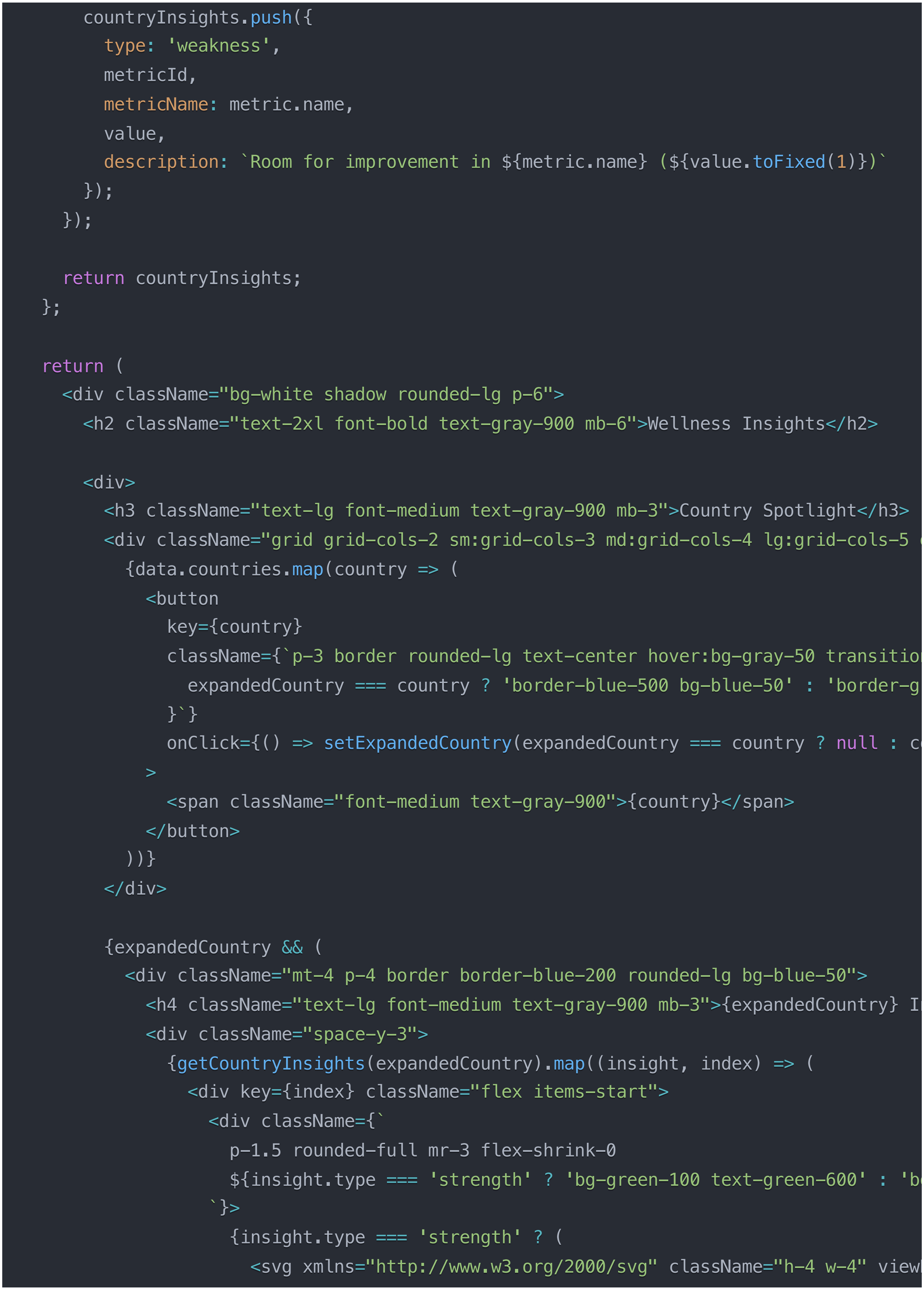
4. Create WellnessInsights Component

Copy the simplified version below to `components/WellnessInsights.js`:

