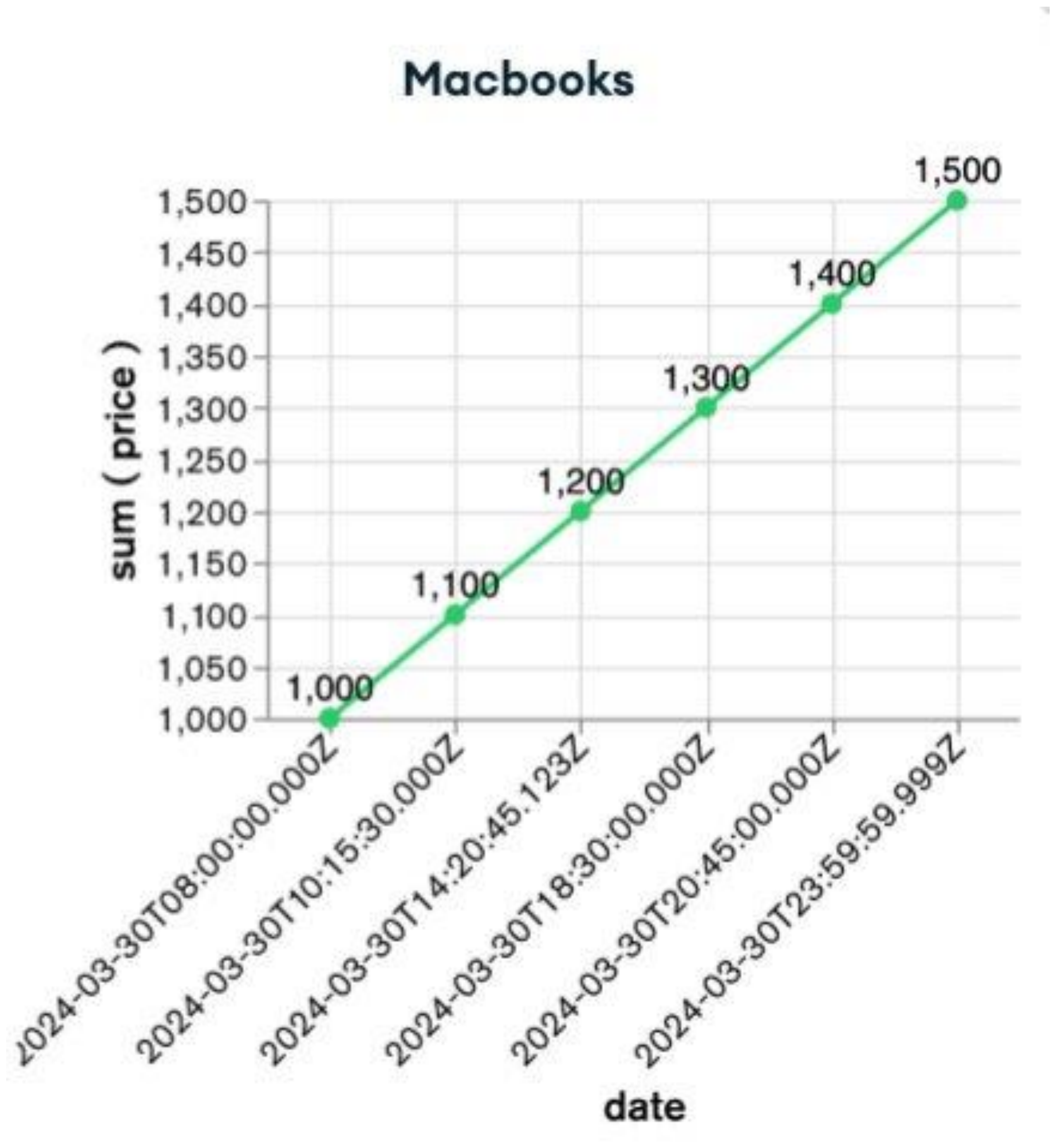
# Product Price Analysis App

Christopher Calixte, Liam Daly, Tyler Martin, Alexis Nunez

## Purpose

Our team will achieve this product through a Kubernetes cluster to handle the web scraping, data cleansing and saving to a database, and the presentation of the data through an interactive UI. Containers using a combination of Linux and Python will be utilized for web scraping and cleaning, while a lightweight Linux distribution node will handle the database management; Another node will handle the web server for the UI. By using Kubernetes, we can readily deploy containers and build web scraping nodes to scale to scale and increase speed by each container performing its own data cleansing.

