MICHIGAN STATE UNIVERSITY

Project Plan Presentation

Remediating AWS Security Gaps Using Generative AI

The Capstone Experience

Team Amazon

Ilyas Abdulrahman
Jaden Cabansag
Ndiaga Diouf
Nate Mikkola
Sardar Murtaza
Valdine Pegy Tchinda Pegou

Department of Computer Science and Engineering
Michigan State University

Fall 2024

From Students…
…to Professionals

Project Sponsor Overview

- Largest global online storefront service
- Amazon Web Services: Amazon's cloud computing platform, offers a wide range of services (e.g. storage, databases computing power)
- AWS is a leader in the cloud industry and serves millions of customers worldwide.

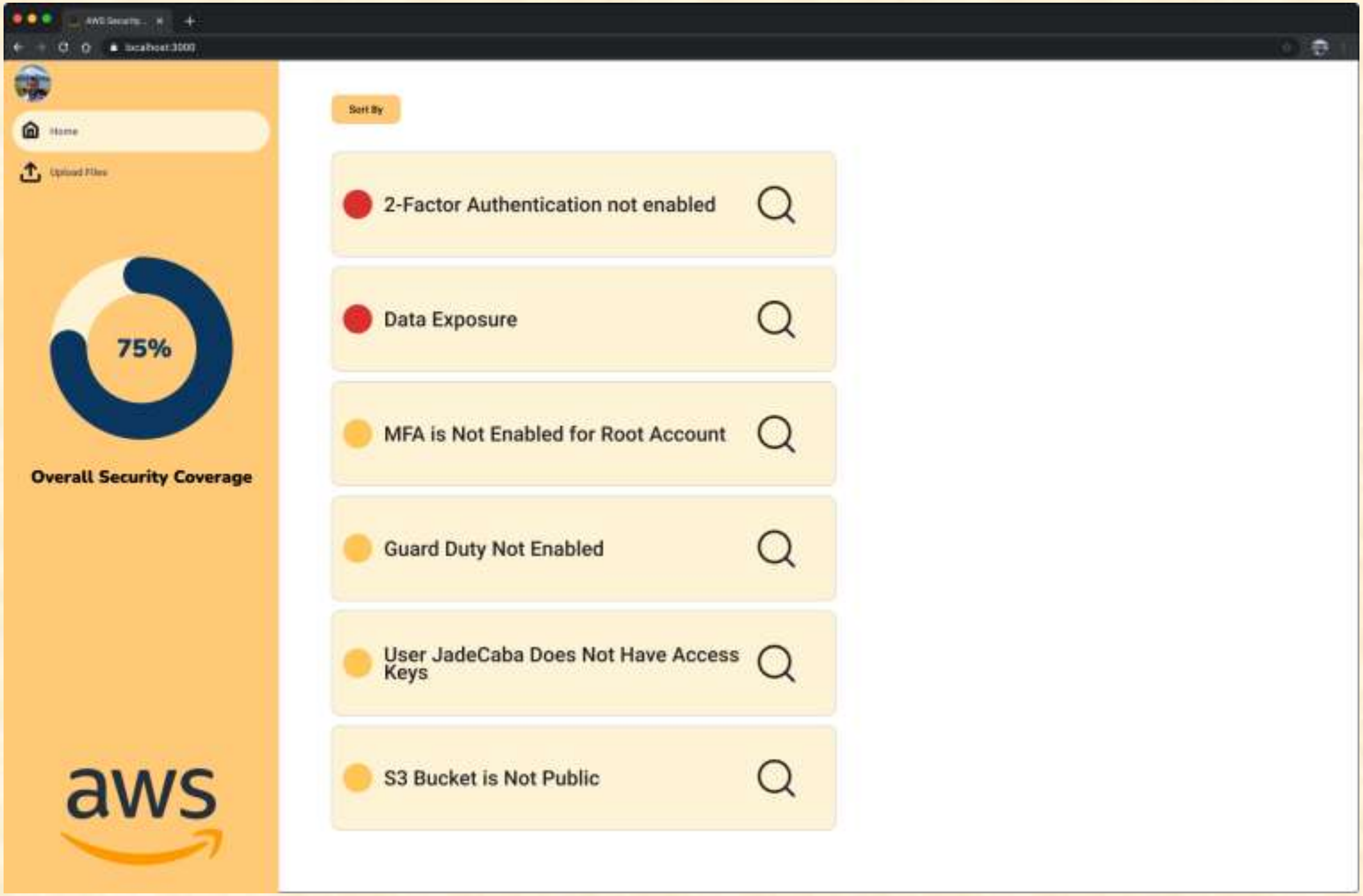
Project Functional Specifications

- AWS account security is highly dependent on a user’s knowledge of the service’s security-based features.
- This can lead to overlooked details or misconfigured features.
  - Misconfiguring an MFA has the potential for the account to be under malicious attack
- Self-Assessment tools provide a scan of an account but can be difficult to read.
- Our application will use AI to provide better and user-friendly insights to users’ account security

Project Design Specifications

- Our client preferred a straightforward interface, which guided our decision to design the app with most of its functionality on a single home page.
- The app is built for both technical and non-technical users, which is why we prioritized an intuitive, easy-to-navigate layout.
- Users can easily sort security findings based on various criteria such as severity, cost to fix, required skill level, time to fix, and an overall weighted score
- The dashboard includes a pie chart that visualizes how far along users are in fixing all security findings which helps users quickly gauge progress on securing their account