javascript

Copy

```javascript
import { useState } from 'react';

export default function WellnessInsights({ data, metrics }) {
  const [expandedCountry, setExpandedCountry] = useState(null);

  // Get country-specific insights
  const getCountryInsights = (country) => {
    const countryInsights = [];

    // Get all metrics for this country
    const countryData = {};
    Object.keys(data.metrics).forEach(metricId => {
      countryData[metricId] = data.metrics[metricId][country];
    });

    // Find this country's strengths and weaknesses
    const metricEntries = Object.entries(countryData);
    const sortedMetrics = [...metricEntries].sort((a, b) => b[1] - a[1]);

    const strengths = sortedMetrics.slice(0, 2);
    const weaknesses = sortedMetrics.slice(-2).reverse();

    // Format strengths
    strengths.forEach(([metricId, value]) => {
      const metric = metrics.find(m => m.id === metricId);
      if (!metric) return;

      countryInsights.push({
        type: 'strength',
        metricId,
        metricName: metric.name,
        value,
        description: `Strong performance in ${metric.name} (${value.toFixed(1)})`
      });
    });

    // Format weaknesses
    weaknesses.forEach(([metricId, value]) => {
      const metric = metrics.find(m => m.id === metricId);
      if (!metric) return;
```



```
      countryInsights.push({
        type: 'weakness',
        metricId,
        metricName: metric.name,
        value,
        description: `Room for improvement in ${metric.name} (${value.toFixed(1)})`
      });
    });

    return countryInsights;
  };

  return (
    <div className="bg-white shadow rounded-lg p-6">
      <h2 className="text-2xl font-bold text-gray-900 mb-6">Wellness Insights</h2>

      <div>
        <h3 className="text-lg font-medium text-gray-900 mb-3">Country Spotlight</h3>
        <div className="grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5
          {data.countries.map(country => (
            <button
              key={country}
              className={`p-3 border rounded-lg text-center hover:bg-gray-50 transitio
                expandedCountry === country ? 'border-blue-500 bg-blue-50' : 'border-g
              }`}
              onClick={() => setExpandedCountry(expandedCountry === country ? null : c
            >
              <span className="font-medium text-gray-900">{country}</span>
            </button>
          ))}
        </div>

        {expandedCountry && (
          <div className="mt-4 p-4 border border-blue-200 rounded-lg bg-blue-50">
            <h4 className="text-lg font-medium text-gray-900 mb-3">{expandedCountry} I
            <div className="space-y-3">
              {getCountryInsights(expandedCountry).map((insight, index) => (
                <div key={index} className="flex items-start">
                  <div className={`
                    p-1.5 rounded-full mr-3 flex-shrink-0
                    ${insight.type === 'strength' ? 'bg-green-100 text-green-600' : 'b
                  `}>
                    {insight.type === 'strength' ? (
                      <svg xmlns="http://www.w3.org/2000/svg" className="h-4 w-4" view
```

```
                        <path fillRule="evenodd" d="M12 7a1 1 0 110-2h5a1 1 0 011 1v5a
                      </svg>
                    ) : (
                      <svg xmlns="http://www.w3.org/2000/svg" className="h-4 w-4" view
                        <path fillRule="evenodd" d="M12 13a1 1 0 100 2h5a1 1 0 001-1V9
                      </svg>
                    )}
                  </div>
                  <p className="text-sm text-gray-700">{insight.description}</p>
                </div>
              ))}
            </div>
          </div>
        )}
      </div>
    </div>
  );
}
```

Hour 7-8: Test Locally and Fix Issues

Run Development Server

bash

Copy

```bash
npm run dev
```

Visit http://localhost:3000 to test your application. Make sure everything works:

- Country and metric selection
- Data visualization
- Country insights

Fix any issues that appear.

Commit Changes

bash

Copy

```bash
git add .
git commit -m "Implement core application features"
git push
```

Hour 9-10: Deploy to Vercel

Set Up Vercel

1. Sign up for a free account at Vercel
2. Install Vercel CLI:

```bash
npm install -g vercel
```

Deploy

```bash
# Login to Vercel
vercel login

# Deploy the application
vercel

# Follow the prompts:
# - Set up and deploy? Yes
# - Which scope? [Your account]
# - Link to existing project? No
# - What's your project's name? global-wellness-perspectives
# - In which directory is your code located? ./
# - Want to override settings? No
```

Your application will be deployed to a URL like: `https://global-wellness-perspectives.vercel.app`

Hour 11-16: Add Secondary Components

Add SocietalStructuresContext Component

Copy the code from the societal-structures-context artifact to `components/SocietalStructuresContext.js`.

Update Main Page to Include SocietalStructuresContext

In `pages/index.js`, update your imports and add the component to the rendering section.

Test and Deploy Updates

bash

Copy

```bash
npm run dev  # Test locally
git add .
git commit -m "Add societal structures context component"
git push     # Vercel will automatically deploy updates
```

Hour 17-22: Add Educational Resources and FAQ

Add Educational Resources Component

Copy the code from the educational-resources artifact to `components/EducationalResources.js`.

Add FAQ Component

Copy the code from the faq-component artifact to `components/FAQ.js`.

Update Main Page

In `pages/index.js`, update your imports and add the components to the rendering section.

Test and Deploy Final Version

bash

Copy

```bash
npm run dev  # Test locally
git add .
git commit -m "Add educational resources and FAQ components"
git push     # Vercel will automatically deploy updates
```

Hour 23-24: Polish and Share

Create a Custom Domain (Optional)

In the Vercel dashboard:

1. Go to your project
2. Click "Settings" → "Domains"
3. Add your custom domain

Share Your Application

1. Create social media posts with your promotional video
2. Share the URL with colleagues and potential users
3. Gather initial feedback for future improvements

Monitoring and Analytics

Add Google Analytics (Optional)

1. Create a Google Analytics account
2. Get your measurement ID
3. Add the Google Analytics script to your Next.js app in `pages/_app.js`

Set Up Error Monitoring

Add Sentry or a similar tool for error monitoring to catch any production issues.

Post-Launch Next Steps

Once your application is live, consider these next steps:

1. **Gather User Feedback**: Add a simple feedback form
2. **Enhance Data**: Replace mock data with real data from public APIs
3. **Add Authentication**: For saving user preferences
4. **Mobile Optimization**: Further refine the mobile experience
5. **Performance Optimization**: Improve loading times and interactivity

This quickstart guide prioritizes getting a functional version online quickly. You can refine and expand the application incrementally after launch.