

Apple Vision Pro Team Member Contributions

Sam Kwon:

- Researched Hugging Face models and AWS technology
- Setup AWS cloud platform (initially attempted to use AWS Lambda)
- Wrote backend WebSocket server code using FastAPI and Transformers library
- Attempted to install dependencies on AWS Lambda using multiple measures (Docker, S3 bucket, etc.)
- Concluded that we needed to move to EC2 due to resource constraints and persistency issues on Lambda
- Setup and configured AWS EC2 instance
- Ported code to EC2 through secure copy protocol
- Installed all dependencies in EC2
- Got WebSocket server up and running on EC2 instance
- Hit server with Postman to validate that it was working
- Worked on unit tests for EC2 backend using pytest
- Created installation guide for backend instructing users on how to get server setup and running
- Updated documentation website with general modifications (grammar, formatting, etc.)
- Worked on various Sprint deliverables
- Assisted in lab meetings for development, testing, and demo recordings
- Met with sponsor to confirm direction and get feedback
- Met with cyber team to understand and discuss cybersecurity issues

Trang Do

- Created a visionOS app to allow users to pick an image from their photo library
- Converted the selected image to Base64 to send it to AWS EC2 via WebSocket
- Parsed the response from the backend and displayed the result as a table view to users
- Created a splash screen for the visionOS app after the app starts
- Updated the project website using NextJS framework and Framer Motion to add animations. Also used Tailwind CSS to make it more appealing, interactive and user-friendly
- Refactored the visionOS code to follow Model-View-ViewModel (MVVM) architecture
- Displayed the Voice Recognition View based on Navigation Stack
- Set up the documentation website from Read the Docs Tutorial

- Worked on the documentation website section for how to install software
- Did many screen recordings for how to install the visionOS app, to start up EC2 instance, to start the server, to use all the features on the app, and to run test cases
- Updated the project website with the software features, sprint deliverables for all sprints, and demo video
- Worked on deliverables and assisted in lab meetings
- Met with cyber team weekly to ensure security

Anthony Rutherford:

- Researched Hugging Face models and AWS technology
- Setup AWS Lambda platform (eventually had to switch to EC2)
- Installed dependencies on AWS Lambda using Layers which with further testing ensured that this project was going to be too large to complete with the free version of AWS Lambda
- Communicated with sponsor to seek preference in projects next step (we decided on EC2 as it would allow us to avoid cold start times by using a physical server.
- Implemented cyber security recommendations into server code
- Ported test versions to EC2 to ensure new versions would work
- Hit EC2 instance with Postman to validate test versions
- Worked on unit tests for EC2 cyber-security backend
- Assisted in editing installation guide for backend, updated documentation website
- Worked on deliverables and assisted in lab meetings where we ran tests and recorded demos
- Communicated with sponsor to get feedback as well ensuring our vision for the project matched his expectations
- Communicated and met with cyber-team to ensure our application held a balance of usability and security

Raudel Armenta:

- Created basic capstone website in sprint 1
- Researched speech to text for visionOS
- Created and stored API key into a scheme
- Created speech to text so user can ask OpenAI questions
- Worked on debugging a VoiceRecognitionView where response was not consistent
- Worked on deliverables and assisted in lab meetings where we ran tests and recorded demos
- Communicated with sponsor to get feedback as well ensuring our vision for the project matched his expectations

- Communicated and met with cyber-team to ensure our application held a balance of usability and security
- Worked on Extensions/Modifications for documentation website

Donovan Kohler:

- Researched Hugging Face models and different potential cloud platforms to host the backend code.
- Researched FastAPI and created a local Uvicorn server to run backend processing code.
- Tested the local server with different Hugging Face models.
- Decided on AWS as hosting platform and researched the different options it provided in terms of resources and pricing.
- Initially chose AWS Lambda based on research and troubleshooted AWS Lambda dependencies for Python.
- Determined that AWS Lambda did not meet our needs and switched to EC2 for adequate resources and ease of use.
- Implemented error handling to test the backend using a mock client, ensuring invalid data was handled correctly.
- Verified successful WebSocket connections and model processing.
- Added testing instructions to the GitHub README.
- Added documentation on how to use different product functions on the documentation website.
- Worked on deliverables and attended lab meetings for product testing, troubleshooting, and demo recordings.
- Met with cyber team to ensure our product met security requirements.