



Presented by Arjan

Mission 1: Turner's Car Insurance Project

Role: Advanced Full Stack Developer

Client: Turner's Car Auctions

Goal: Prototype an AI solution that identifies vehicle types from uploaded car images to support accurate insurance pricing.

My Focus:

- Understand project context
- Identify key stakeholders
- Build and present an AI-powered web app

Task 1: Identify Stakeholders

01

Purpose: Understand who is involved and what they care about.

Key Stakeholders

- **Digital Transformation Manager** – Oversees the system redesign and ensures alignment with company goals.
- **IT / Development Team** – Builds and maintains the AI prototype.
- **Insurance Department** – Uses AI results to help calculate premiums.
- **Marketing & Sales Team** – Promotes new features to customers.
- **End Customers** – Upload car images for quotes (main users).
- **Compliance / Legal Team** – Ensures data privacy and insurance compliance.
- **Project Manager / Product Owner** – Manages tasks, communication, and delivery.

Task 2: Understand Project Context

Purpose: Identify what to know before starting development.

Top 5 Things to Know

- Project Goals & Success Metrics** – What are we improving and how success will be measured.
- Technical Requirements** – Confirming tools and tech stack (e.g., Azure, React).
- Data & Privacy** – Understanding available datasets and compliance needs.
- Target Users** – Who will use the app and what experience they expect.
- Timeline & Process** – Deadlines, Jira workflow, and collaboration plan.

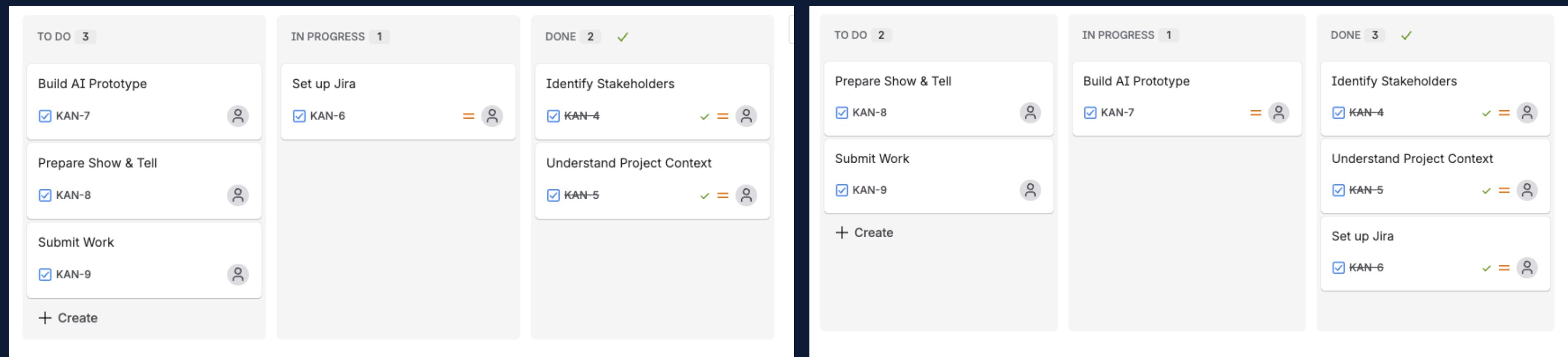
Task 3.- Jira Board

Task 4 - Prototype

Final Kanban and Thought's

RESEARCH PAPERS

Task 3 — Jira Board



Task 4 - Prototype

M1-turnersCarVision Computer vision

Search Delete

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Resource Management

- Keys and Endpoint
- Pricing tier
- Networking
- Identity
- Cost analysis
- Properties

Security

Monitoring

Automation

Essentials

Resource group (move) M1-Turners-api-project

Status Active

Location Australia East

Subscription (move) Azure subscription 1

Subscription ID a9634ccf-4182-4da1-b22d-5d4b9171f993

Tags (edit) Add tags

API Kind ComputerVision

Pricing tier Free

Endpoint https://m1-turnerscarvision.cognitiveservices.azure.com/

Manage keys Click here to manage keys

Get Started

Get started with your resource in Vision Studio

Try out all Computer Vision features and build your own custom models

Go to Vision Studio

JSON View

Turner's AI Vehicle Type Detector

Choose file No file chosen

Upload Car Image

Prototype Test

This is a preview of the fully functional website.

Turners AI Vehicle Type Detector

Upload a Car Image

UPLOAD CAR IMAGE



Detected Vehicle Type: SPORTS SEDAN

Thank you

A screenshot of a Kanban project management interface titled "My Kanban Project". The interface shows three columns: "TO DO", "IN PROGRESS", and "DONE".

- TO DO:** Contains one item: "Submit Work" (KAN-9).
- IN PROGRESS:** Contains one item: "Identify Stakeholders" (KAN-4).
- DONE:** Contains five items: "Understand Project Context" (KAN-5), "Set up Jira" (KAN-6), "Build AI Prototype" (KAN-7), and "Prepare Show & Tell" (KAN-8).

A screenshot of a code editor showing a file named "App.jsx". The code is a React component named "App" that handles file uploads to an Azure endpoint.

```
mission-1 > src > App.jsx > App > handleUpload
1 import { useState } from "react";
2 import "./App.css";
3
4 function App() {
5   const [image, setImage] = useState(null);
6   const [carType, setCarType] = useState("");
7   const [loading, setLoading] = useState(false);
8   const [error, setError] = useState("");
9
10  const endpoint = import.meta.env.VITE_AZURE_ENDPOINT;
11  const key = import.meta.env.VITE_AZURE_KEY;
12
13  const carTypes = ["suv", "sedan", "truck"];
14
15  const handleUpload = async (event) => {
16    const file = event.target.files[0];
17    if (!file) return;
18
19    setImage(URL.createObjectURL(file));
20    setCarType("");
21    setError("");
22    setLoading(true);
23
24    try {
25      const response = await fetch(`${endpoint}/vision/v3.2/tag`, {
26        method: "POST",
27        headers: {
28          "Ocp-Apim-Subscription-Key": key,
29          "Content-Type": "application/octet-stream",
30        },
31      });
32
33      const data = await response.json();
34      console.log(data);
35    } catch (err) {
36      setError(`Error: ${err.message}`);
37    }
38  };
39
40  return (
41    <div>
42      <h1>Welcome to AI Project</h1>
43      <img alt="Image placeholder" src={image} />
44      <p>Selected Car Type: {carType}</p>
45      <p>Is Loading: {loading}</p>
46      <p>Error: {error}</p>
47      <input type="file" onChange={handleUpload} />
48    </div>
49  );
50}
```