Q1. Tell me about yourself?

Success

Yes, sure you know my name is Sherin Mathew, I have been in the IT industry for the last 20 years. I specialise in developing software. In the last 3 months I have been working as tester

Strength

My real strength is my ability to truly understand what the business wants and the clients want. I pride myself on my reputation on working in that environment.

Situation

What I am looking for is a company that I could add value to, that I could produce a positive return on investment for , where I could join a strong team. Is this what your company looking for?

Thank you for taking time to see me for the interview.

Build and grow whatever I know, improve my skills and achieve the goals like a test architect. The importance of QA is If the QA say NO that means NO. You can’t release.

TAC & WorkSafe

The TAC is a Victorian Government-owned organisation whose role is to promote road safety, support those who have been injured on our roads and help them get their lives back on track.

Our purpose

To care for the lives of everyone who travels on Victoria’s roads.

Our vision

To be the safest place in the world for road travel.

Our mission

To champion road safety and help those injured get their lives back on track.

Our values

These values guide our behaviour, our thinking and ultimately the treatment we provide injured Victorians.

**We value life.**  
We deliver world-class road safety initiatives to save lives. We value quality of life for our clients, and treat them with care and respect to get their lives back on track. We value the well-being of our people.

**We make every conversation count.**  
We understand that trust is built and reinforced, one conversation at a time. We ensure every conversation with colleagues, clients, partners and our community is authentic, is heard, and is acted on.

**We will find a better way, today.**  
We are bold and brave in seeking better ways to care for our clients and prevent road trauma. Our people are empowered to achieve success for our clients through innovation and creativity.

**We make the complicated simple.**  
We remove the barriers that get in the way of achieving the very best outcome for clients and in doing so, simplify processes and interactions - and not just for clients but our partners and the broader community too.

WorkSafe Victoria

WorkSafe is Victoria's workplace health and safety regulator. We are also the workplace injury insurer.

Our job is to reduce workplace harm and improve outcomes for injured workers. This is our promise to the Victorian community and the reason we exist.

Maintained support tickets in Jira and ensured its timely action based on the risk profile.

Jira is a popular project Management and issue tacking software.

Used Jira board with 3 columns “To DO’, “In Progress”, “Done”

Implemented and maintained data security measures to protect sensitive information.

We have SOP based on GDPR (General Data Protection Regulation). Document rules and procedures for data protection. Use strong passwords, multi-factor authentication (MFA), and single sign-on (SSO). Regularly update and patch software to protect against vulnerabilities. Regularly back up data and store it securely. Third party Data Backup service providers, who are comply with your security standards.

Liaise scope and estimation with other stakeholders (sales consultants/business analysts) based on client requirements and provide regular updates on project progress and status.

Effective communication and collaboration with stakeholders such as sales consultants and business analysts are crucial to ensuring project success.

Performed code review and provide support to junior programmers and ensure code quality and standards are met.

Establish Clear Coding Standards, Define Guidelines: Consistency, Peer Reviews, Open Communication, Document and Share Knowledge

Developed and Tested API’s.

Xerox API

I successfully integrated the Xerox web application with our company website using APIs. Xerox offers a browser-compatible web application called Xerox Workflow Central, which is designed for creating and managing workflows.

To enhance our customer experience, I leveraged Xerox APIs to establish a seamless connection between our company website and the Xerox workflow system. This integration allows our customers to directly upload sample files, such as spreadsheets, through our website. These uploads automatically create corresponding jobs in the Xerox Workflow application.

The integration was implemented using Node.js scripting, which provided the necessary flexibility and efficiency to handle the API interactions and data flow between our website and the Xerox platform.

This solution streamlined our workflow process, improved customer interaction, and demonstrated my ability to work with third-party APIs and create practical, user-focused integrations.

Node.js and APIs

 Node.js has built-in modules and popular third-party libraries that make it easy to send HTTP requests, which is how most APIs are accessed.

 The most commonly used methods for making API calls in Node.js include:

* Built-in http or https modules
* Third-party libraries like axios, node-fetch, or request

 Node.js's asynchronous, event-driven architecture makes it particularly well-suited for handling API calls, especially when dealing with multiple requests or long-running operations.

Used the Postman tool for testing the API.

Designed & developed web applications for various clients based on requirements using technologies such as Python, JavaScript, NodeJS and mongoDB.