Screen Mockup: Home Page



75%

Overall Security Coverage

● 2-Factor Authentication not enabled

● Data Exposure

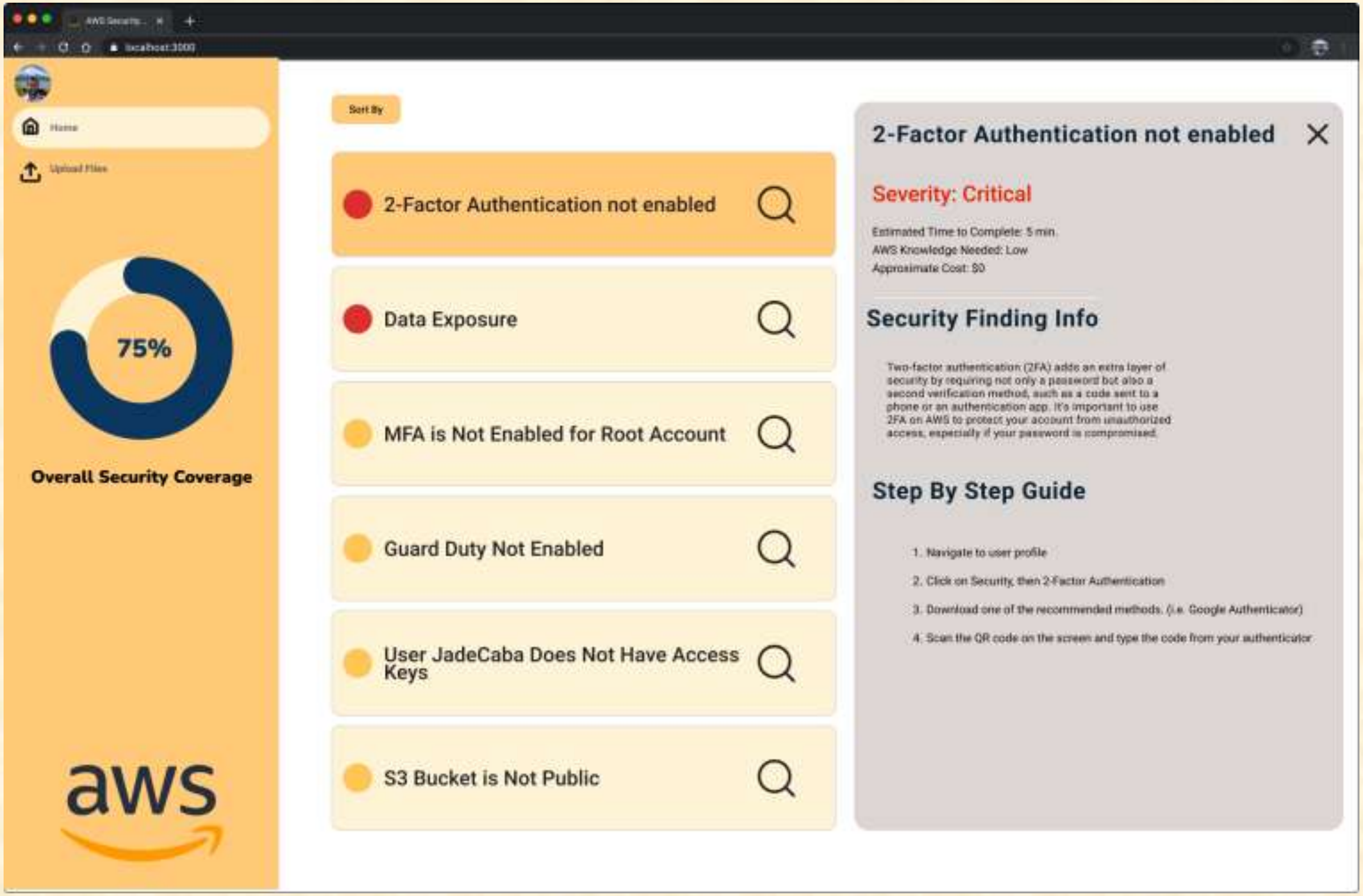
● MFA is Not Enabled for Root Account

● Guard Duty Not Enabled

● User JadeCaba Does Not Have Access Keys

● S3 Bucket is Not Public

Screen Mockup: Home Page (Detailed)



- 2-Factor Authentication not enabled
- Data Exposure
- MFA is Not Enabled for Root Account
- Guard Duty Not Enabled
- User JadeCaba Does Not Have Access Keys
- S3 Bucket is Not Public

2-Factor Authentication not enabled

Severity: Critical

Estimated Time to Complete: 5 min.
AWS Knowledge Needed: Low
Approximate Cost: $0

