Reverbyte Decode Forgotten Knowledge from Silent Artifacts

Team: Nothing to lose

Team Lead: SIDDHI SAI KOKKULA

Date: June 2025

Project Title: Reverbyte - Decode Forgotten Knowledge from Silent Artifacts

---

Objective:

To develop a cutting-edge AI-powered platform that decodes historical, cultural, and environmental context from real-world objects such as artifacts, tools, architecture, or fossils by combining 3D scanning, computer vision, cultural databases, and AI-generated storytelling.

Problem Statement:

Millions of artifacts, tools, and natural objects remain under-documented in museums, excavation sites, and even households. These items carry invaluable untold stories about human culture, ancient technology, environmental patterns, and lost civilizations. There is no system today that brings them to life through context-rich storytelling using AI.

Key Features:

1. **Object Scanning & 3D Modeling**

   - Smartphone or AR device-based scanning to generate object data (LIDAR/camera).

   - Extracts dimensions, textures, and visible wear.

2. **Computer Vision Analysis**

   - Uses pre-trained models (YOLOv8, CLIP) to identify visual cues.

   - Analyzes material type, possible function, region-specific designs.

3. **Environmental & Historical Context Layer**

   - Pulls data from satellite (Google Earth API), local weather, soil info, regional databases.

   - Matches against historical databases (e.g., Wikidata, ancient trade maps).

4. **Story Synthesis Engine**

   - AI (via GPT-4 + LangChain logic) generates:

     - Probable object origin, usage, significance.

     - Hypothetical day-in-the-life or event-based reconstructions.

     - Immersive storytelling in text, audio, or AR.

5. **Crowdsourced Validation**

   - Users, historians, or archaeologists can upvote, correct, or add to stories.

   - Builds trust and accuracy score for each artifact.

6. **AR Visualizations (Optional)**

   - Augmented Reality scenes showing object use in past environments.

   - Reconstructed architecture or cultural practices.

Tech Stack:

- **Frontend:** Flutter or React Native
- **3D & AR:** Unity + ARKit / Polycam SDK / Luma AI
- **AI & Vision:** YOLOv8, CLIP, GPT-4 Vision, LangChain
- **Database:** Firebase / Firestore + Neo4j (graph database for history links)
- **Cloud:** Google Cloud or AWS for heavy inference tasks
- **External APIs:** Wikidata, Google Maps, OpenAI, Earth Engine

---

Use Cases:

- Museums providing live interactive experiences
- Archaeologists in the field identifying found objects
- Educators bringing lost history into classrooms
- Tourists interacting with local cultural heritage in AR

Impact Potential:

- Revives forgotten or undocumented heritage
- Empowers education and awareness through immersive learning
- Provides new avenues for crowd-sourced historical validation

- Bridges gaps between AI and anthropology

Team Roles (Hackathon Setup):

- **ML/AI Developer:** Vision model integration and GPT fine-tuning

- **Mobile Developer:** UI for scanning and experience layer

- **Backend Developer:** API integration, database design

- **Historian/Researcher (Mentor Role):** Source validation and narrative input

- **AR/3D Specialist:** Scene building and rendering for historical reconstructions

---

Next Steps:

1. Develop MVP that accepts object photo + gives predicted purpose

2. Integrate GPT-4 Vision + storytelling output

3. Add 3D scanning and match with historical dataset

4. Implement user validation and feedback

5. Test on real museum artifacts or household items

Stretch Goals:

- Cross-lingual storytelling for global accessibility

- Time-travel AR portal for key historical events

- Integration with VR headsets for museums

Tagline:

"Reverbyte: Breathing life into silent artifacts."

Prepared For:

World's Largest Hackathon Competition

Prepared By:

Nothing to lose (Team Lead: SIDDHI SAI KOKKULA)

Date: June 2025

Implementation Steps & Sample Code

---

1. Setup Environment

- Install dependencies:

```bash

pip install opencv-python torch torchvision transformers langchain firebase-admin

```

---

2. Object Detection (YOLOv8 + CLIP Integration)

```python
import cv2
from transformers import CLIPProcessor, CLIPModel
import torch

model = CLIPModel.from_pretrained("openai/clip-vit-base-patch32")
processor = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32")

image = cv2.imread("artifact.jpg")
inputs = processor(images=image, return_tensors="pt")
outputs = model.get_image_features(**inputs)
```

---

3. GPT-4 Vision for Storytelling

```python
from openai import OpenAI

response = openai.ChatCompletion.create(
    model="gpt-4-vision-preview",
    messages=[
        {"role": "system", "content": "You are an AI historian."},
        {"role": "user", "content": "Describe the history and possible use of this artifact."}
    ],
    temperature=0.7
)
print(response['choices'][0]['message']['content'])
```

---

4. Firebase Integration

```python
import firebase_admin
from firebase_admin import credentials, firestore

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred)
db = firestore.client()
doc_ref = db.collection("artifacts").add({"description": "Generated story", "tags": ["metal", "weapon"]})
```

---

5. AR Rendering (Unity with Polycam/Luma AI Assets)

- Import scanned 3D models into Unity.

- Use ARKit/ARCore SDK to render models in real-world space.

Final Notes

- Each module should be modularized and independently testable.

- Use a simple React Native or Flutter frontend for user interaction and camera access.