

Product Requirements Document

NASA Global Temperature Analysis

Interactive Data Visualization Platform

Version	1.0
Date	February 15, 2026
Project Type	Portfolio Project - Data Visualization
Deployment	Vercel (Next.js)

Executive Summary

This project creates an interactive web application that visualizes NASA's Global Surface Temperature Analysis (GISTEMP) data spanning 140+ years. The platform transforms complex climate data into engaging, accessible visualizations that reveal temperature trends, regional patterns, and the acceleration of global warming. This portfolio piece demonstrates advanced data analysis, statistical modeling, interactive visualization, and modern web development skills.

Project Goals

Primary Objectives

- Create an interactive, user-friendly platform for exploring global temperature data
- Demonstrate proficiency in data analysis, visualization, and web development
- Build a compelling portfolio piece that showcases technical and design skills
- Deploy a production-ready application accessible via custom domain

Success Metrics

- Application loads in under 3 seconds on standard connections
- All visualizations are interactive and responsive across devices
- Clean, documented codebase suitable for portfolio presentation
- Successfully deployed to Vercel with 99%+ uptime

Target Audience

Primary Audience

- Hiring managers and recruiters evaluating data science/analytics capabilities
- Technical leads assessing coding quality and system design skills
- Portfolio reviewers looking for production-quality projects

Secondary Audience

- General public interested in climate data visualization
- Students and educators seeking accessible climate change information
- Data visualization enthusiasts and developers

Core Features & Requirements

Feature 1: Global Temperature Timeline

Description: Interactive line chart showing global temperature anomalies from 1880 to present

Requirements:

- Display annual mean temperature anomaly with trend line
- Show statistical significance bands (confidence intervals)
- Enable zoom/pan functionality for detailed exploration
- Highlight key milestones (e.g., hottest years on record)
- Include hover tooltips with exact values and contextual information

Feature 2: Regional Comparison Dashboard

Description: Multi-series visualization comparing temperature trends across hemispheres, continents, and climate zones

Requirements:

- Compare Northern vs Southern Hemisphere trends
- Display Land vs Ocean temperature differences
- Allow users to toggle different regions on/off
- Calculate and display warming rates for each region

Feature 3: Geographic Heat Map

Description: Interactive world map showing spatial distribution of temperature changes

Requirements:

- Color-coded map showing temperature anomalies by region
- Time slider to show evolution over decades
- Highlight polar amplification (Arctic warming faster than global average)
- Click regions for detailed local statistics

Feature 4: Warming Stripes Visualization

Description: Famous 'warming stripes' graphic showing temperature progression as colored bands

Requirements:

- Generate stripes for global and regional data
- Allow customization of color scales and time periods
- Enable download as high-resolution image

Feature 5: Statistical Insights Panel

Description: Key statistics and data-driven insights presented clearly

Requirements:

- Display current global temperature anomaly
- Show warming rate in °C per decade
- List 10 warmest years on record
- Calculate and display acceleration metrics
- Compare current decade to historical averages

Feature 6: Data Download & API

Description: Allow users to export processed data and access via API

Requirements:

- Export data in CSV and JSON formats
- Provide simple REST API for programmatic access
- Include API documentation

Technical Requirements

Data Source

- **Primary:** NASA GISS Surface Temperature Analysis (GISTEMP v4)
- **URL:** <https://data.giss.nasa.gov/gistemp/>
- **Format:** CSV and NetCDF files
- **Update Frequency:** Monthly (automate data refresh)

Technology Stack

Frontend:

- Next.js 14+ (React framework with App Router)
- TypeScript for type safety
- Tailwind CSS for styling
- Plotly.js or Recharts for interactive visualizations
- Leaflet or Mapbox for geographic mapping

Data Processing:

- Python for ETL pipeline (pandas, numpy, scipy)
- Pre-process data into optimized JSON for frontend
- Statistical analysis with scipy.stats

Backend/API:

- Next.js API Routes for data endpoints
- Edge functions for performance
- Vercel KV or Vercel Postgres for caching (optional)

Deployment:

- Vercel platform (automatic CI/CD from GitHub)
- Custom domain with SSL
- CDN for static asset delivery

Version Control:

- Git with GitHub for repository hosting
- Branch strategy: main (production), develop (staging), feature branches
- Conventional commit messages

Performance Requirements

- Initial page load: < 3 seconds
- Time to Interactive: < 4 seconds
- Lighthouse Performance score: > 90
- Chart render time: < 500ms
- Mobile-responsive design (320px - 2560px widths)

Accessibility Requirements

- WCAG 2.1 Level AA compliance
- Keyboard navigation support
- Screen reader compatibility
- Colorblind-friendly palettes
- Text alternatives for all visualizations

