Task 1:

Imagine the following situation: You need to establish a QA process in a cross functional team. The team builds a front-end application using REST APIs.

1. Where would you start? What would be your first steps?

   The main aim here is the right product is built. This means ensuring requirements are correctly defined and development team has an understanding of the functionality before they start coding.
   So, it will be wise to start testing right from the beginning of the development process.
   I would form teams consisting of both developer and testers who would actively participate in building product right from the beginning. The ratio would be 3:1 (1 QA per 3 Dev).
   Step1: Story Workshops where the product owner, developers and testers gather in a room and start elaborating and fleshing out the details of the stories. This will give QA resources a better understanding of the functionality and they will be in a better position to develop right test cases. It will give a better understanding of the acceptance criteria.

   ---

2. Which process would you establish around testing new functionality? How would you want the features to be tested?

   Step2: In Sprint Testing/Verification Everyone should be responsible for quality of the product and not just testers. There needs to be sufficient amount of "developer testing" to ensure that the code written is of high quality before being deployed to a test environment for further testing. Each new piece of functionality should be unit tested. On top of that, there needs to be integration tests, API tests as well as UI tests. After developer has finished developing and unit testing the story the QA would verify this story as part of in sprint verification. Also, perform a certain level of regression testing to ensure existing functionality does not break. Apart from this there should be code reviews, test case reviews (buddy reviews) where testers can help developer with unit test cases.

   Step3: Story Sign Off QA will provide a sign off on the story when they are satisfied with the finished story with right evidence to support successful development.

   Step4: Testing System as a whole/Regression Testing After the sign off the new functionality test cases would be added in the regression test pack and this regression suite would be executed on the entire system to verify any issues so that the developers can provide a quick fix and it can be verified immediately.

   Automated Testing In an agile project, where a sprint typically lasts about two weeks, there is not enough time to do all the testing manually. There is testing of new stories as well as regression testing. So, it would be wise that regression tests should be automated to reduce the repeated execution. The activity of development and maintenance of automation

framework/scripts should be done at the very beginning of the sprint when the stories are still under development.

Continues Integration on Test Environment Here we can create hooks by using certain tools for smoke/sanity tests. The system should be setup in such a way that whenever a new code is checked in, a basic automated sanity suite starts the test execution and delivers a status report. This will help in identifying any initial issues the process/product.

Cross Browser/Other/Non-Functional Testing: Whenever required ad-hoc, exploratory or non-functional testing, such as performance, load and security testing should be performed on the system.

---

3. If you would do test automation which techniques or best practices would you use?

Since I would be working in an agile environment along with stakeholders and product owners, it would be wise to go with the BDD (Behavior Driven Development) framework. It can be applied for any type of testing including unit tests, component, integration. By writing tests with BDD we can also create specifications that can help the team understand tests and requirements much better. This means that along with writing tests, we are creating a clear tests documentation. Also, BDD helps the Business side (e.g. Test and Project managers) understand these tests, this brings additional value to testing because they can make recommendations based on business benefits.

Identify test cases which could be automated from the manual test suite.

Maintain different suits for Regression, smoke tests, cross browser tests.

Calculate the automation testing effort and compare it with the manual effort and document it (Effort Estimation). Also have automation test coverage document.

Other techniques would be going for a set design pattern like Page Object Model.

It would be also beneficial to create test jobs on a common server so that these jobs can be triggered by anyone, anytime without any dependency.