Tell us about yourself?

This is **-**, I ----- been in testing industry for about six years. I have been doing Manual and Automation Testing of Client server and web based applications. I have involved myself in software development lifecycle from start to finish until we deliver it to client. I have extensive experience in doing different types of testing like Functional, integration, backend, UAT and System Testing. I am expert in using tools like Quality center, Selenium, also familiar with Soap UI. In my past experience I have created many documents for example Test plan, Test Cases, RTM, Test Analysis report, meeting minutes and user manuals. That's pretty much about myself.

Tell us about your current Project?

Washington Post, Washington DC

APPLICATION- Application is about Paywall system that prevents user to access Washington post content without a paid subscription.

MODULES-

1. Initially enrolling in home delivery print customers for Sunday paper should get free access to all digital.

* Washington post employees should be able to have free access to all contents.
*Users on .gov and .mil should be able to access all contents

Tested modules:
Homepage
Section Fonts
Videos
Commercials (classified, Shopping deals)
Education module content will be free for any user

Email Management:
Alerts/Reminders are triggered from the system as they are trying to reach 20 limited contents
Billing Management:

User is able to pay by credit card only for digital subscription visa, MasterCard and amex
* While completing the enrolment, the system provides guidance and messages as to what information is required for the enrolment type and if any entered information is invalid.
* During the submission process, the system validates and confirms information and allows the users to electronically submit an enrolment for processing and approval.

I have done several things with these modules, such as front-end and back-end testing to ensure that the correct data was being transmitted from our back-end database; I also created invalid credentials and tested them to ensure privileges are given to the correct user. The methodology being used at Washington post is agile methodology. When I started with the company they were already on the design phase. I found a lot of bug/defect every day when I reported them to the developers they were fix based on priorities and severities levels, I also performed Smoke, GUI, Functional, Regression, System, UAT, Interface, Integration, and Backend Testing based on necessarily. I worked with SQL queries based on requirement so any other that not a requirement is definitely a defect. I also used HP QTP to create automation scripts for my test cases, since the application was unstable and a new build was being released often, I developed script using descriptive programming; later on, I created a data driven framework first because in the beginning we had to test the application with different set, and combination of data, I then continues with keyword driven framework, and later combined both of my framework to and hybrid framework. I enhanced my script with the use of environmental variables, check points, custom functions, and recovery scenarios. I used advanced VB scripting and descriptive programming for each object in my application.

Project Activities: (Story)

When I joined **Washington Post**, I was involved in planning phase. I was present in the meeting with the client, product development team, developers, designers, DBA, Business Analyst gather the necessary information for the project and provided my feedback. After final approved requirements, I have created test plan based on requirement document, design document, Project Plan. I called Test Plan review meeting and based on the feedback from the team members, I finalized Test Plan and sent to Project Manager for approval. After approval I uploaded Test Plan on Share point so all the team members have access to it. Developers had already started coding and I attended code reviews meetings and basically getting I acquainted with all the design and application requirements. Based on requirements I started creating Test Cases using QC. Initially I performed several functional tests such as smoke, functional (Positive & Negative), boundary value analysis. While performing manual testing I reported defects using Quality Center to track system to log, manage and report all the defects. When developer fixed the defect, I performed regression testing to make sure the fixed bug didn’t create problems in other parts of application. As I was proceeding towards automation, I based my automation testing on the analysis of test cases considering which functionality was most important to the project intended, most visible to the user, has largest safety and financial impact, what kind of test could easily cover multiple functionalities, what kind of problems would cause the most customer service complaints etc. As I was performing Functional testing using QTP to verify whether the product meets the intended specifications and functional requirements laid out in the documentation. I also used HP QTP to create automation scripts for my test cases, since the application was unstable and a new build was being released often, I developed script using descriptive programming; later on, I created a data driven framework first because in the beginning we had to test the application with different set, and combination of data, I then continues with keyword driven framework, and later combined both of my framework to and hybrid framework. I enhanced my script with the use of environmental variables, check points, custom functions, and recovery scenarios. I used advanced VB scripting and descriptive programming for each object in my application. I followed functional testing techniques for example Ad-Hoc sometimes I performed ad hoc testing prior to the starting with any of the testing. Often times it helped me learn the application and decide the scope and duration of the various other testing. I also caught a lot of calculations, data integrity, decimal points, sql errors, and boundary value analysis bugs. Once all the major bugs were fixed. We moved to UAT testing, our client manager brought few Testers from client side and I was the one chosen to perform Demo/walkthroughs for the UAT Testers. Then I became part of UAT

Testing team and tested alongside Client side testers. No major bugs were found, user suggested some modifications and new requirements and they were very satisfied with the system and impressed by user friendliness of how process works in our system. We also provided some ongoing support for them.