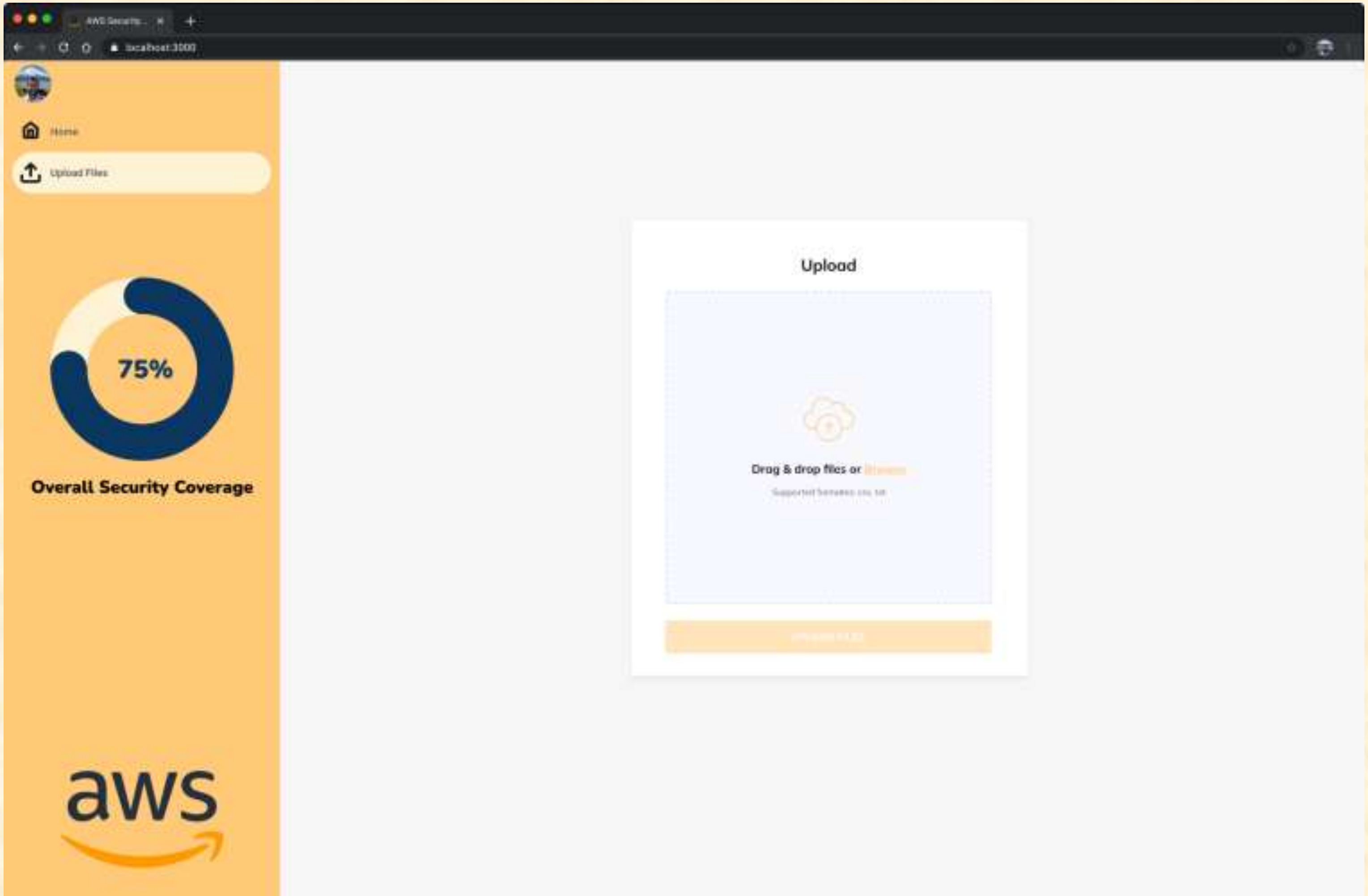
Security Finding Info

Two-factor authentication (2FA) adds an extra layer of security by requiring not only a password but also a second verification method, such as a code sent to a phone or an authentication app. It's important to use 2FA on AWS to protect your account from unauthorized access, especially if your password is compromised.

