For the Python part, I start by fetching the Yahoo Finance “most active” page with the requests library, making my request look like it comes from a real browser. Once I get a successful response, I save the HTML locally and hand it to BeautifulSoup to parse. I find the table whose first header is “Symbol,” build a map from each column name to its position, and then loop over every row in the table body. For each row I pull out the symbol, name, price, change, and volume text, trim away extra spaces, and assemble a simple dictionary. Finally I connect to MongoDB, clear out any old records, and insert each dictionary as its own document so the database always holds the current snapshot of active stocks.

For the PHP part, I load the Composer autoloader so I can use the MongoDB client library. I connect to the same database and collection, then look at the URL query string to see which field the user wants to sort by, defaulting to the index order if nothing is specified. I run a MongoDB find query with that sort option and then build an HTML page around it. In the table header I turn each column name into a clickable link that reloads the page sorted by that field. In the body I loop through each result document and print its values in the right order. The whole page is wrapped in standard HTML tags and styled with simple CSS so it is easy to read and lets the user sort on any column with a single click.