The Exercise

1. From this list, rank your 5 most favourite and 5 least favourite support tasks. Provide a brief explanation for each.

Ans: There are two kinds of problems in your life: the urgent and the important. The urgent are not important, and the important are never urgent.

That being said, I would like to go through these tasks one by one and separate them by quadrant. Once I can see these tasks in the designated categories, I will be able to schedule them and accomplish my most important work.

Once you know how to distinguish between designated categories (urgent and important tasks), you can begin separating your tasks into the four quadrants.

Quadrant 1: Do

Tasks in this quadrant are both urgent and important. They demand immediate attention as they have significant consequences and impact long-term goals. These are the tasks that are most pressing and likely causing you stress.

Quadrant 2: Schedule

This quadrant is for tasks that are important but not urgent. They contribute to long-term goals but don't require immediate action, allowing you to schedule them for later. Prioritize these tasks after completing those in Quadrant 1, employing strategies like the Pareto principle or the Pomodoro technique for effective time management.

Quadrant 3: Delegate

Quadrant three houses tasks that are urgent but not important to your long-term goals. Since these don't necessitate your personal attention or a specific skill set, delegate them to others. This approach not only lightens your workload but also aids in team development.

Quadrant 4: Delete

After sorting tasks into the first three quadrants, what remains are neither urgent nor important. These tasks are distractions hindering your progress and should be eliminated, fitting into the "delete" quadrant.

These are the common support activities that one has to do at a certain point of time. I would like to rather organize the tasks by frequency also– Daily, Weekly, Biweekly, and Monthly:

Daily Tasks

- Respond to 50+ support requests via email: Addressing customer queries to ensure prompt support.
- Dig through logs to troubleshoot a customer's broken project: With a focus on automating repetitive patterns.
- Help resolve billing issues for customers: Ensuring smooth financial transactions for user satisfaction.
- Manage a support team: Leading and guiding the support team on a daily basis.
- Work with 3rd party partners to track down tricky situations: Collaborating to solve complex customer issues.
- Act as a dedicated CSE for key customers: Providing specialized support to ensure their success.
- Work with engineering teams during incidents: Coordinating to resolve technical issues promptly.
- Work with people to figure out if the product is suitable for their use case: Guiding potential users daily.

Weekly Tasks

- Write and maintain support articles and docs pages: Keeping documentation updated to help users self-serve.
- Create video tutorials for specific features or use cases: Enhancing learning resources for users.
- Identify, file (and, where possible, resolve) bugs: Keeping the product quality high by managing its issues.
- Find and recruit teammates for the support team: Strengthening the team with new talent.
- Help train and onboard new support teammates: Ensuring new members are well-prepared.
- Respond to queries on Twitter, Reddit, Hacker News, and other 3rd party sites: Engaging with the user community.
- Scheduling time-off coverage: Planning to maintain team efficiency and support coverage.

Biweekly Tasks

- Run ticket review sessions to ensure tone consistency: Maintaining quality and uniformity in customer interactions.

- Work with the product team on new feature development based on feedback: Collaborating for product improvements.

Monthly Tasks

- Analyze hundreds of support tickets to spot trends: Identifying larger trends to inform product development.
- Engage multiple users in public discussions: Managing broader community engagement and support.

2. What do you want to learn or do more of at work?
Ans.
Learning is indeed a never-ending process, and I wholeheartedly embrace it as a fundamental part of my professional journey. I live by the philosophy that 'People will forget what you said, people will forget what you did, but people will never forget how you made them feel.' This principle not only shapes my interactions with others but also influences my approach to customer service and team leadership.

Guided by the philosophy that people will never forget how you made them feel, I place the customer at the center of everything I do. I firmly believe in aligning my actions with what's best for our customers, convinced that their success directly translates to our success. This approach has shaped my decision-making process, often leading me to opt for solutions that provide long-term value over immediate, short-term gains.

What I have done in my leadership journey is beyond my routined responsibility, I am always ++ to my role and I would continue doing the same here as well.

Projects:

Case deflection: Owned the challenge of external case deflection. Involved in creating necessary knowledge goals for the team, improving search engine tuning, and redesigned search panels. Resulted in 300% improvement in search deflection.

AWS and Azure Cost notifiers: Built infrastructure cost control dashboard, weekly slack notifiers with thresholds. Build processes to shutdown VM's and storage that don't have any tags. Cost governance significantly saved costs over 100K/month.