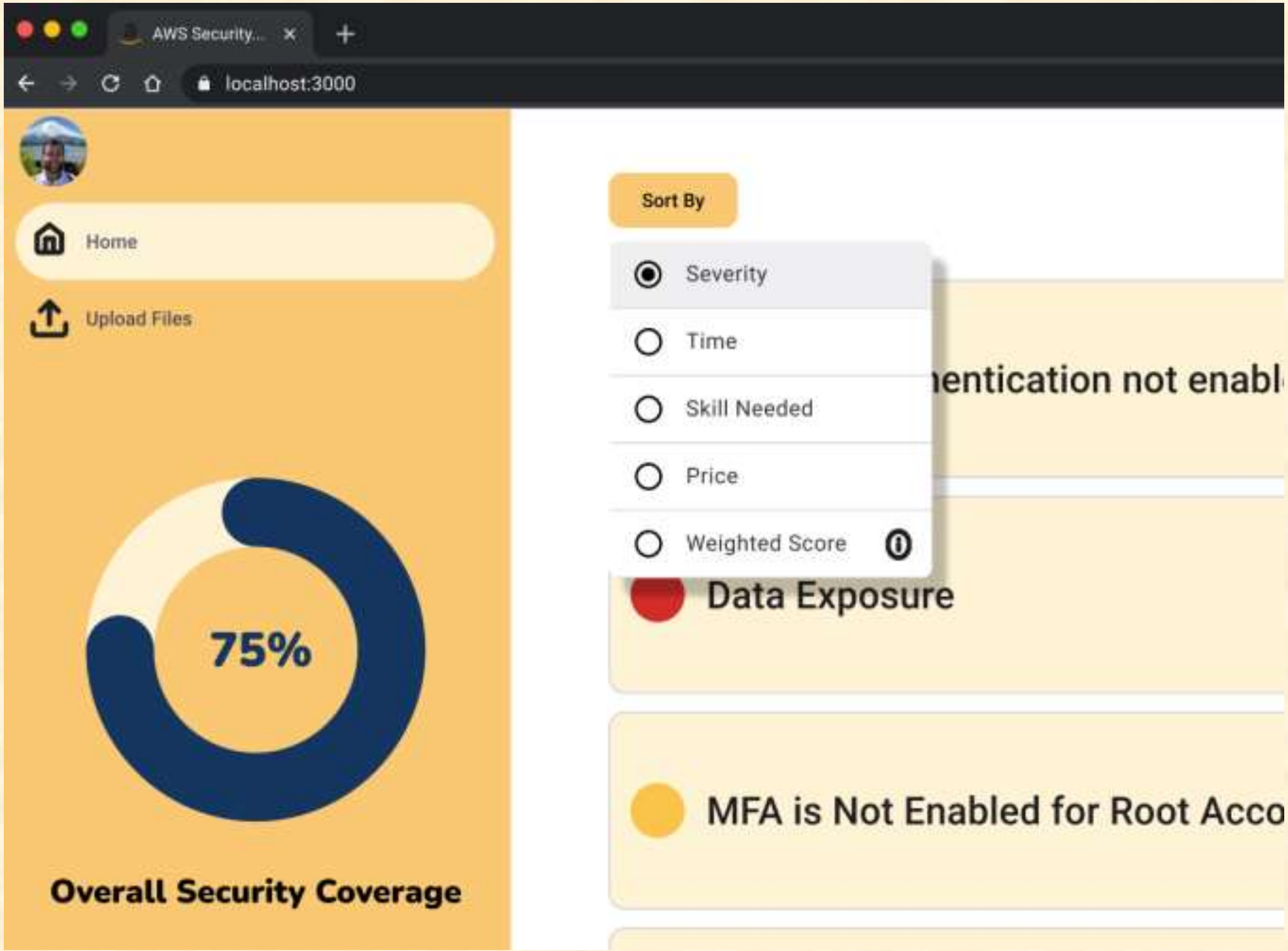
Step By Step Guide

1. Navigate to user profile
2. Click on Security, then 2-Factor Authentication
3. Download one of the recommended methods. (i.e. Google Authenticator)
4. Scan the QR code on the screen and type the code from your authenticator

Screen Mockup: File Upload Page


AWS Security...
localhost:3000
Home
Upload Files
75%
Overall Security Coverage
aws
Upload
Drag & drop files or Browse
Supported formats: csv, txt
UPLOAD FILES