User Experience Design

Visual Design Principles

- Clean, modern interface with professional aesthetics
- Data-first design: visualizations are the hero
- Consistent color scheme: blues for cool, reds for warm temperatures
- Generous white space for readability
- Professional typography (Inter or System fonts)

Navigation Structure

- Home: Overview with key statistics
- Global Trends: Timeline and trend analysis
- Regional Analysis: Compare different areas
- Geographic View: Interactive map
- About/Methodology: Data sources and calculations

Interaction Patterns

- Hover to reveal detailed information
- Click to zoom/filter
- Drag to pan through time
- Toggle switches for data series
- Smooth animations and transitions

Data Processing Pipeline

Step 1: Data Acquisition

1. Download NASA GISTEMP datasets (global, hemispheric, zonal means)
2. Verify data integrity with checksums
3. Store raw data in /data/raw directory

Step 2: Data Cleaning

4. Handle missing values (interpolation for short gaps, flagging for long gaps)
5. Convert temperature anomalies to consistent units
6. Validate data ranges and flag outliers
7. Standardize date formats

Step 3: Feature Engineering

8. Calculate rolling averages (5-year, 10-year windows)
9. Compute trend lines using linear regression
10. Calculate acceleration metrics (rate of change over time)
11. Generate decadal averages for comparison
12. Identify statistical change points

Step 4: Statistical Analysis

13. Calculate confidence intervals for trends
14. Perform significance testing (Mann-Kendall trend test)
15. Compute correlation matrices between regions
16. Generate summary statistics (mean, median, std dev by period)

Step 5: Output Generation

17. Export processed data as optimized JSON files
18. Create aggregated datasets for performance (monthly, annual, decadal)
19. Generate metadata files with data dictionary and statistics
20. Store processed data in /public/data directory for frontend access

Development Phases & Timeline

Phase 1: Setup & Data Pipeline (Week 1)

- Initialize Next.js project with TypeScript
- Set up GitHub repository with proper structure
- Create Python ETL scripts for data processing
- Download and process NASA data
- Generate initial statistical analysis

Phase 2: Core Visualizations (Week 2)

- Implement global temperature timeline chart
- Build regional comparison dashboard
- Create statistical insights panel
- Implement responsive layout

Phase 3: Advanced Features (Week 3)

- Develop interactive geographic heat map
- Add warming stripes visualization
- Implement data export functionality
- Build API endpoints

Phase 4: Polish & Deployment (Week 4)

- Optimize performance (code splitting, lazy loading)
- Add loading states and error handling
- Implement SEO optimization
- Write comprehensive documentation
- Deploy to Vercel with custom domain
- Conduct final testing across devices

Quality Assurance

Testing Requirements

- Data accuracy validation against source files
- Cross-browser testing (Chrome, Firefox, Safari, Edge)
- Mobile device testing (iOS and Android)
- Performance benchmarking
- Accessibility audit using axe DevTools

Documentation Requirements

- Comprehensive README with project overview
- Code documentation and inline comments
- API documentation with example requests
- Data methodology explanation
- Setup and deployment instructions

Risks & Mitigation Strategies

Risk	Impact	Mitigation
Data file format changes	Pipeline breaks	Version datasets, add format validation
Performance with large datasets	Slow load times	Aggressive data aggregation, lazy loading
Browser compatibility issues	Features break	Use polyfills, progressive enhancement
Statistical calculation errors	Inaccurate insights	Unit tests, cross-validation with published results

Future Enhancements (Post-MVP)

- Add CO2 concentration data overlay for correlation analysis
- Implement simple climate projection models
- Compare multiple climate datasets (NOAA, Berkeley Earth, HadCRUT)
- Add extreme weather event markers on timeline
- Create shareable custom reports
- Integrate with other climate indicators (sea level, Arctic ice)
- Add multi-language support

Project Success Criteria

The project will be considered successful if:

21. All core visualizations are functional and interactive
22. Application achieves performance targets (< 3s load, > 90 Lighthouse score)
23. Data accuracy validated against NASA source files

24. Responsive design works seamlessly on mobile, tablet, and desktop
25. Code is clean, documented, and follows best practices
26. Successfully deployed to Vercel with custom domain
27. GitHub repository is well-organized with comprehensive README
28. Project effectively demonstrates technical skills for portfolio purposes

Appendix

Useful Resources

- **NASA GISTEMP:** <https://data.giss.nasa.gov/gistemp/>
- **NASA Data Documentation:** <https://data.giss.nasa.gov/gistemp/faq/>
- **Next.js Documentation:** <https://nextjs.org/docs>
- **Plotly.js:** <https://plotly.com/javascript/>
- **Vercel Deployment:** <https://vercel.com/docs>
- **Warming Stripes:** <https://showyourstripes.info/>