ML implementation of Case Routing: Spearhead the initiative to assign cases to spark and platform from traditional rule-based model to ML model. This improved case assignment accuracy from 60% rule based to 85% via ML model. These saved support leads time in reassigning cases and aided in faster resolution.

Dashboard to understand rate limits: Customers often raised cases as they hit rate limits due to surge in their peak loads. Launched an initiative to develop a dashboard to detect if there are customer's job APIs hitting a limit. This project improved time to resolve customer cases.

PVC Billing: Streamlined monthly billing process for Private cloud customers, where Databricks is deployed with customer environments. This process helped in bringing back revenue to Databricks on the 10th of every month.

Automation initiatives: Engineers often spend time in finding the log URL's, Genie URLs to login to customer workspace. Lead an initiative to create a browser plugin to auto construct these URLs for every case to improve engineers' productivity.

Ask Support: Dedicated Support Assistant powered by AI who can assist customers in fixing issues , answering queries related to our product.

3. Describe how you solved a challenge or technical issue that you faced in a previous role
(preferably in a previous support role). How did you determine that your solution was Successful?

Ans:
In a previous role as a solution engineer manager in a support team, I faced a situation where a customer faced repeated issues with their Spark jobs in the Databricks environment, leading to multiple escalations. The issue revolved around performance degradation and inefficiencies in processing big data tasks. Given the complexity and technical depth of the problem, direct debugging wasn't the onlysolution; it required a thorough approach to ensure customer satisfaction and job optimization.

Strategy and Implementation:

Understanding the critical nature of the issue, I spearheaded a strategic approach focusing on Spark job optimization. We began by revisiting the basics of data partitioning, ensuring it was strategically aligned with the workload characteristics to enhance parallelism and reduce processing times. Recognizing the importance of caching and persistence, we advised on effectively caching frequently accessed DataFrames, which significantly reduced recomputation needs.

To address data skewness—a common yet often overlooked issue—we utilized Databricks' tools for identifying and mitigating skewed data, such as adjusting the spark.sql.shuffle.partitions configuration. This was complemented by leveraging Databricks' auto-optimization features and integrating Delta Lake for its advanced data management capabilities, further enhancing performance.

We also emphasized the importance of cluster configuration, guiding the customer to select the appropriate cluster size and instance type based on their data size and job complexity. This advice extended to storage considerations, advocating for SSDs to improve shuffle operation performances.

Ensuring Success:

The determination of our solution's success was twofold. Firstly, technical metrics indicated a significant improvement in the Spark jobs' performance, with reduced processing times and more efficient resource utilization. Secondly, and most importantly, was the customer's feedback. The client expressed high satisfaction with the impact on their operations, noting the marked improvement in their data processing tasks' efficiency and cost-effectiveness.

This positive outcome was reinforced by continuous follow-up and the establishment of a feedback loop, ensuring that the optimizations remained effective over time and that the customer felt supported throughout the process. The experience underscored the importance of a holistic approach to problem-solving in technical support roles—not just addressing the immediate technical challenges but also ensuring the customer feels valued and heard, aligning with the philosophy that people will never forget how you made them feel.

4. When would you choose to use Edge Functions, Serverless Functions, or Edge Middleware with Vercel?
Ans: NA

5. Imagine a customer writes in requesting help with a build issue on a framework or technology that you've not seen before. How would you begin troubleshooting this and what questions would you ask the customer to understand the situation better?

Ans:

When faced with a customer's build issue in an unfamiliar framework or technology, my initial approach to troubleshooting would involve a structured process to understand the issue in depth, gathering necessary information, and leveraging available resources for an effective solution. Here's how I would proceed:

Initial Steps for Troubleshooting:

I would First, acknowledge the customer's issue and reassure them that I am here to assist. It is important to make the customer feel supported and confident in your willingness to help solve their issue.

Lets conduct a preliminary research on the specific framework or technology mentioned by the customer. This can involve reviewing official documentation, forums, or any internal knowledge bases for similar issues.

Key Questions to Ask the Customer:

To probe is an important aspect in customer facing role:

Could you provide a detailed description of the build issue you are facing, including any error messages or logs?

Can you specify the version of the framework or technology you're using, along with details about your development environment (e.g., operating system, IDE version)?

What steps can we take to reproduce the issue on our end? This information is crucial for diagnosing the problem accurately.

Have there been any recent changes to the codebase, dependencies, or environment prior to the issue occurring?

