# Take-Home Assignment Instructions

## Objective

In this take-home project, we aim to assess your:

1. Ability to build a non-trivial application from scratch.
2. Comfort in picking up unfamiliar technologies.
3. Architectural decisions, abstractions, and rigor in your approach.

We respect your time, so please try not to spend more than **8 hours** on this assignment. We understand that this is a challenging task under time constraints, and we will take that into account when evaluating your solution.

## Problem Statement

You will build a **mini patent infringement check app**.

### What is Patent Infringement?

Patent infringement occurs when a party makes, uses, sells, or offers to sell a patented invention without permission from the patent holder. In essence, it means violating the exclusive rights granted to the patent owner, typically leading to legal disputes.

## Core Functionality

The app will allow users to:

- Input a **patent ID** and a **company name**.
- Run a patent infringement check against the specified company.
- Return the **top two infringing products** of the company along with explanations of why these products potentially infringe the patent, specifically detailing which claims are at issue.

The **patent ID** and its corresponding **patent claims** are provided in the `patents.json` file. Additionally, you will receive a list of companies and their corresponding products and summaries in the `company_products.json` file.

To determine relevance and potential infringement, you may use any LLM model (e.g. GPT, Perplexity, etc).

# Technical Requirements

- Please include a simple **README.md** file detailing how to run the app.
- You need to **dockerize** the application (avoid Port 5000) so that we do not need to install additional development dependencies or set up the environment manually.
- We ask you to use Python for Backend and Typescript for Frontend, but you are free to use any frameworks or libraries as you see fit.
- **Bonus Points**: You will earn bonus points if you can host and run your app on a web server.
- **Bonus Points**: Allow basic user input parsing so you can do some fuzzy match with the given dataset.
- **Bonus Points**: Once the results are generated, users will have the option to save the results as a report and later return to the app to view all previously saved reports.

# Example

## User Input

**Patent ID**: US-RE49889-E1

**Company Name**: Walmart

## Infringement Analysis

This is an example output format that includes all the basic details. However, you can structure your data in any way you prefer. Just ensure that when you display the infringement report on the UI, it is clear and easy to understand.


```
{
  "analysis_id": "1",
  "patent_id": "US-RE49889-E1",
  "company_name": "Walmart Inc.",
  "analysis_date": "2024-10-31",
  "top_infringing_products": [
    {
      "product_name": "Walmart Shopping App",
      "infringement_likelihood": "High",
      "relevant_claims": ["1", "2", "3", "20", "21"],
      "explanation": "The Walmart Shopping App implements several key elements of the pat
      "specific_features": [
        "Direct advertisement-to-list functionality",
        "Mobile app integration",
        "Shopping list synchronization",
```

```
          "Digital weekly ads integration",
          "Product data payload handling"
        ]
      },
      {
        "product_name": "Walmart+",
        "infringement_likelihood": "Moderate",
        "relevant_claims": ["1", "40", "41", "42"],
        "explanation": "The Walmart+ membership program includes shopping list features tha
        "specific_features": [
          "Shopping list synchronization across devices",
          "Deep linking to product lists",
          "Advertisement integration in member benefits",
          "Cloud-based list storage"
        ]
      }
    ],
    "overall_risk_assessment": "High risk of infringement due to implementation of core pat
}
```


## Assumptions

- This app obviously doesn't need to be production-ready, but you should at least be aware of any issues you may encounter in more real-world scenarios.
- Bonus points are nice to have, but not required. Don't feel pressured to implement these bonus features if you're running short on time or would prefer to focus on polishing the core functionality.
- While clean code, proper documentation, and software engineering best practices are crucial in production environments, we understand that this is a time-constrained assignment. We don't expect the same level of polish or documentation that you would provide in a production codebase. Focus on implementing the core functionality and making the code readable and maintainable within reasonable bounds for a take-home assignment.

## Help and Clarifications

- If you are not sure, feel free to make fair assumptions and document them in README.md. If you need clarification on any aspect of the problem statement while working on this assignment, please feel free to email us at **eng-takehome@patlytics.com**. We will respond as quickly as possible to assist you.
- You are encouraged to use any resources available on the Internet to help you tackle this challenge: guides, technical documentation, sample projects on GitHub—anything is fair game!

We want to see how you can build solutions in a real-world environment where no information is off-limits.

## Submission Instructions

When you've finished, please send back your results to eng-takehome@patlytics.com and CC our recruiting lead Sara (sara@patlytics.com) via email as a zip file. Make sure to include any instructions about how to run the app in the README.md.

## Thank You

As a thank you for spending time on this assignment, we will send you a $50 (or $1,500 TWD) gift card once you complete the project with all the requirements and have a working end-to-end solution, regardless of whether we move you forward in the hiring process.

We kindly ask that you do not share this assignment with anyone else to ensure fairness in the evaluation process.