








Section Code Extractor - Chrome Extension

A powerful Chrome extension designed to extract HTML, CSS, and JavaScript from specific webpage sections, perfect for recreating designs in Elementor WordPress.

Features

-  **Visual Element Selection:** Hover and click to select any webpage section
-  **Clean Code Extraction:** Get formatted, indented, ready-to-use code
-  **Three Code Types:** Extract HTML, CSS, and JavaScript separately
-  **Export as ZIP:** Download complete section files in organized folders
-  **Elementor-Ready:** Optimized output for WordPress Elementor integration
-  **Context Menu Integration:** Right-click any element to extract
-  **Syntax Highlighting:** Beautiful code display with Prism.js

Installation

Method 1: Load Unpacked Extension (Development)

1. Download/Clone the Extension Files

- Create a folder named `section-code-extractor`
- Add all the files from the artifacts above

2. Create Icon Images (Required)

- Create an `icons` folder inside the extension directory
- Add three icon files:
 - `icon16.png` (16x16 pixels)
 - `icon48.png` (48x48 pixels)
 - `icon128.png` (128x128 pixels)
- You can use any icon or create simple colored squares as placeholders

3. Load Extension in Chrome

- Open Chrome and go to `chrome://extensions/`
- Enable "Developer mode" (toggle in top-right corner)
- Click "Load unpacked"
- Select the `section-code-extractor` folder
- Extension should now appear in your extensions list

Method 2: Create Icons Programmatically

If you don't have icon images, create simple placeholder icons using HTML Canvas:

```
html

<!-- Create this as icon-generator.html and open in browser -->
<!DOCTYPE html>
<html>
<body>
<canvas id="canvas"></canvas>
<script>
const sizes = [16, 48, 128];
sizes.forEach(size => {
  const canvas = document.getElementById('canvas');
  canvas.width = size;
  canvas.height = size;
  const ctx = canvas.getContext('2d');

  // Gradient background
  const gradient = ctx.createLinearGradient(0, 0, size, size);
  gradient.addColorStop(0, '#667eea');
  gradient.addColorStop(1, '#764ba2');
  ctx.fillStyle = gradient;
  ctx.fillRect(0, 0, size, size);

  // Text
  ctx.fillStyle = 'white';
  ctx.font = `bold ${size/2}px Arial`;
  ctx.textAlign = 'center';
  ctx.textBaseline = 'middle';
  ctx.fillText('<>', size/2, size/2);

  // Download
  canvas.toBlob(blob => {
    const url = URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.href = url;
    a.download = `icon${size}.png`;
    a.click();
  });
});
</script>
</body>
</html>
```

File Structure

```
section-code-extractor/  
├── manifest.json      # Extension configuration  
├── background.js      # Background service worker  
├── content.js         # Content script for element selection  
├── content.css        # Styles for element highlighting  
├── popup.html         # Popup UI structure  
├── popup.css          # Popup UI styles  
├── popup.js           # Popup UI logic  
├── icons/  
│   ├── icon16.png    # 16x16 icon  
│   ├── icon48.png    # 48x48 icon  
│   └── icon128.png   # 128x128 icon  
└── README.md         # Documentation
```

Usage

Quick Start

1. Activate Highlight Mode

- Click the extension icon in Chrome toolbar
- Click "Toggle Highlight Mode" button
- OR right-click anywhere → "Toggle Highlight Mode"

2. Select a Section

- Hover over webpage elements to see them highlighted in green
- Click on the desired section to extract its code

3. View Extracted Code

- Click the extension icon to open the popup
- Switch between HTML, CSS, and JavaScript tabs
- Click "Copy" buttons to copy code to clipboard

4. Export for Elementor

- Click "Export as ZIP" button
- Download contains organized files ready for use

Context Menu Options

Right-click on any webpage element to access:

- **Extract Section Code** - Extract code from clicked element
- **Toggle Highlight Mode** - Enable/disable element selection mode

Using with Elementor

Step 1: Extract the Section

1. Navigate to the website you want to replicate
2. Activate highlight mode and select the section
3. Export as ZIP or copy individual code sections

Step 2: Import to WordPress

HTML:

- In Elementor, add an "HTML" widget
- Paste the extracted HTML code
- Adjust container settings as needed

CSS:

- Go to Elementor → Custom CSS (bottom of left panel)
- Paste the extracted CSS code
- OR use "Custom CSS" section in Theme Customizer