Have you tried any solutions or workarounds so far? This helps avoid repeating steps and can offer insights into the problem's nature.

Approach to Finding a Solution:

Consult with colleagues or teams who might have experience with the technology or have faced similar issues. Their insights can be invaluable.

Engaging with the SME can uncover common pitfalls or solutions.

Try to isolate the issue by breaking down the build process into smaller parts. This can help identify the exact point of failure.

If possible, replicate the customer's environment and attempt to reproduce the issue. Hands-on troubleshooting can often reveal aspects of the problem that are not immediately obvious.

Ensuring Resolution and Follow-up:

After identifying a potential solution or workaround, communicate clearly with the customer, providing step-by-step guidance. Confirm with the customer whether the solution resolved the issue and offer further assistance if needed. Finally, document the problem and the resolution process to enhance the knowledge base for future reference, improving the team's readiness for similar issues.

6. The customer from question 5 replies to your response with the below:
“I’m so frustrated. I’ve been trying to make this work for hours and I just can’t figure it out. It must be a platform issue so just fix it for me instead of asking me questions.”
Please write a follow-up reply to the customer.

Ans: Subject: Your Support Request: Let’s Get This Resolved.

Dear Customer,

I understand how challenging technical issues can be, especially after dedicating hours trying to find a solution. Please rest assured, we’re here to support you every step of the way and are committed to resolving this matter as quickly as possible. Your efforts and patience are greatly appreciated.

To ensure we address this efficiently, gathering specific details is crucial for pinpointing the exact issue within the Databricks platform. This method helps us in providing a precise solution directly aimed at the root cause. However, I realize that after much effort, additional questions can feel overwhelming.

To move forward and get this resolved promptly, here’s our immediate plan of action:

> **Escalation**: Your issue has been escalated to our senior support engineers, who are well-versed in swiftly resolving complex issues.
> **Review**: Alongside, I'll look into similar platform issues to see if there are immediate insights or solutions applicable to your case.
> **Direct Assistance**: A quick call or screen-sharing session often leads to faster resolution. Please let me know a time that works for you, and we’ll make it happen.

Your satisfaction is our priority, and we’re focused on turning this situation around for you. If there's a specific time you’re available for a brief call or any additional details you recall – no matter how minor they seem – it could be invaluable to our troubleshooting.

Thank you for your cooperation and for being part of our community. We are on this together and I am optimistic we’ll find a solution soon.

Best regards,

Rajeev Kumar
Senior Solution Manager

7. A customer writes in to the Helpdesk asking "How do I do a redirect from the /blog path to https://example.com?" Please write a reply to the customer. Feel free to add any information about your decision making process after the reply.

Ans:
Subject: Assistance with Your Redirect Query
Dear Customer,

Thank you for reaching out with your question about setting up a redirect from the "/blog" path to "https://example.com". I'm here to help you through the process, and I'll guide you step-by-step.

For Apache Web Server Users

Find or Create Your .htaccess File: This file is usually located in the root directory of your website. If it doesn't exist, you can create a new text file named **.htaccess.**

Edit the .htaccess File: Add the following line to the file:

```
Redirect 301 /blog https://example.com
```

This command permanently redirects visitors from “/blog” to “https://example.com”.

For Nginx Users

Locate Your nginx.conf File: **Typically found in /etc/nginx or /usr/local/nginx/conf.**

Edit the nginx.conf File: Insert the server block below into your configuration:

```
server {
   location /blog {
      return 301 https://example.com;
   }
}
```

Adding Screenshots for Clarity:

- Editing Files: A screenshot showing the edited part of the .htaccess or nginx.conf file with the new line highlighted.
- File Location: A screenshot of the file directory where the .htaccess or nginx.conf file is located, to help users find it.

Important Note: Before making changes, it’s crucial to back up your configuration files to prevent any potential site access issues.

Should you need further assistance or have additional questions, don't hesitate to ask. I'm here to ensure everything runs smoothly for you.

Best regards,

Rajeev Kumar
Senior Solution Manager

8. A customer is creating a site and would like their project not to be indexed by search engines. Please write a reply to the customer. Feel free to add any information about your decision making process after the reply.

Ans:
Subject: Assistance with Preventing Search Engine Indexing for Your Site

Dear Customer

Thank you for reaching out with your request. I understand you're looking to ensure your project remains unindexed by search engines, and I'm here to guide you through the process.

