ValueSnap

Project leads: Denis Tzaka & Yathin Vemula
Repository: ValueSnap

Vision

ValueSnap is your Instant Object Valuation App. Discover the true worth of your possessions with ValueSnap, the ultimate app for instant object identification and value estimation. Whether you're a collector, an appraiser, or just curious about the value of items around you, ValueSnap provides accurate and reliable information at your fingertips. Key Features: Instant Identification: Simply snap a picture of any object, and ValueSnap's advanced AI algorithms will identify it in seconds. Accurate Valuation: Get up-to-date value estimates based on market data and trends, ensuring you always know the worth of your items. User-Friendly Interface: Easy-to-use design makes taking pictures and receiving valuations quick and effortless. Comprehensive Database: Access a vast database of items, from antiques and collectibles to everyday objects. Secure and Private: Your data is safe with us. ValueSnap prioritizes user privacy and ensures that your information is securely stored. Transform your smartphone into a powerful tool for valuation with ValueSnap. Download now and start discovering the value of the world around you!

Tech Stack

Front-End:

1. **Mobile App Development**
    - **React Native**: For building a cross-platform mobile app that works on both iOS and Android.
2. **UI Components**
    - **React Native Elements**: For pre-styled UI components.

Back-End:

1. **Server Framework**
    - **Node.js**: For running the server.
    - **Express.js**: For building RESTful APIs.
2. **Database**
    - **MongoDB**: For storing user data and object information.
3. **File Storage**
    - **Amazon S3**: For storing user-uploaded images.

Machine Learning:

1. **Model Development and Deployment**
    - **TensorFlow Lite**: For running pre-trained object detection models on mobile devices.
    - **Flask**: For serving the model if additional server-side processing is needed.

Cloud Infrastructure:

1. **Cloud Provider**
    - **Amazon Web Services (AWS)**: For cloud infrastructure.
2. **Services**
    - **Amazon S3**: For storing images.
    - **AWS Lambda**: For serverless functions to handle image processing if needed.
    - **Amazon EC2**: For scalable compute capacity if required.

DevOps and CI/CD:

1. **Version Control**
    - **Git**: For version control.
    - **GitHub**: For repository hosting and collaboration.
2. **CI/CD**
    - **GitHub Actions**: For continuous integration and deployment.

Monitoring and Analytics:

1. **Monitoring**
    - **AWS CloudWatch**: For monitoring AWS resources and applications.
2. **Analytics**
    - **Google Analytics**: For tracking user interactions and behavior within the app.

Security:

1. **Data Security**
    - **SSL/TLS**: For encrypting data in transit.
2. **API Security**
    - **JWT (JSON Web Tokens)**: For secure user authentication and session management.

Goals

- **Object Identification**
  - Accurately identify objects in images taken by users.
- **Value Estimation**
  - Provide reliable and up-to-date value estimates for identified objects.
- **User-Friendly Interface**
  - Ensure the app is easy to navigate and use for all users.
- **High Performance**
  - Maintain fast and efficient image processing and data retrieval.
- **Cross-Platform Compatibility**
  - Develop a mobile app that works seamlessly on both iOS and Android devices.
- **Secure Data Handling**
  - Implement robust security measures to protect user data and privacy.
- **Scalable Architecture**
  - Design the system to handle increasing numbers of users and data efficiently.
- **Continuous Improvement**
  - Regularly update the app with new features, improved algorithms, and user feedback integration.

Milestones

1) **June**
   - Finalize project requirements and scope.
   - Design app wireframes and user flow.
   - Set up development environment and initial project repository.
   - Start development of the front-end using React Native.
   - Implement user authentication and basic UI components.

2) **July**
   - Complete front-end development of core features (image capture, upload).
   - Develop back-end APIs using Node.js and Express.js.
   - Integrate MongoDB for user data and object information storage.
   - Implement AWS S3 for image storage.
   - Begin integration of TensorFlow Lite for on-device object detection.

3) **August**
    - Complete integration of machine learning model for object detection and value estimation.
    - Conduct thorough testing of the app (unit testing, integration testing).
    - Perform beta testing with a select group of users and gather feedback.
    - Refine and optimize the app based on feedback.
    - Prepare for app deployment on the App Store and Google Play Store.