Daily Routine:

A big part of my day is dealing with bug reports - making sure they have enough detail for developers to understand, going back to users with workarounds or questions about browsers, setup etc. Finally making sure bugs reports have the right priority and adding them in to the work backlog (yep, we're talking agile here).

Next priority is making sure I know what's coming up, talking to the product managers, tech leads (lead developers), designers and being part of acceptance criteria (requirements) creation. Using this info to write tests plans before development is complete.

Next up its releases, I make sure all releases are passing our release acceptance criteria (currently a set of tests). Sometimes I test them but other times I've worked with the developers so they know what to test. New features are tested (and earlier I would have written a test plan).

Strategy gets squeezed in when there is time - reviewing how tools are working for us, reading books, blogs, STC etc. Coming up with new ideas to make things faster and less buggy

Agile-Scrum:

Scrum is an Agile process that allows to focus on delivering the highest business value in the shortest time. It is considered as lightweight and people-based / skill based rather than plan based.

Sprints: In the Scrum method of agile software development, work is confined to a regular, repeatable work cycle, known as a sprint or iteration. In by-the-book Scrum, a sprint is 30 days long, but many teams prefer shorter sprints, such as one-week, two-week, or three-week sprints. But how long each sprint lasts is something for the team to decide, who must weigh the advantages or disadvantages of a longer or shorter sprint for their specific development environment. The important thing is that a sprint is a consistent duration. (It could be 2 weeks or upto 1 month)

Team Members:

IT TEAM: 8 – 10 Developers, 2 – 4 QA, 1 Tech Lead, 1 QA Lead, IT Manager, Project Managers, Business analysts and System Analysts, Agile Master.

Bugs:

Calculations, Data Mapping, Data Integrity, Decimal Points, Field Level Validations (Boundary value analysis), SQL procedures error

ALM Activities: As a User, Admin, and Project Admin (will send you an email separate Document)

- **QTP Activities:** QTP Experience

  Introduction: QTP is a functional and regression automation testing software that tests both web-based and windows applications. Since QTP is fully automated testing method, therefore it is way faster than manual testing, easily repeatable, reusable and cost efficient.
  There are two ways I created a QTP script. Such as, by recording and descriptive programming:
  There are three recording mode in QTP, such as Normal Mode, Low Level Mode, and Analog Mode. QTP uses Normal Mode by default, which records almost anything. When Normal Recoding Mode fails due to recognition problem, then I use Low Level recording Mode to force QTP to record specific objects. Analog Mode is the recognition of keyboard inputs, mouse clicks, and signature.

  After I recorded the script I have done some enhancement in the script. First able, I put various cheek point in the script. A checkpoint verifies that expected information is displayed in an Application while the test is running. There are few kinds of check points such as Standard checkpoint, Image checkpoint, Table checkpoint, Page checkpoint, Text checkpoint, Bitmap checkpoint, Database checkpoint.

I used synchronization in the script. Synchronization is waiting for one object to finish its action to go to next object. Sometimes QTP script runs faster than web pages. That's why I used wait statement into the script so that QTP will wait for few seconds to display the objects properly.

Then I did parameterization of the script. Parameterization is using same script for multiple sets of data. There are two types of parameterization, input parameterization and output parameterization. In Input parameterization data is entered manually before executing the test. On the other hand, output parameterization retrieve data dynamically from database during or after the executing the test.

I did regular expression to my scripts. Regular expression is a string that will capture the dynamic value during running the script. Every time, we run the script there could be come up some dynamic values in href Properties. I used regular expression formula in order to capture the dynamic value. The formula is [0-9a-zA-Z\w\W]*.

I also used optional step which is used to block unnecessary pop up box, warning, and massages. If I know the location of the pop up box or massage comes in the script then I use optional step. I can fix this problem with if – then-else statement. However, there is a chance that an unexpected error or security alert comes in the script at any time or any place in the script then, I create recovery scenario and add to the script in order to pass the test.

