NAME : ANAMIKA PANDEY
REGISTRATION NO. : 12105805
UNIVERSITY : LOVELY PROFESSIONAL UNIVERSITY
GMAIL: anupandey1177@gmail.com

FRUGALTESTING

-EAGER TO WAIT FOR THE RESPONSE

1.Please confirm your consent to sign a bond for 30 months (including a 1-year internship). Only Submit your assignment if you are comfortable with this bond.

- Yes, I give my consent to sign a bond for 30 months(including a 1-year internship).

2.Have you received the stipend and CTC details? Please confirm by providing the details here.

- Yes , I have received the stipend and CTC details.
- CTC After Internship: 8 LPA(Where CTC Rs. 6 LPA Fixed + CTC Rs. 2 LPA Variable)
- Notice Period: 3 Months

3.Are you willing to relocate to Gachibowli, Hyderabad?

- Yes , I am comfortable to relocate to Gachibowli, Hyderabad.

4.What motivated you to pursue a career in software testing?

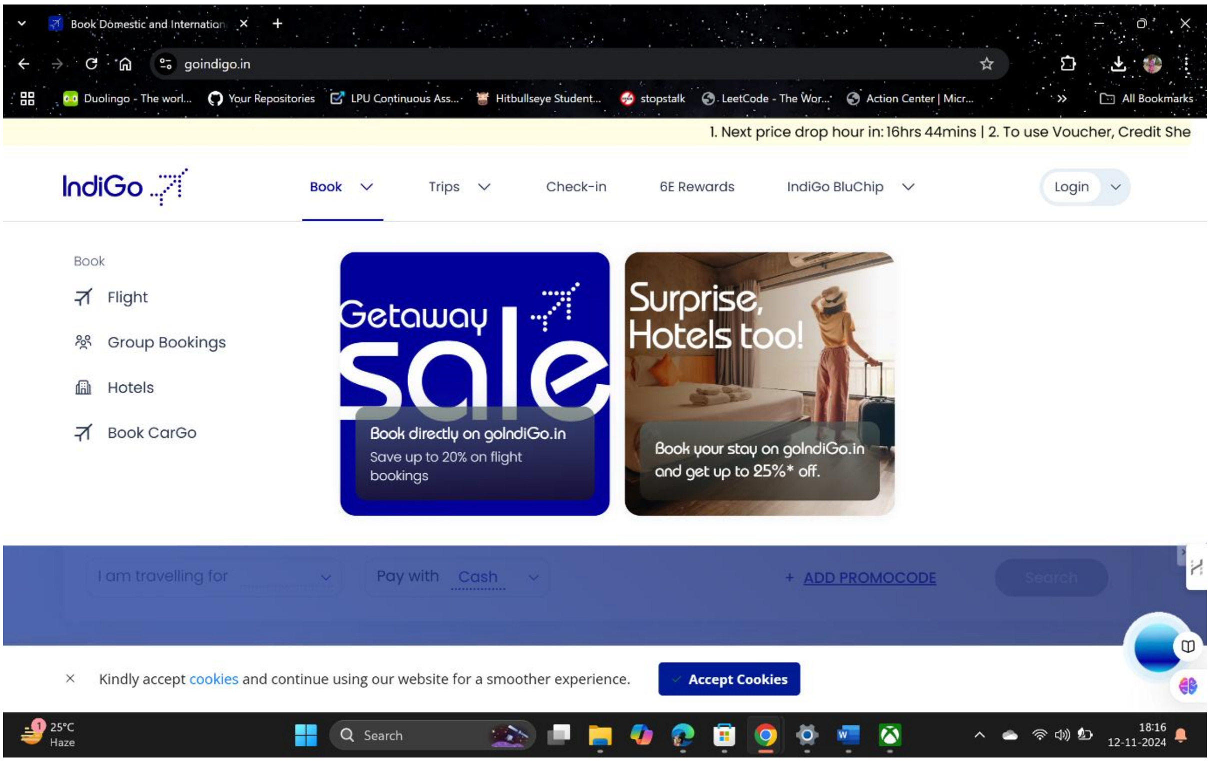
- As a student, I got interested in software testing because I've always been curious about how software really works and how it can be made more reliable. Testing just felt like a natural fit—it's all about problem-solving, thinking critically, and paying close attention to details, which I really enjoy. Plus, I love the idea of being involved in the development process in a way that makes a real impact, making sure the final product is actually good quality and works well for users. It's rewarding to think I could help create software that people trust and enjoy using.

5. Why do you want to join a software testing firm like Frugal Testing?

- I want to join a software testing firm like Frugal Testing because it's a company that truly values quality, efficiency, and innovation in software testing. I appreciate that the company focuses on affordability without compromising quality, which shows a strong commitment to making high-quality testing accessible for a wide range of clients.
  Joining Frugal Testing would allow me to work in a collaborative environment where I can learn from experienced professionals, sharpen my testing skills, and gain hands-on experience with real-world projects.