Python and Node.js can be used together for web development. While it's more common to use one language or the other for a project, there are scenarios where both can be utilized. For example, you could use Python for the backend logic and Node.js for real-time interactions using WebSockets.

Create API call: This can be tested using postman

GET all users: GET http://localhost:3000/api/users

GET user by id: GET http://localhost:3000/api/users/1

POST new user: POST http://localhost:3000/api/users with body {"name": "Charlie"}

MongoDB is a document oriented NoSQL database offers high performance, high availability, and easy scalability.

Created and executed unit, SIT & end to end test cases for EMC Captiva document capture & management solutions, ensuring robust functionality and performance.

Creating Unit Test Cases. Use a unit testing framework using Python. This units combined can be tested called SIT(System Integration Testing)

Create SIT Cases. Write test cases to cover all integration points. Use Postman for testing APIs.

For the Web application we have UI Testing, Selenium is used for testing web based application using Python.

Identify the Use Cases of any application or web page.

Negative Scenario and Positive Scenario. How the break the application.

Captiva used pre-defined rules and templates for capturing unique identifiers.

EzeScan has a template based Extraction using samples.

Git

git –version

clear

**Configure Git**

git config --global user.name "Sherin Mathew"

git config --global user.email [sherin.mathew@gmail.com](mailto:sherin.mathew@gmail.com)

**Two ways you can work with Git. 1. Create a new program or task. 2. Use some ones code to import and work.**

1. **Create new project from scratch.**

git init

git status

*Git Status is to find the status of the git.*

git add index.html

*This will a file in to the cache*

git status

This will give the status the new file is added.

git commit -m “Initial Commit”

This is the first time we have the file to be committed to the repository with a label ‘Initial Commit’. Saying working directory tree is clean.

Give some changes to the file. Say Hello How are you?

Then check the status, you will the comment file modified.

Now create another commit.

For that again do

git add index.html

git commit -m “Added First Edit”

git log

This will show all the commits happened to the file.

git checkout 75e238e30886821c884edd0b1448d9a0d66f2152

This will move to a different commit stage, here the above id is the previous stage.

Look at the file index.html, it contains the first version.

Now if we look at

git log –all

will show the ‘master’ is at the last commit and ‘HEAD’ is at the latest checkout point. Hear master is the branch name.

Now if you want to checkout to the last commit point again do

git checkout master

Check the index.html file, will contains the updated file. Now the HEAD moves to the last commit.

**Branching**

git branch feature

This will create a branch named FEATURE. Now you can add a file or edit the file in the repository.

You can switch between the master and feature branches.

Merging

Fast Forward

This will merge two branches together. The HEAD will go with both branches. This merge is called Fast-Forward Merge.

git branch

This will show where you are at. You can simply switch branches by using git checkout to that branch.

git diff

This will show the difference of the file before committing. You can compare the file content.

git stash

This will work as an undo comment, this will remove all the changes before committing a file.

Recursive Merge

Suppose we have two branches, master and feature. Currently we are in master and did some changes in one text file. Add and commit that file. Now the Head is with master branch. Then checkout to feature branch, do some changes in the second text file, add and commit that file. Now the head is at feature branch. Here we have two branches master and feature going in two different directions. Now this merging is called recursive merging. For that we use the same merge command

git merge feature

This will merge, where you can add comment in git text file, using Insert key and :wq

Conflict

Some time if the developers did changes in the same file. This time the git can’t automatically suggest which needs to be merged. Here we need to manually intervene and choose the correct file.

Here checkout in master do some changes in the 1st text file. Add and commit, then checkout to feature do some changes in the same text file edited earlier. Add and commit. Now when you merge you will get a message that, there are conflicts. Open the text you will both versions on the same file, omit the unwanted and include the needed code. Then add and commit the same text file. That will fix the conflict.

Connect to GitHub, GitLab and upload it from git

Login to GitHub, create a repository and copied the link of that repository.

git remote add testfolder <https://github.com/Shermath/interview.git>

This command actually add the git content in to the cloud GitHub. Here testfolder is the location name we created.

git remote

will show the name of the remote file.

$ git remote -v

This will give the folder locations.

Push

git push testfolder master

This will push the files from the computer to the GitHub.

You can push feature as well.

Now it is in GitHub, you can make changes always in the files. Later add, commit and simply push with the above command.