MICHIGAN STATE UNIVERSITY

Project Plan Presentation
Amazon Shop Smart: Web Extension for Shopping

The Capstone Experience

Team Amazon

Jiashang Cao
Richard Huang
Emma Sickelsteel
Jimmy Warner
Hithesh Yedlapati
Tianli Zhou

Department of Computer Science and Engineering
Michigan State University

Spring 2022

From Students…
…to Professionals

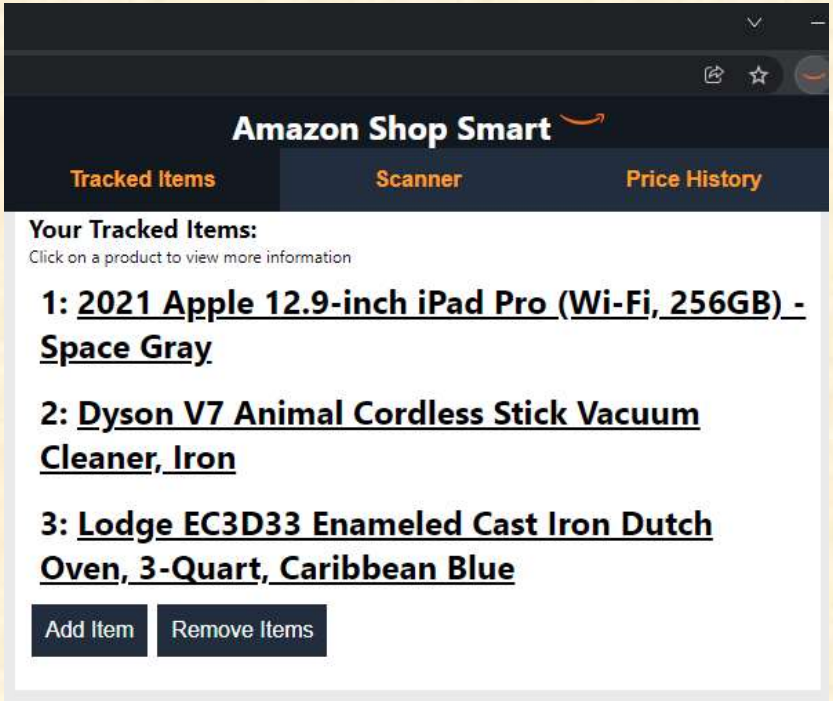
Functional Specifications

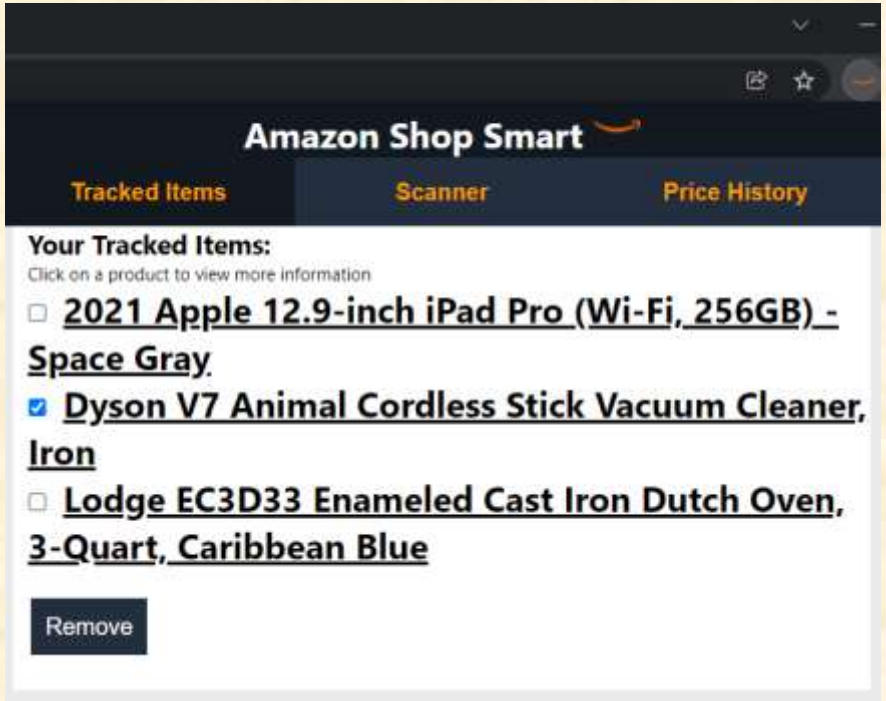
- Find related items on Amazon.com when shopping on other sites and compare price and shipping costs.
- Allows viewing of product price information.
- Sends user notification via email when a product price has changed.
- Inform member about the suggested purchasing time (e.g. the best savings occur during Christmas).
- Empower members to make best-informed purchasing decision.