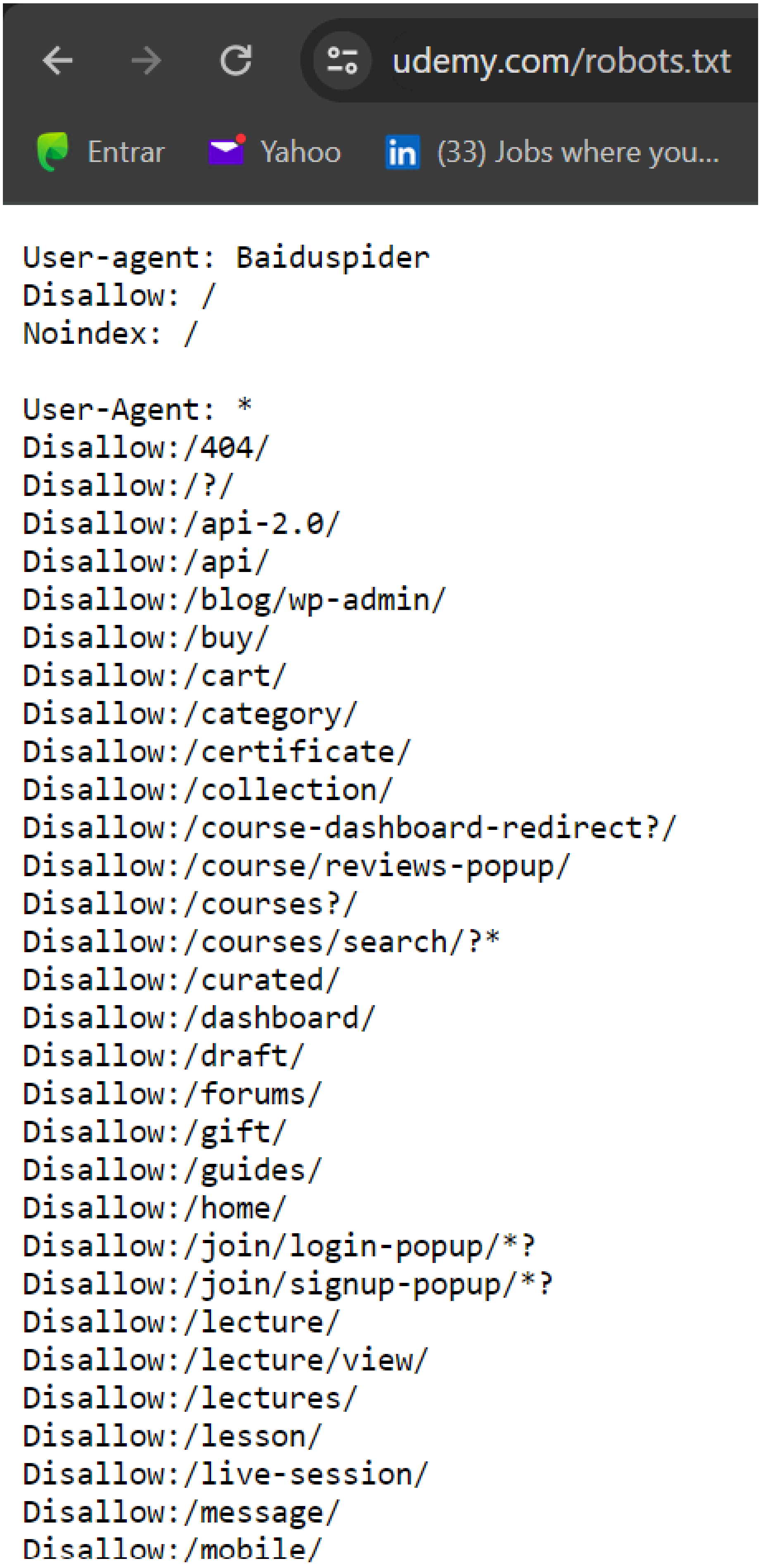
To prevent search engines from indexing your site, you can use a combination of the robots.txt file and meta tags within your site's HTML pages. Here's how:

Using robots.txt

Create a robots.txt file in the root directory of your website.
Add the following lines to the file:

```
User-agent: *
Disallow: /
```