SECTION A:

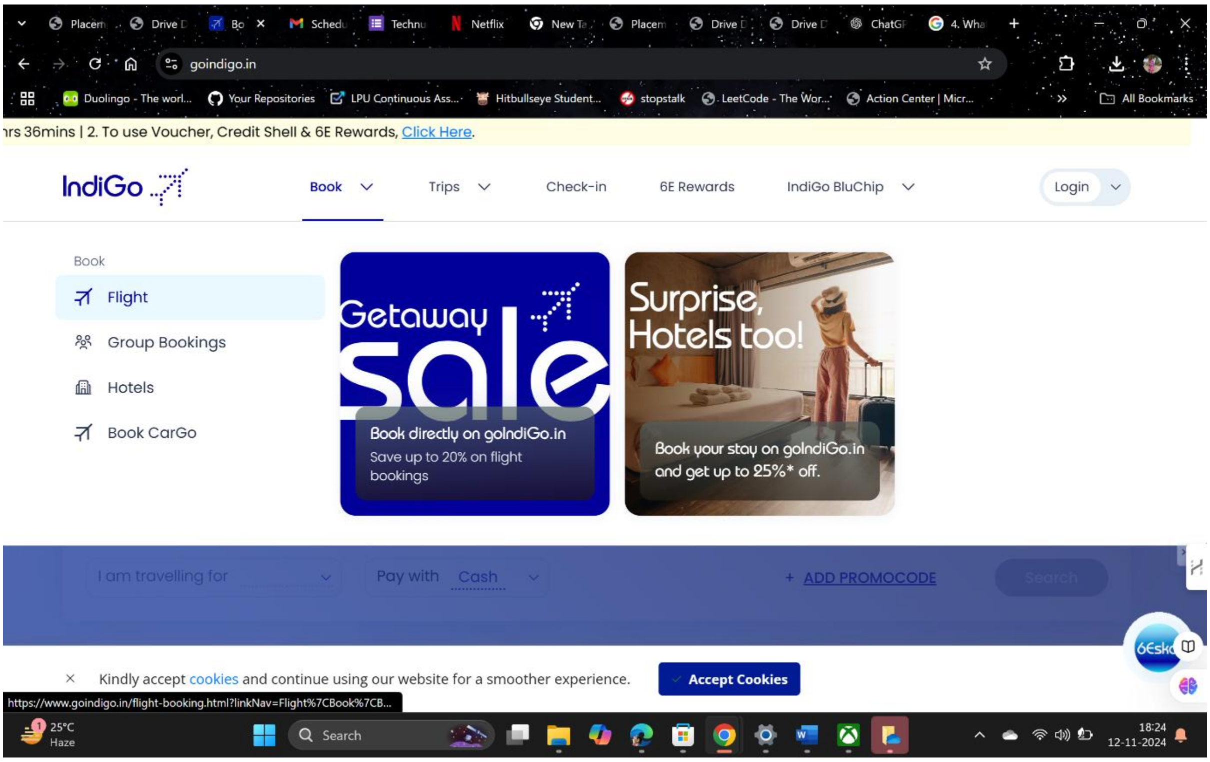
1.Step1. Open the link given in the assignment pdf and go to book



1. Next price drop hour in: 16hrs 44mins | 2. To use Voucher, Credit She

Kindly accept cookies and continue using our website for a smoother experience.