JavaScript:

- Install "Code Snippets" or "Insert Headers and Footers" plugin
- Add the JavaScript code
- OR add to your theme's `functions.php` wrapped in script tags

Step 3: Cleanup & Optimization

1. Remove Unnecessary Classes

- Clean up auto-generated class names
- Keep only structural and styling classes

2. Fix Asset URLs

- Replace relative URLs with absolute URLs for images
- Upload images to WordPress media library
- Update image paths in HTML/CSS

3. Adjust Responsiveness

- Test on mobile, tablet, desktop
- Add/modify media queries as needed
- Use Elementor's responsive controls

4. Optimize for Performance

- Compress images
- Minify CSS/JS if needed
- Remove unused styles

Advanced Features

Filtering CSS Output

The extension extracts important CSS properties including:

- Layout (display, position, flex, grid)
- Spacing (margin, padding)
- Sizing (width, height, max/min values)
- Background & borders
- Typography
- Transforms & animations

JavaScript Detection

The extension captures:

- Inline event handlers (onclick, onchange, etc.)
- Inline script tags within the section
- Direct event listener assignments

Note: External JS libraries and complex event listeners may not be fully captured.

Configuration

Modifying Extracted Properties

Edit `content.js`, locate the `importantProps` array in `extractCSS()` function:

```
javascript

const importantProps = [
  'display', 'position', 'top', 'left',
  // Add or remove properties as needed
];
```

Changing Highlight Color

Edit `content.css`:

CSS

```
#section-extractor-overlay {  
  border: 3px solid #4CAF50; /* Change color here */  
  background: rgba(76, 175, 80, 0.1);  
}
```

Troubleshooting

Extension Not Loading

- Verify all files are in the correct directory
- Check that icon files exist in `(icons/)` folder
- Look for errors in `(chrome://extensions/)` page

Elements Not Highlighting

- Ensure highlight mode is activated
- Check browser console for errors (F12)
- Try refreshing the webpage
- Some sites may have security restrictions

Code Not Copying

- Grant clipboard permissions when prompted
- Check if popup has focus
- Try using keyboard shortcut (Ctrl+C) as fallback

Incomplete CSS Extraction

- Some styles may be in external stylesheets blocked by CORS
- Inline styles are always captured
- Computed styles reflect the final rendered state

JavaScript Not Working

- Only inline JS and event handlers are captured
- External libraries need to be included separately
- Event listeners added via frameworks may not be detected

Privacy & Permissions

The extension requires:

- **activeTab**: Access current tab content
- **contextMenus**: Add right-click menu options
- **scripting**: Inject code for element selection
- **storage**: Save extracted data temporarily
- **host_permissions**: Access webpage DOM

No data is sent to external servers. All processing happens locally.

Best Practices

1. Start with Clean Pages

- Extract from production sites, not development environments
- Avoid pages with excessive scripts or tracking

2. Extract Modularly

- Extract small, reusable sections
- Combine sections in Elementor rather than extracting entire pages

3. Test Before Deployment

- Always test extracted code in a staging environment
- Check for conflicts with WordPress/Elementor scripts

4. Respect Copyright

- Only extract code for inspiration or learning
- Don't replicate copyrighted designs without permission

Development

Debugging

Enable developer mode and check:

- **Background Script**: `chrome://extensions/` → Extension → "Service Worker"
- **Content Script**: Press F12 on webpage → Console tab
- **Popup Script**: Right-click popup → "Inspect"

Modifying Code

After making changes:

1. Go to `chrome://extensions/`
2. Click refresh icon on your extension
3. Reload any open tabs where extension is active

Technical Details

Technologies Used

- Chrome Extension Manifest V3
- Vanilla JavaScript (ES6+)
- Prism.js for syntax highlighting
- JSZip for file compression
- CSS Grid & Flexbox for UI layout

Browser Compatibility

- Chrome 88+
- Edge 88+
- Brave (Chromium-based)
- Opera (Chromium-based)

Note: Not compatible with Firefox (uses Manifest V2)

Contributing

Feel free to modify and enhance this extension for your needs!

License

This extension is provided as-is for educational and development purposes.

Support

For issues or questions:

1. Check the Troubleshooting section
2. Review browser console for errors
3. Verify all files are correctly structured

Made for Elementor developers and web designers 🧑‍💻

Happy extracting! 🚀