This tells all search engines not to crawl any part of your site.



```
User-agent: Baiduspider
Disallow: /
Noindex: /

User-Agent: *
Disallow:/404/
Disallow:/?/
Disallow:/api-2.0/
Disallow:/api/
Disallow:/blog/wp-admin/
Disallow:/buy/
Disallow:/cart/
Disallow:/category/
Disallow:/certificate/
Disallow:/collection/
Disallow:/course-dashboard-redirect?/
Disallow:/course/reviews-popup/
Disallow:/courses?/
Disallow:/courses/search/?*
Disallow:/curated/
Disallow:/dashboard/
Disallow:/draft/
Disallow:/forums/
Disallow:/gift/
Disallow:/guides/
Disallow:/home/
Disallow:/join/login-popup/*?
Disallow:/join/signup-popup/*?
Disallow:/lecture/
Disallow:/lecture/view/
Disallow:/lectures/
Disallow:/lesson/
Disallow:/live-session/
Disallow:/message/
Disallow:/mobile/
```

Using Meta Tags

Add the following meta tag to the <head> section of each HTML page you wish to be unindexed:

```
<meta name="robots" content="noindex, nofollow">
```

This prevents search engines from indexing the page and following links from it.

Important Considerations

- robots.txt is effective for most crawlers, but it's a directive, not an enforcement. Some search engines may choose to ignore it.
- Meta tags provide a page-level instruction, which is generally adhered to by search engines.
- Ensure no internal links lead to the pages you want to keep unindexed, as this can sometimes lead to accidental indexing.
- For complete privacy, consider password-protecting the pages or using a login system, as these methods ensure that content is not accessible without proper credentials.

If your site is hosted on a platform with specific settings for managing search engine visibility, there might be an easier way to adjust these settings directly from the platform's dashboard. Please let me know if you need instructions for a specific platform.

Should you have any further questions or need more detailed assistance, feel free to reply to this email. We're here to help ensure your site is set up exactly as you need.

Best regards,
Rajeev Kumar
Senior Solution Manager.

9. What do you think is one of the most common problems which customers ask Vercel for help with? How would you help customers to overcome common problems, short-term and long-term?

Ans: I would like to talk about the most common problems customers seek help with from Databricks perspective, a few key areas emerge, notably around optimizing Spark jobs,

managing environments, implementing CI/CD practices, and efficiently handling data movement.
Here’s how I would address these issues for customers, both in the short and long term:

Common Problems:

Performance Optimization:
Short-term: Guide customers on utilizing monitoring dashboards in Azure Databricks to identify and address performance bottlenecks in Spark jobs.
Long-term: Offer workshops on advanced Spark tuning and optimization techniques to prevent future bottlenecks.

Compute Behavior Troubleshooting:
Short-term: Provide immediate support in diagnosing and resolving compute behavior issues within their workspace.
Long-term: Develop a series of best practice guides for environment setup and maintenance to mitigate compute issues.

CI/CD Practices:
Short-term: Assist with the integration of CI/CD practices in data structures for smoother development workflows.
Long-term: Host training sessions on advanced CI/CD methodologies tailored to data projects.

Data Movement:
Short-term: Advise on using Databricks for fast, parallelized compute solutions and managing table versioning and history effectively.
Long-term: Create in-depth tutorials on data architecture strategies to optimize data movement and storage.

Environment Management:
Short-term: Confirm all necessary components (Spark, Delta, Java) are installed and properly configured.
Long-term: Continuous education on managing and updating these environments without disrupting workflows.

Cost and Job Monitoring with Overwatch:
Short-term: Help set up Overwatch for monitoring job costs and performance.
Long-term: Regular reviews of cost metrics and performance data to refine budgeting and resource allocation.

Autoscaling and Tuned Spark Clusters:
Short-term: Assist in setting up autoscaling and tuning Spark clusters for immediate efficiency gains.

Long-term: Periodic review sessions to adjust scaling and tuning parameters as workloads evolve.

Leveraging Databricks Features (SQL, MLFlow, etc.):
Short-term: Quick-start guides on implementing these features for new projects.
Long-term: Deep-dive sessions and case studies showcasing innovative uses of Databricks features.

Navigating Frequent Updates:
Short-term: Summarize key points of the latest updates for easy consumption.
Long-term: Establish a subscription-based update notification and briefing service for customers.

Adhering to Best Practices (Delta Lake, ML Ops, Unity Catalog):
Short-term: Provide checklists and basic guides on best practices for immediate improvement.
Long-term: Develop comprehensive training modules and certification programs around these best practices to ensure deep, institutional knowledge.

10. How could we improve or alter this familiarization exercise?
Ans: This approach is good to understand the thought process of any manager to check his experience about the support life cycle.