🧠Advanced Disk Scheduling Simulator

A full-stack Operating System project that simulates popular disk scheduling algorithms and visualizes head movement with interactive graphs and animations.

Perfect for: - 🎓College OS projects - 📚Learning disk scheduling algorithms - 🖥️Understanding seek time optimization - 🧪Simulation and visualization practice

---

😢Features

Frontend (HTML + CSS + React)

- Fully responsive, modern UI
- Beautiful glassmorphism card design
- Smooth head-movement animation
- Real-time charts using Chart.js
- Input validation and clean UX
- Works on any browser (Chrome recommended)

Backend (Python Flask)

- Fast REST API for simulation
- CORS enabled for frontend communication
- Clean `scheduler.py` with all algorithms implemented

Algorithms Supported

🔧FCFS (First Come First Serve)
🔧SSTF (Shortest Seek Time First)
🔧SCAN (Elevator Algorithm)
🔧C-SCAN (Circular SCAN)
🔧LOOK
🔧C-LOOK

👂 Project Structure

```
disk-scheduler/
│
├── frontend/
│   ├── index.html
│   ├── style.css
│   └── app.js
│
└── backend/
    ├── app.py
```

```
    ├── scheduler.py
    └── requirements.txt
```

👉🏿Installation & Setup

Follow these steps to run the project on your laptop.

🤩Install Backend Requirements

Open terminal:

```
cd disk-scheduler/backend
pip install -r requirements.txt
```

🤪Run Backend (Flask Server)

```
python app.py
```

You must see:

```
Running on http://127.0.0.1:5000
```

🤫Run Frontend

Option A — Using VS Code Live Server (Recommended)

Right-click `index.html` → **Open with Live Server**

Option B — Using Python HTTP Server

```
cd disk-scheduler/frontend
python -m http.server 8000
```

Open:

```
http://localhost:8000
```

👹Visualizations Included

👈🏽Disk Head Movement Animation

A horizontal track with: - Moving head indicator - Dot markers for all requests - Smooth transitions

👈🏽Seek Distance Bar Graph

Shows distance moved at each step.

👈🏽Head Movement Line Graph

Visualizes head position per time step.

😽Sample Input

```
95 180 34 119 11 123 62 64
```

Algorithm selected: `SCAN`

👇🏽Sample API Request

`POST /simulate`

```
{
  "requests": [95, 180, 34, 119],
  "initial_head": 50,
  "algorithm": "SSTF"
}
```

👇🏽Sample Output

```
{
  "algorithm": "SCAN",
  "order": [50, 62, 64, 95, 119, 123, 180],
  "seek_sequence": [12,2,31,24,4,57],
  "total_seek": 130
}
```

👃 Algorithms Explained Briefly

FCFS

Serves requests in the order they arrive.

SSTF

Selects the closest request from current head position.

SCAN

Moves like an elevator: one direction to end → reverse.

C-SCAN

Moves in one direction only; jumps back to start.

LOOK / C-LOOK

Same as SCAN/C-SCAN but stops at last request instead of disk end.

🎉Technologies Used

Frontend

- HTML5, CSS3
- React (CDN)
- Chart.js
- Modern UI Styling

Backend

- Python
- Flask
- Flask-CORS

👉How to Upload to GitHub

In project root:

```
git init
git add .
git commit -m "Initial commit"
git branch -M main
```

```
git remote add origin https://github.com/YOUR-USERNAME/disk-scheduler.git
git push -u origin main
```

👆 Screenshots

(Add screenshots here once you upload images.)

👂 Final Notes

This project is fully ready for: - College submission - Resume portfolio - Learning OS scheduling - Extending into a full simulator

Need help adding **feature comparison**, **export to CSV**, **dark mode**, or **deployment on Render/AWS/ Heroku**? Just ask!