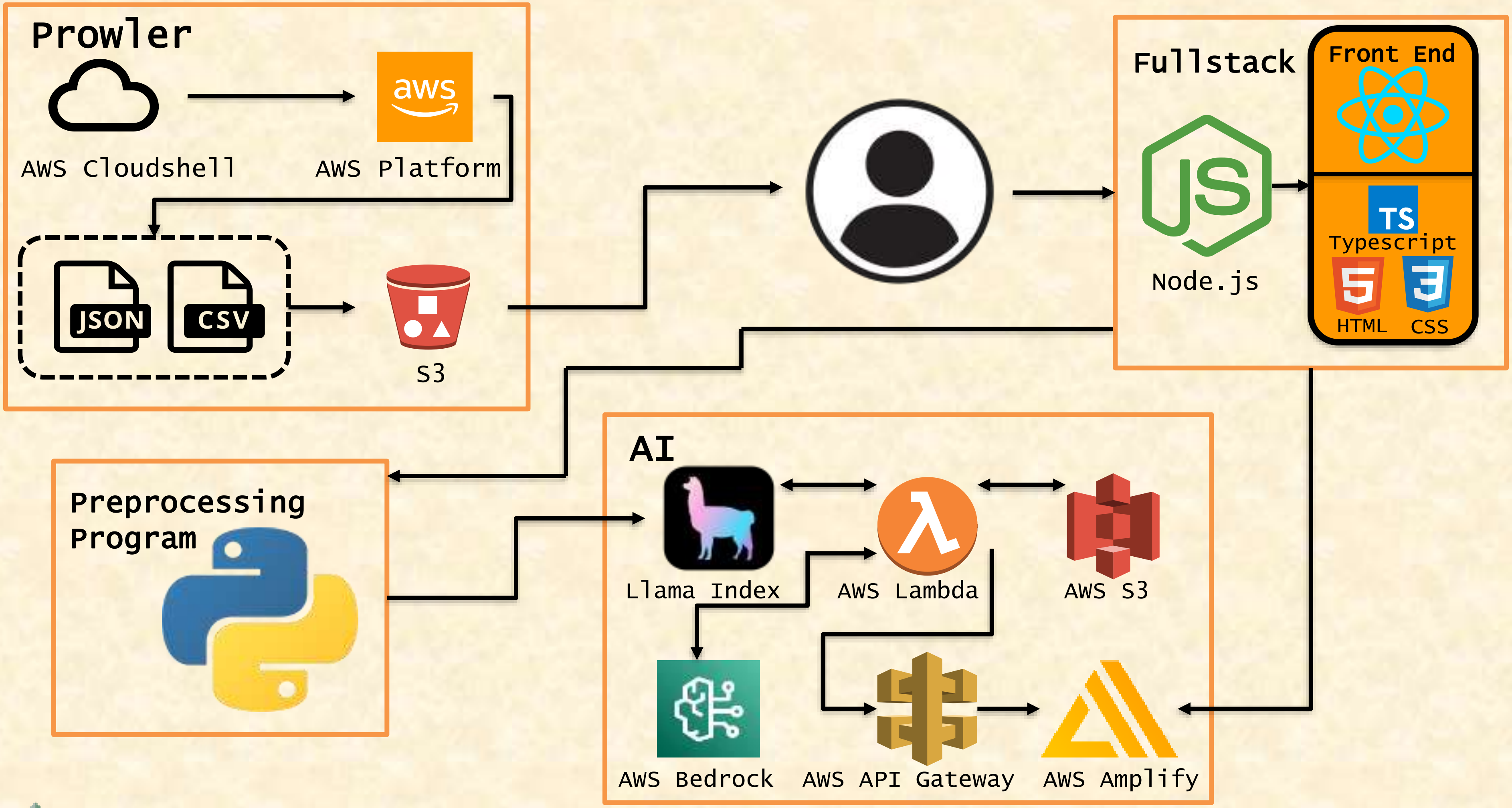
Screen Mockup: Sort By



Project Technical Specifications

- Security Assessment results are generated by Prowler service in an S3 bucket which evaluates an AWS account's configurations for any security gaps.
- Amazon Bedrock is utilized for analyzing the data after being preprocessed by Llama Index
- AWS Lambda manages communication between the preprocessing stage and the AI analysis
- AWS Lambda stores results from AI response to an S3 bucket and acts almost like a cache
- Amplify generates URL for application to host our front end and enable API calls through API Gateway
- AI model will be getting trained to generate the best remediations working backwards from the customer

Project System Architecture



Project System Components

- Hardware Platforms
  - S3 Storage
  - Lambda Execution Environment
  - Amplify Hosting
  - Local development
- Software Platforms / Technologies
  - React
  - AWS Lambda
  - Bedrock
  - API Gateway
  - Amplify

Project Risks

- Risk 1: Shortage of AWS accounts for testing
  - Description: Currently only have 1 AWS account used for testing purposes
  - Mitigation: Refer to public database of security findings to generate test data
- Risk 2: Generative AI Hallucination
  - Description: How will we confidently validate the data our AI responds with?
  - Mitigation: A RAG Model (Restrictive) to completely control an AI's knowledge
- Risk 3: Time needed to preprocess data to be fed to AI model
  - Description: We are unsure of how long it will take in the workflow to preprocess data before we feed it to AWS Bedrock
  - Mitigation: In the case of time taking too long to preprocess this data provided to us by our customer, we can directly feed the data to AWS Bedrock.
- Risk 4: AWS Service Expenses
  - Description: Amazon clients have not finalized a limit on our usage of their AWS Services for our project
  - Mitigation: Track our budget and set up CloudWatch alarms to alert us if we are close to being over our limit.

Questions?

? ? ? ?

? ?

? ? ?