

Complete Project Structure

Here's the full project structure with all the files needed to implement the trading panel:

```
your_project/
|
├── web_interface.py          # Modified Flask application (provided)
├── coinex_trader.py         # Your existing trading logic
|
├── templates/
|   ├── base.html           # Base template (provided)
|   ├── dashboard.html      # Dashboard template (provided)
|   ├── trading_panel.html  # Trading panel component (provided)
|   ├── index.html          # Your existing template
|   ├── error.html          # Your existing error template
|   └── setup.html          # Your existing setup template
|
├── static/
|   ├── css/
|   |   ├── main.css        # Your existing CSS (if any)
|   |   └── trading_panel.css # Trading panel styles (provided)
|   └── js/
|       ├── main.js         # Your existing JS (if any)
|       └── trading_panel.js # Trading panel functionality (provided)
|
└── trader_config.json       # Your existing configuration file
```

Implementation Steps

1. Create Folder Structure (if needed)

Make sure you have the following directory structure:

```
bash

# Create directories if they don't exist
mkdir -p templates
mkdir -p static/css
mkdir -p static/js
```

2. Save New Files

Download and save each file to its appropriate location:

- Save "Modified web_interface.py" as `web_interface.py` (replace your existing file)
- Save "Simple Trading Panel" as `templates/trading_panel.html`
- Save "Dashboard Template" as `templates/dashboard.html`
- Save "Base Template" as `templates/base.html`
- Save "Trading Panel CSS" as `static/css/trading_panel.css`
- Save "Trading Panel JavaScript" as `static/js/trading_panel.js`

3. Create main.css and main.js (if needed)

If these files don't exist yet, create them with basic content:

static/css/main.css:

css

```
/* Main application styles */
body {
  background-color: #121212;
  color: #e0e0e0;
}

.flash-messages {
  position: fixed;
  top: 20px;
  right: 20px;
  z-index: 1000;
  max-width: 350px;
}
```

static/js/main.js:

javascript

```
// Common application functionality
document.addEventListener('DOMContentLoaded', function() {
  // Auto-dismiss flash messages after 5 seconds
  setTimeout(function() {
    const alerts = document.querySelectorAll('.alert');
    alerts.forEach(function(alert) {
      const closeBtn = alert.querySelector('.btn-close');
      if (closeBtn) closeBtn.click();
    });
  }, 5000);
});
```

4. Run Your Application

Start your Flask application:

```
bash
```

```
python web_interface.py
```

Navigate to [/dashboard](#) to see your new trading panel!