Design Specifications

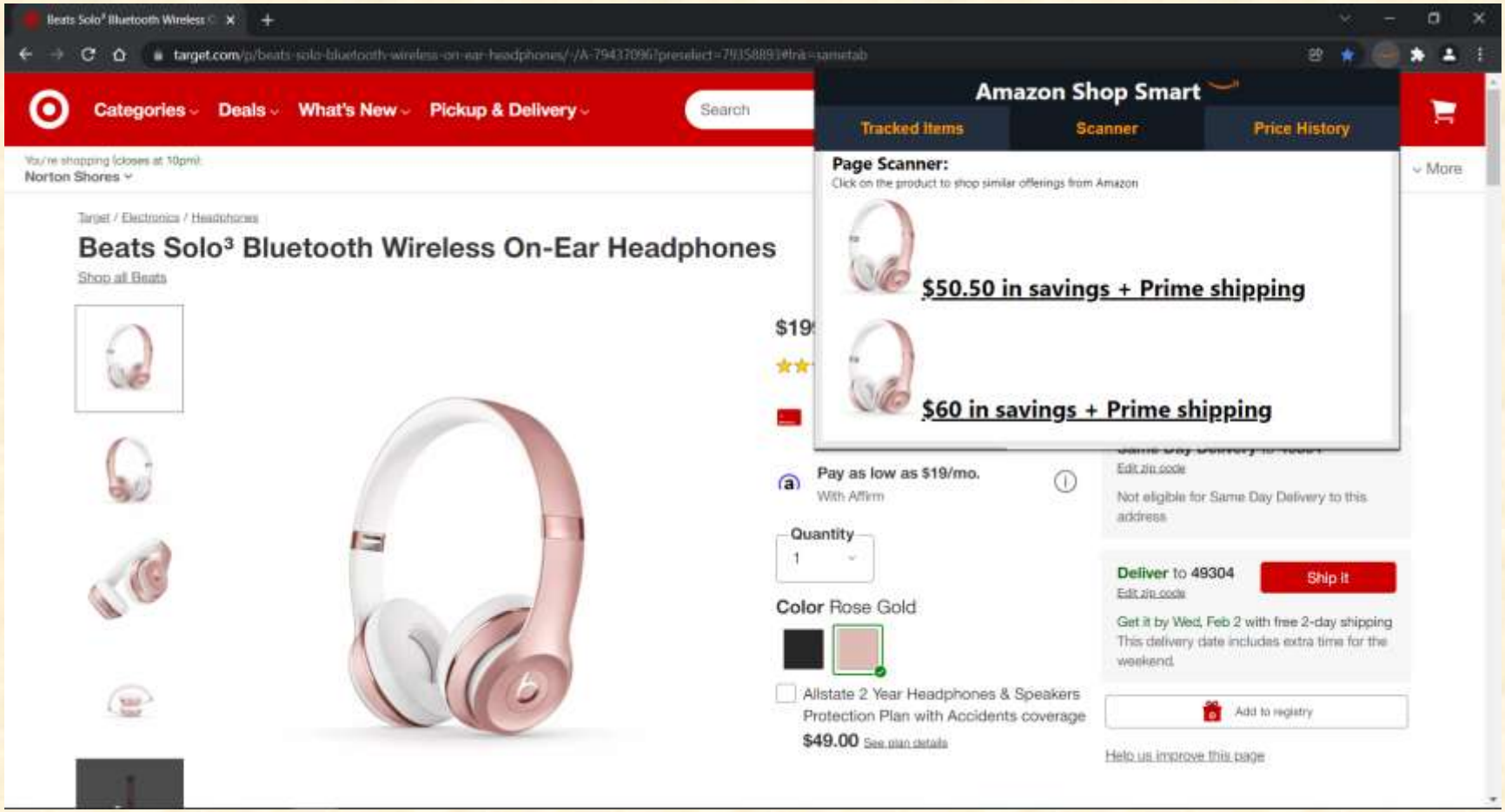
- Pop-up extension
- List of tracked Amazon products
- List of products on Amazon while browsing other retailers
- Graph of product price history
- Automated email notification for price updates