I also used Modular test method to my script which is dividing a test into several actions that can be reused or edited or called by several other tests. It is also known as code re-usability.

I also created QTP script by using descriptive programming and custom functions. Descriptive programming is used to bypass the use of object repository. In this case object spy is used to gather the property of an object and then hand code the properties in the script. For example, I would like to send an email from yahoo mail, now let me give you an example how to write code for Descriptive Programming.

```
SystemUtil.Run “iexplore.exe”, www.yahoo.com
Browser(“micclass:=Browser”).Page(“micclass:=Page”).Link(“micclass:=Link”,
“Name:=Mail”).click
```

Now let me talk about Custom Function. As we know there are a lot of built-in Function in QTP, however if an application requires a function that is not in QTP, then I need to write the specific or custom functions in order to enhance my script. I can create functions like clickLink, clickImage, enterEdit, chosewebElement, selectCheckBox, choseRadioGroup etc.

I did ODBC connection to my script. ODBC is open database connectivity which is used for retrieve data from different types of database including Access, Excel and Oracle.

In addition, scheduling so that selected QTP scripts will run automatically when I am away from my PC, QTP batch runner has some limitations that it can not open QTP. That’s why I created driver script and add to the scheduler so that it will open QTP and run the scripts automatically.

Dictionary object is similar to a typical array. Each dictionary object has a unique key associated with every item. This unique key can help us in calling that item whenever required. The Dictionary object enables us to assign values to variables that are accessible from all actions (local and external) called in the test in which the Dictionary object is created. To use the Dictionary object, we must first add a reserved object to the registry by following the below mentioned steps:

Go to Start-> Run. Run dialog opens.
Type regedit.exe and click ok. Registory Editor opens
Go to HKEY_CURRENT_USER\Software\Mercury Interactive\QuickTest Professional\MicTest\ReservedObjects\
Under ReservedObjects create a new Key (Highlight ReservedObjects, go to Edit menu->New ->Key, and name it Dictionary. Right click on DIctionary, go to New-> String Value, Type the name as ProgID. Double click on this ProgID, Edit String dialog opens, type the Value data as Scripting.Dictionary and click ok)

Environment variables in QTP are like global variables which can be accessed through any part of the script. The values of these variables remains same irrespective of the number of iterations (unless I change them through scripting). These variables can prove to be very useful when you want a variable to be shared across various reusable actions.
There are two types of environment variables:
Built-In: These are the internal variables that are provided by QTP. Among others they can provide you valuable information like the path of the folder where test is located, the path of the results folder, the name of the action iteration or the OS version, in Built-in variables (OS,OS Version, Product Dir, Product Name, Result Dir, Test Dir, Test Name, User Name). Other kind of Environmental Variable is User Defined which is we add the variables in the dictionary. There are two types of user define environmental variables, internal and external.

SQL Query: For interview purpose

SQL Experience:

The list of things we generally cover in database testing:
1. Data integrity-The complete data belonging to each entity should be stored in the database. There should not be any missing data
2. Correctness of the data stored in the database- The data stored should be correct and stored in right place.
3. Data type testing
4. Data size testing- Generally we test Data size only at the front end, but it is essential to test it at back end separately
5. Database performance
6. Data security
7. In case of data migration check Correctness and completeness of data
8- If you are good in database also test for Stored procedure. in this we need to test Every Stored Procedure separately. which may cover:
-. The no. of arguments being passed
-. The data type of each of the arguments being passed
-. The order of the arguments being passed
-. The return value
-. The data type of the return value

9. Input Item verification- In this we verify the input items like text box, combination box, active X controls. generally ask your developer to test this during unit testing.

As in point 5, it is pointed out that we need to test for performance. So what can we check in database. We measure the executing time to see the performance. Also check the indexing. The poor indexing may cause your application slowness. So always there should be a proper Indexing for better performance.

I am pointing out a simple example here that how can we verify the database entries manually.
Suppose we have a registration form. So in from end we only have a UI where there must be some input fields and a submit button. Most common test is to fill

the form and verify weather that user is created or not. But we don't know what is going on in database. So taking this example we first gain full knowledge of our database. Which field is mapped with which table and what are the columns where data will be inserted.

Once you are familiar with the data model, we can write some queries using various tables and columns joining.

Now First thing to do- check in front end (execute all test case)

and second- test back end same thing by executing the queries to insure that the data is inserted in right table and the data is correct.