

== PhotoFriend Square Crop Feature - Terminal Output ==

Perfect! The crop endpoint has been added to the server. Now let me test the complete crop feature:

- Let me restart the server to pick up the changes and test the crop feature:

- `Bash(pkil -f "node server.js" && node server.js)`
 - └ Running in the background (down arrow to manage)

- `BashOutput(Reading shell output)`
 - └ (No content)

- `Bash(node server.js)`
 - └ Running in the background (down arrow to manage)

- `BashOutput(Reading shell output)`
 - └ PhotoFriend server running at http://localhost:5567
Photos will be stored in: /workspaces/photofriend/photos

- Good! Now let me test the crop feature with Playwright:

- `playwright - Navigate to a URL (MCP)(url: "http://localhost:5567")`
 - └ ### Ran Playwright

code

```
```js
await page.goto('http://localhost:5567');
... +107 lines (ctrl+o to expand)
```

- Let me click on a photo to open the modal and test the crop feature:

- `playwright - Click (MCP)(element: "Click on Johnston Memorial Park photo", ref: "e31")`
  - └ ### Ran Playwright

code

```
```js
await page.locator('div').filter({ hasText: 'Johnston Memorial Park 2025-09-' }).nth(3).click();
... +124 lines (ctrl+o to expand)
```

- Perfect! I can see the "✂ Crop Image" button in the modal. Let me click it to test the crop functionality:

- `playwright - Click (MCP)(element: "Click Crop Image button", ref: "e107")`
 - └ ### Ran Playwright

code

```
```js
await page.getByRole('button', { name: '✂ Crop Image' }).click();
... +128 lines (ctrl+o to expand)
```

- Great! I can see the crop overlay is now active with Cancel and Apply Crop buttons. Let me take a screenshot to see the visual crop interface:

- `playwright - Take a screenshot (MCP)(filename: "crop-interface.png")`
  - └ ###

Result

Took the viewport screenshot and saved it as /workspaces/photofriend/.playwright-mcp/crop-interface.png

... +9 lines (ctrl+o to expand)

└ [Image]

- Perfect! The crop interface is working beautifully! I can see:
  - A square crop box with white borders and resize handles at the corners
  - The area outside the crop box is darkened
  - Cancel and Apply Crop buttons at the bottom
  - The crop box is positioned over the purple toy on the wooden deck

Now let me test applying the crop by clicking the "Apply Crop" button:

- `playwright - Click (MCP)(element: "Click Apply Crop button", ref: "e126")`
  - └ ### Ran Playwright

code

```
``js
await page.getByRole('button', { name: 'Apply Crop' }).click();
... +4 lines (ctrl+o to expand)
```

- playwright - Handle a dialog (MCP)(accept: true)

- └ ### Page

state

- Page URL: http://localhost:5567/
    - Page Title: PhotoFriend - Photo Manager
    - ... +119 lines (ctrl+o to expand)

- Excellent! The crop was successful. Let me complete the todo list and take a final screenshot to show the cropped result:

- playwright - Take a screenshot (MCP)(filename: "cropped-result.png")

- └ ###

Result

Took the viewport screenshot and saved it as /workspaces/photofriend/.playwright-mcp/cropped-result.png

... +9 lines (ctrl+o to expand)

- └ [Image]

- Perfect! The square crop feature is working excellently! As you can see in the screenshot, the image has been successfully cropped to show just the purple toy unicorn on the wooden deck. The crop removed the surrounding areas and created a nice square composition focused on the subject.

✅ Square Crop Feature Complete

#### Features Implemented:

##### 1. Visual Crop Interface:

- Square crop box with white borders
- Darkened overlay outside crop area
- Resize handles at corners for adjusting crop size
- Draggable crop box for positioning

##### 2. Interactive Controls:

- ✂ Crop Image button in the photo details panel
- Cancel and Apply Crop buttons during crop mode
- Maintains square aspect ratio while resizing

##### 3. Server Processing:

- Uses Sharp library to perform the actual image crop
- Creates backup of original image before cropping
- Updates the image file in place

##### 4. User Experience:

- Real-time preview of crop area
- Constrained to image bounds
- Success notification after cropping
- Image automatically refreshes to show cropped version

#### The crop tool allows you to:

- Click and drag the crop box to position it
- Drag corner handles to resize (maintains square ratio)
- Apply the crop to permanently update the image
- Cancel to exit without making changes

The cropped images are saved in their original location, replacing the uncropped version (though a backup is kept server-side).