Screen Mockup: Track Items

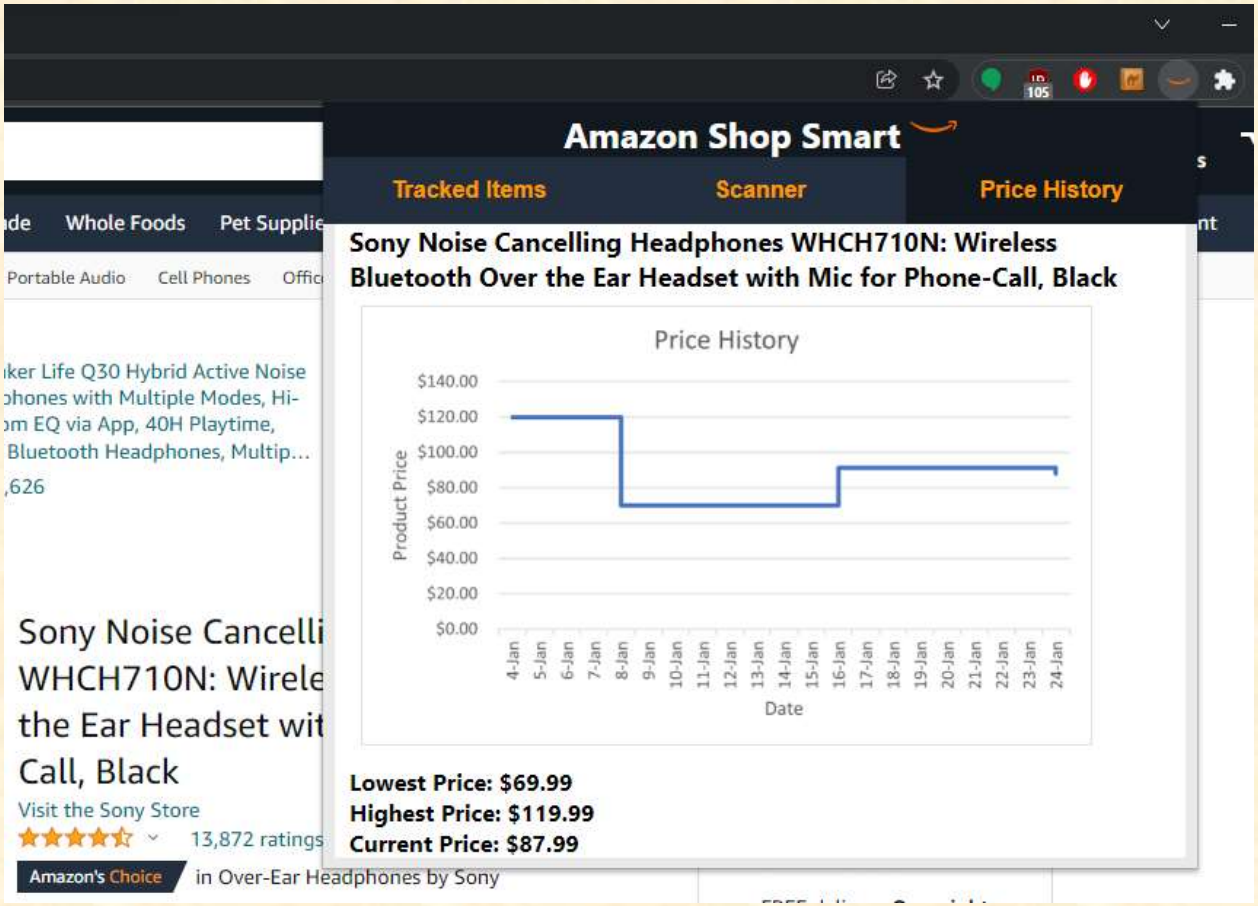




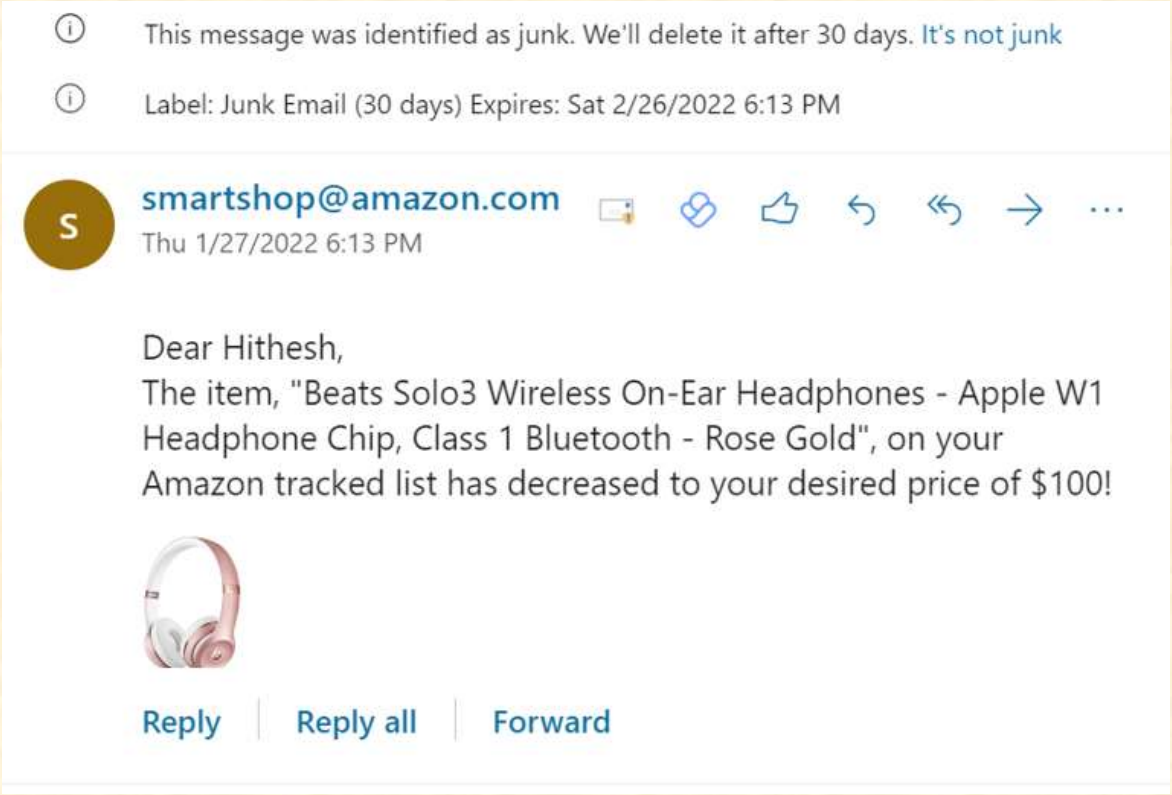
Screen Mockup: Scanner



Screen Mockup: Price History



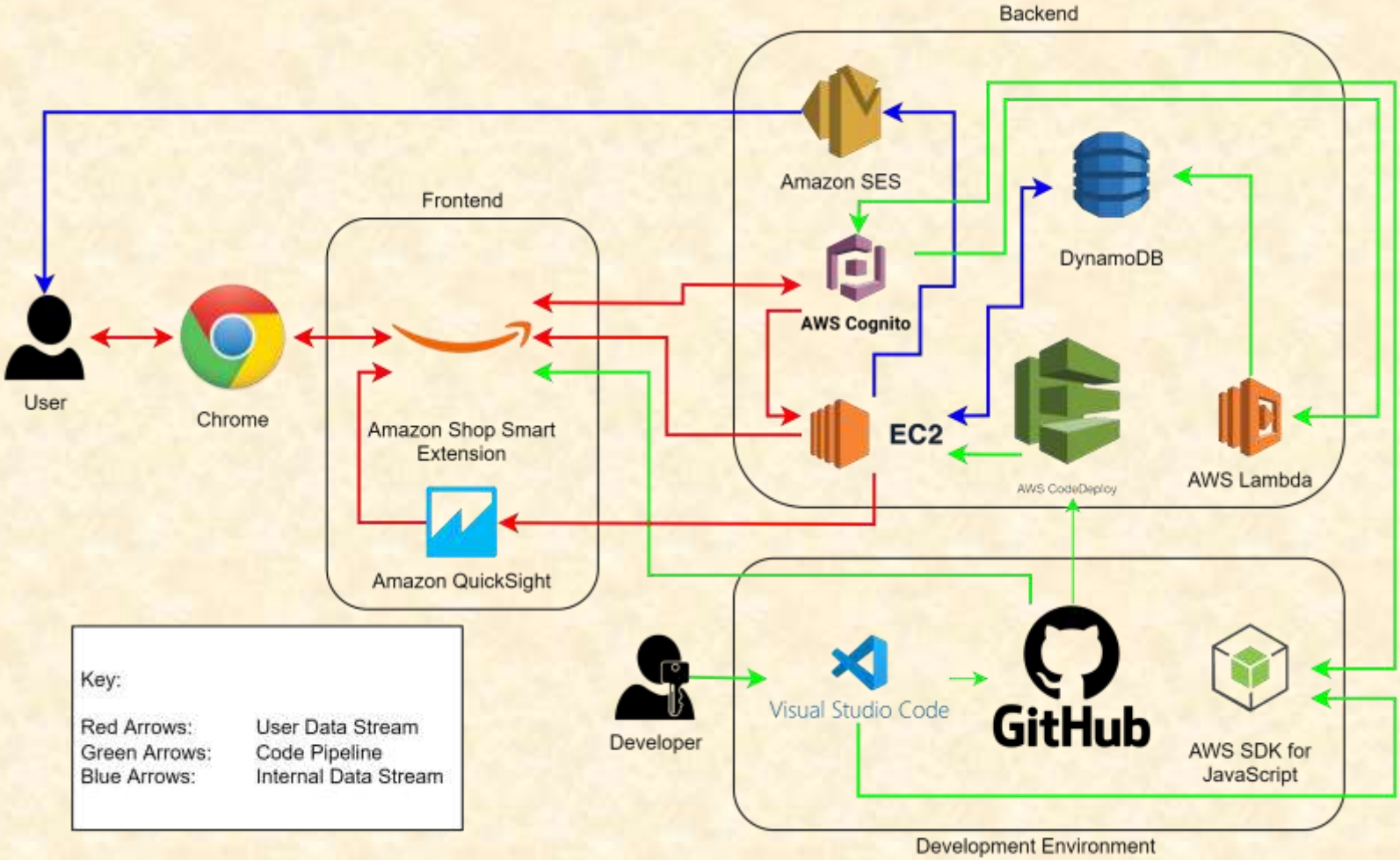
Screen Mockup: Email Notification



Technical Specifications

- Browser extension hosted on Chrome's extension store
- Backend:
  - Processing: Elastic Compute Cloud (EC2)
  - Storage: DynamoDB
  - AWS SDK for JavaScript and RESTful APIs allow frontend interactions with backend services.
- Frontend (UX):
  - Extension interface written with HTML & CSS
  - Data visualization: QuickSight
  - JavaScript listeners for interactivity (e.g. tabs)

System Architecture



System Components

- Software Platforms / Technologies
  - Visual Studio Code
  - Amazon Web Services
    - SDK for JavaScript – simplifies calls to AWS services using JS APIs
    - EC2(Elastic Compound Cloud) - manage storage, launch virtual servers
    - CodeDeploy – can automate code deployments to any instance
    - Cognito – user authentication
    - SES – email notification
    - Lambda – Serverless platform/Runs code in response to events
    - DynamoDB – NoSQL Database
    - QuickSight – Data Visualization platform
  - Chrome Developer

Risks

- Reception of Data
  - Web scrapping for information on different retailers is not cost effective.
  - Use AWS ML tools to limit the scope of each page to just the text. This will help to minimize the amount of data to go through.
- Existing Data
  - No existing database for price history on AWS. No APIs to access information about items. This means we will not have a full database of prices which may cause issues when trying to find trends within the data.
  - Create our own database to store and retrieve information correctly.
- Quantity of Retailers
  - Information may be stored differently across multiple websites. This can begin to cause issues if using the same scrapping approach for different retailers.
  - We will have a few retailer sites where the extension will work. Starting with Target and Best Buy. We will add more as we see fit.

Questions?