|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Ashok Regar | | |
| **Email** | Regar.ashok7@gmail.com | | |
| **Current Role** | Application Development Team Lead | **Number of Years/Months in the role** | 2 Years |
| **Current Responsibilities** | * Develop new features using React Js and MaterialUI for micro services application web app. * Follow Agile development, attend scrum meetings and finish task assigned in sprint of two weeks. * Coordinate and guide fellow teammates to deliver fast. | | |

|  |
| --- |
| **Please describe technical competences you specialize in.**  **Note: use N/A if no experience.** |
| **Front-end Technologies: React, TypeScript, Redux-Saga, JavaScript, HTML, CSS, MaterialUI, JQuery**  **Back-end Technologies: Node, Express, SQL, TypeORM, GraphQL**  **Databases: PostgrSQL, MongoDB**  **Cloud Platforms: AWS** |

|  |
| --- |
| **What is your exposure to building web applications that leverage AI/ML Models? How is the ML model integrated with the application? Please describe using a project you have worked on.** |
| **I have worked on building ML application for tagging email Deadline email or not. Also worked with technologies involving clustering and classification models.**  **My experience in tagging email as Deadline:**  **Problem Statement: Emails contains phrases for deadline. We some time miss to read those sometimes as we get lot of emails. So, using the Ml model if we can detect and put them separate folder or label them, then user can read those and take on action in time.**  **Approaches:**  **We tried two strategies.**   1. **One is probability Bayesian Theorem in which we train the model by giving phrases weights like for eg Submit on=3, Date regex: 2, all other phrase: 1. And then after user can give feedback on predicted email on false or true to improve the model.** 2. **Second was the RNN model to create tuples of words in train data and see if the prediction is acceptable or not. And tweak layers and RNN model parameters to see if results are improved.**   **We had good results from First approach as we tried it in our gmail account. Accuracy was around 75-80%.** |

|  |
| --- |
| **Design and implement a web application that would allow users to:**  **Functional Requirements**   * **Upload and persist pricing feeds from retail stores using CSV files which contain Store ID, SKU, Product Name, Price, Date** * **Search for pricing records using various criteria and be able to edit/save changes to any record**   **Non-Functional Requirements**   * **Standard set of non-functional requirements you would expect a retail stores chain with 3000 stores across multiple countries**   **Please feel free to choose the technology stack and frameworks you are comfortable with and implement a single page web application.**  **Expected Deliverables:**   * **Context Diagram** * **Solution Architecture** * **Design Decisions** * **Non-functional requirements considered and how the design addresses them** * **Assumptions** * **Source for the implementation**   **Upload the artifacts and source to your Github repository and include a reference to it as part of the response.** |
|  |