## User Interface



## Stakeholder Value:

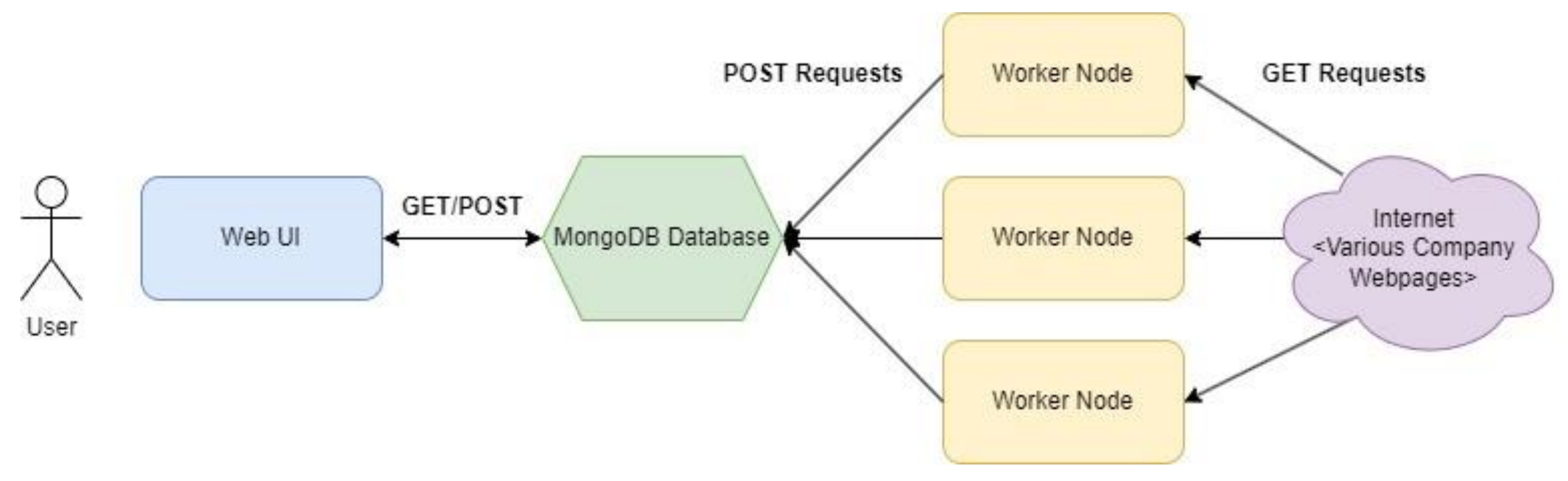
Save users time and money
and gain insights to price fluctuations.

## Database Schema

```
{"product": "macbook", "price": 1000, "date": "March 30, 2024"},
{"product": "mouses", "price": 15, "date": "March 30, 2024"}
```

## Architectural Design

- Alpine Linux webscraper node detects price change or pulls price at irregular intervals.
- Data is pulled down to worker node. This will be performed using a Python script which utilizes the BeautfiulSoup web scraping library.
- Data is parsed for price and time. Parsing will be done with a Python script to pull the price from the data, depending on the storefront.
- Worker node sends POST with cleaned information to MongoDB database server
- MongoDB writes data to database taking into account product name, product price, and time of capture.
- User opens nginx web server UI to see MongoDB Atlas chart
- Data is loaded into UI with information for user. This frontend will provide all needed information with a visually pleasing UI



## Deployment