Performance Testing Plan: Data Analytics Dashboard

---

1. Quick Setup for Testing

Before You Begin

- Open your project in VS Code (or any editor)
- Install Live Server extension (for local testing)
- Chrome/Firefox is recommended

---

2. Real-Time Testing (While You Code)

A. Manual Testing - Basic Checks

1. **Page Load Speed**
   - Open Chrome DevTools (`F12` > "Network" tab)
   - Refresh the page
   - Look for:
     - Total load time (should be under 3s)
     - Large files (>500KB)
2. **Chart Rendering Test**
   - Add this to your browser console after page loads:

javascript3 lines

Click to expand

```
console.time('chartRender');

Chart.getChart('myAreaChart');

...
```

- Should show less than 1s for simple charts

3. **PDF Generation Test**
   Click your export button and check:
   - Does it work cross-browser?
   - Time taken (should be under 5s for 3 charts + table)

B. Automated Tools

1. **Lighthouse Audit** (In Chrome DevTools):
   - `F12` > "Lighthouse" > "Generate report"
   - Target: "Performance" score above 80/100
2. **Network Throttling Test**:
   - In DevTools: `Ctrl+Shift+P` > "Show Network conditions"
   - Simulate "Slow 3G" connection
   - Check if charts still load (even if slowly)

---

3. Load Testing (Advanced)

Simulate Multiple Users

(Requires Node.js installed)

1. Install **Artillery** (load testing tool):

bash

RunCopy code

```bash
npm install -g artillery
```

2. Create `load-test.yml`:

yaml

RunCopy code

```yaml
config:
  target: "http://localhost:5500" # Your Live Server URL
  phases:
    - duration: 60
      arrivalRate: 5 # 5 users per second
scenarios:
  - flow:
      - get:
          url: "/" # Your dashboard path
```

3. Run the test:

bash

RunCopy code

```bash
artillery run load-test.yml
```

Look for errors and response times in results.

---

4. Measuring Performance

Key Metrics to Log

Test Case	Tool	Good Result
Page Load	DevTools	<3s
Chart Render	console.time()	<1s
PDF Export	Manual click	<5s
50 Users	Artillery	No crashes

Sample Performance Log

RunCopy code

```
[2024-02-20]
- Initial load: 2.8s
- Chart render: 0.7s
- PDF export: 3.2s
- 50-user test: 0.1% errors
```

---

5. Fixing Common Issues

If Charts Load Slowly

javascript7 lines

Click to expand

```javascript
// In your Chart.js config:
options: {
...
```

If PDF Export is Slow

javascript6 lines

Click to expand

```
// In html2pdf config:

jsPDF: {

...
```

If Pages Load Slowly

- Optimize images with Squoosh
- Use browser caching by adding to your HTML:

html1 lines

Click to expand

```
<meta http-equiv="Cache-control" content="public, max-age=3600">
```

---

6. What You Should See When Done

□ Page loads under 3 seconds
□ Charts render instantly
□ PDF exports in under 5 seconds
□ Handles 50+ simulated users without crashing