PROJECT ABSTRACT

Disk Usage Visualizer with Treemap and Sunburst Views

A client-side web application that turns a filesystem snapshot into interactive treemap and sunburst visualizations with drill-down, search filtering and per-folder analytics. Delivered built-to-order with the working app, source code, user guide, report, PPT and viva kit.

01. Title

Disk Usage Visualizer with Treemap and Sunburst Views

02. Introduction / Background

Disks fill up silently: a developer's machine accumulates Docker layers, node_modules folders, video renders and forgotten downloads until the operating system starts complaining. Built-in tools list folders by size in a flat table, which hides the structure — a 2 GB folder of videos looks identical to a 2 GB folder of source code. Area-based visualization solves this: a treemap makes each file and folder a rectangle sized by its bytes, so space hogs are visible at a glance, while a sunburst shows the same hierarchy radially. This project implements both views with drill-down navigation, search filtering and category coloring in a single-file web app.

03. Problem Statement

Students understand 'disk full' but rarely understand where the bytes went, and flat size listings do not teach hierarchical thinking about storage. The project addresses this with an interactive visualizer: a realistic filesystem snapshot is rendered as a squarified treemap and a two-level sunburst, with click-to-drill-down, breadcrumb navigation, live search and a per-folder analytics panel — demonstrating data-structure traversal, layout algorithms and canvas rendering in one build.

04. Objectives

- Model a realistic filesystem snapshot (folders, files, sizes, categories) as a JSON tree.
- Implement the squarified treemap layout algorithm rendering to HTML canvas.
- Implement a two-level sunburst chart with arc hit-testing for the same data.
- Provide drill-down navigation with breadcrumbs, hover tooltips and a top-items panel.
- Add live search filtering across files and folders.
- Color items by category (code, media, documents, system, archives) with a legend.
- Deliver the complete student kit: app, source code, user guide, report, PPT and viva Q&A.;

05. Proposed Methodology

The app is a single-file HTML/CSS/JS build with no dependencies. (1) Data layer: a filesystem snapshot is embedded as a nested JSON tree; a rollup pass computes subtree sizes, file counts and dominant categories. (2) Layout layer: the squarified treemap algorithm partitions

rectangles by size; the sunburst maps sizes to arc angles across two rings. (3) Interaction layer: canvas hit-testing drives hover tooltips and click drill-down; breadcrumbs track the path; the search box filters the tree live. (4) Presentation layer: a dark analytics dashboard with a selected-folder panel, summary cards and a ranked top-items list.

06. System Architecture / Working

On load, a scan animation plays while the rollup pass aggregates the snapshot. The treemap view lays out the current folder's children as area-proportional rectangles; the sunburst view draws them as arc segments with a second ring for grandchildren. Clicking a rectangle or arc drills into that folder and updates the breadcrumbs, the selected-folder panel (size, file count) and the top-items list. Typing in the search box filters the tree and re-renders instantly. Hovering any shape shows a tooltip with name, size and category.

07. Hardware / Software Requirements

- Any modern web browser — no installation, no server, runs offline
- HTML5 canvas, vanilla JavaScript (ES6): layout algorithms, hit-testing, JSON data model
- Single-file delivery: one .html file
- Optional: any static host for sharing the demo link

08. Expected Outcomes

- A working visualizer with treemap and sunburst views of a realistic filesystem snapshot.
- Drill-down navigation, search filtering and per-folder analytics panels.
- A documented implementation of the squarified treemap algorithm for the report.
- A complete report, PPT and viva Q&A; on visualization algorithms and the app's design.

09. Applications

- Teaching hierarchical data structures and layout algorithms.
- Demonstrating canvas-based data visualization as a final-year software project.
- A template for storage-audit dashboards in system-administration courses.

10. Future Scope

- Live scanning of the real filesystem via a companion desktop agent.
- Duplicate-file detection with hash-based grouping.
- Historical snapshots with growth-over-time comparison.
- Export of audit reports (largest folders, stale files) as PDF.