Globant Piscine

Project 4

Summary: ServiceDeskai

Version: 1

Contents

Chapter I

A word about this Project

Develop a mobile-first application that allows reporting problems and damaged items in Glober offices or homes using geolocation, machine learning, and other technologies to make the experience as simple as possible.

Chapter II

Introduction

What this Project will show you:

- Developing a mobile-first application for reporting problems and damaged items.
- Using geolocation and machine learning to streamline reporting.
- Implementing user authentication with JWT for secure API access.
- Designing an app that meets WCAG AA accessibility standards.
- Managing multiple user roles: standard users, service desk, and admin.
- Creating reports with personal data, location, and media attachments.
- Sharing reports via email and viewing report history by status.
- Adding features like chat, admin controls, offline mode, and dark mode.
- Enhancing the app with image recognition and AI-driven intelligence using APIs like Google Cloud Vision.

Chapter III

General instructions

Unless explicitely specified, the following rules will apply for every project of this Piscine.

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- Your assignments WON'T be evaluated by your Piscine peers.
- You must not leave in your turn-in your workspace any file other than the ones explicitly requested By the assignments. If the assignment don't precise them, put only the necessary ones to run your Project.
- Using some API Key or Token? Keep them for you! Do not push them on your repository.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the `man` or on the Internet.
- You must read the examples thoroughly. They can reveal requirements that are not obvious in the assignment's description.
- By Thor, by Odin! Use your brain!!!

Chapter IV

Mandatory part

Globant	Exercise 00
ServiceDeskai	
Turn-in directory : $ex00/$	
Files to turn in : `All needed files to run your Project and nothing else`	
Allowed functions : `None`	

The goal is to add as much intelligence as possible to the reporting of damage or failure. Geolocation, object recognition, tagging in the images and the report. The smarter the final solution, the better. Our recommendation is to use an API like Google Cloud Vision or Azure AI Vision.

- **The app must support three user profiles**
  - **Standard user**: Can submit new tickets, upload photos/videos, and track the status of reports.
  - **Service desk user**: Can receive tickets, update their status, close them, and open chats with standard users.
  - **Admin user**: Can create new users and offices, view all relevant information, and access reports.
- **Technology Stack (JavaScript)**
  - **Back End**: Node.js + Express + MongoDB.
  - **Front End**: React + Redux.
- **Must-Have Features**
  - User login with JWT token management for API validation.
  - Mobile-first design, optimized for mobile and desktop.
  - Ensure WCAG AA accessibility compliance.

- Profile management: Personal data, preferences, and preferred office/workstation for easier reporting.
- Report submission: Include necessary personal data, office/workstation details, media (images/videos), and incident geolocation.
- Report sharing via email (SHARE).
- Report history: Separate views for open and closed tickets, with status tracking (open, assigned, in-progress, closed).

You may want to take a look at every technologies referenced in the project description before starting.

This is a group project, remember to communicate with your team members and split the work together. Git Branch are your friends. Use them to split the work and merge it together.

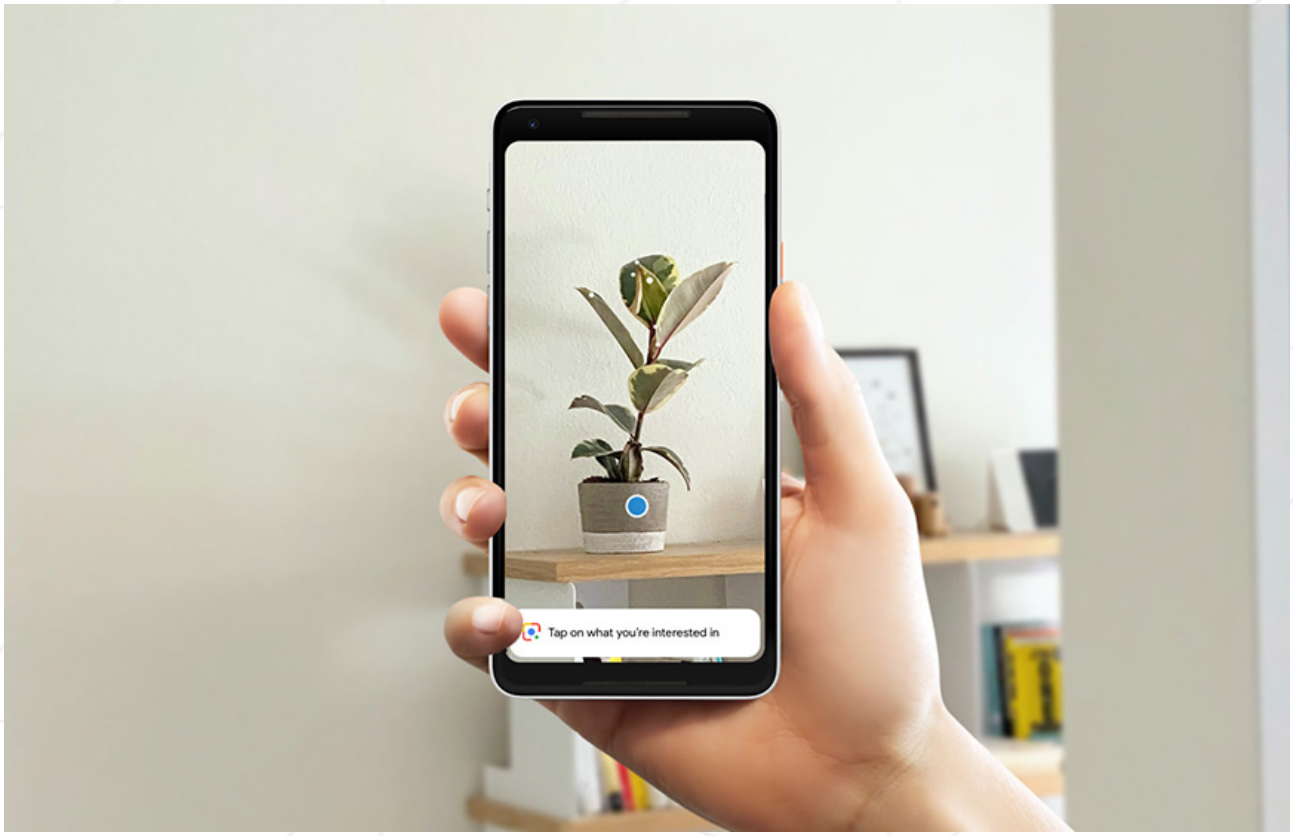

Figure IV.1: Example with Google Lens

Chapter V

Bonus Part

Once mandatory requirements are implemented in the application, you can add any additional features you see fit. Here a quick list of suggestions to give you ideas.

- In-app chat within tickets for communication between users and service desk staff.
- Admin module for managing tickets, generating reports, and overseeing users and offices.
- Profile photo upload via file or camera.
- Offline mode (PWA).
- Dark mode for better user experience.

Those are only examples. The sky's the limit so again, once mandatory functional requirements are met, we are open to any additional functionalities you see fit.

Chapter VI

Submission

- Create a git repo (Github, Gitlab, Bitbucket, etc) and add your project files to it.
- Copy the link to your repository and paste it in the project submission form.
- Project submission form: TYPEFORM

Please note, no modifications made on the repo after the form is sent will be taken into account for the evaluation.

No Peer evaluation for this Piscine, but feel free to share your project with your peers and get feedback.