Step2. Click on the flight

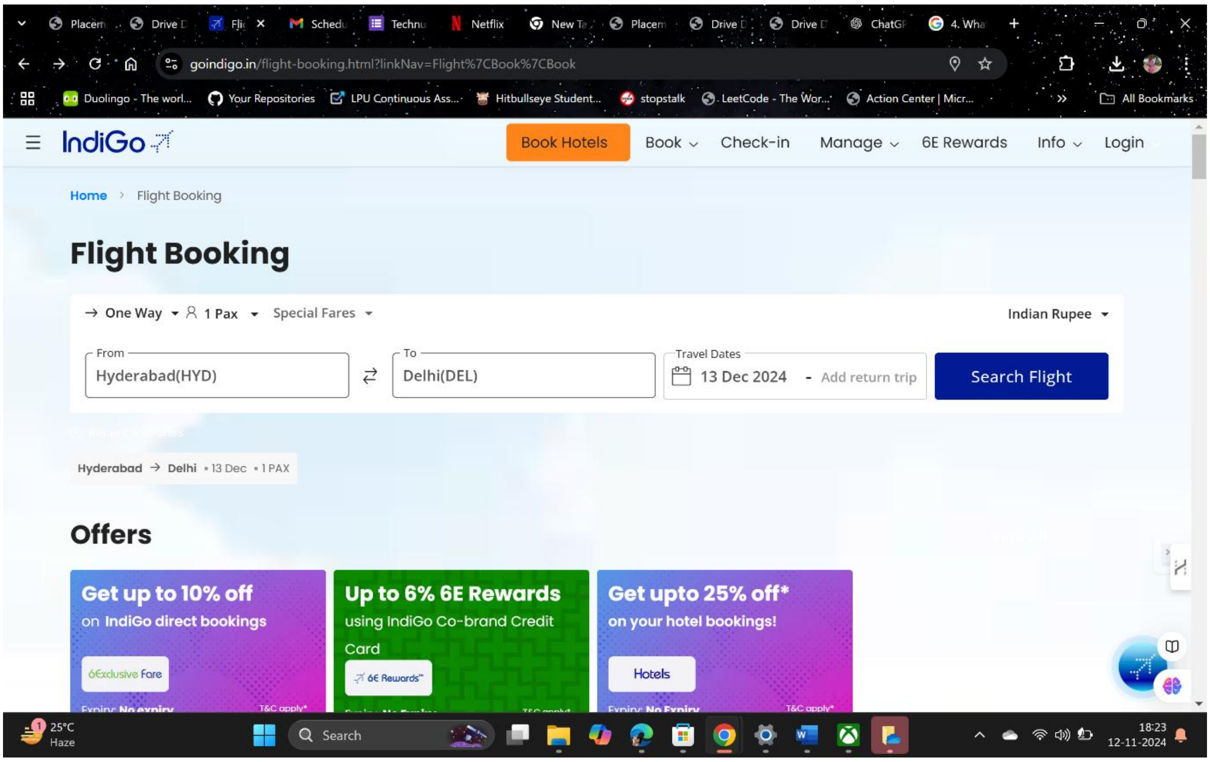


rs 36mins | 2. To use Voucher, Credit Shell & 6E Rewards, Click Here.

Kindly accept cookies and continue using our website for a smoother experience.

https://www.goindigo.in/flight-booking.html?linkNav=Flight%7CBook%7CB...

Step3. Fiil the destination and click on the “Search flight” button



goindigo.in/flight-booking.html?linkNav=Flight%7CBook%7CBook

Flight Booking

One Way · 1 Pax · Special Fares · Indian Rupee

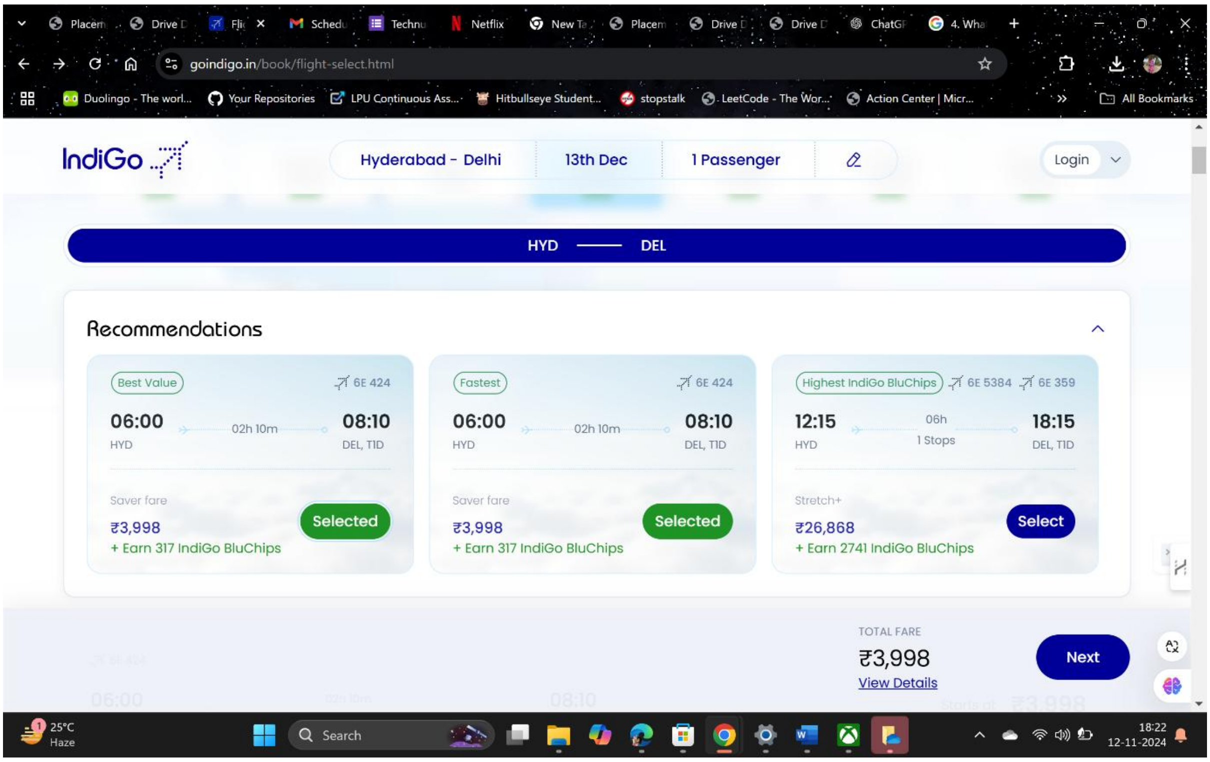
From: Hyderabad(HYD)

To: Delhi(DEL)

Travel Dates: 13 Dec 2024 - Add return trip

Hyderabad → Delhi • 13 Dec • 1 PAX

Step4. Select the first flight



goindigo.in/book/flight-select.html

Hyderabad - Delhi · 13th Dec · 1 Passenger

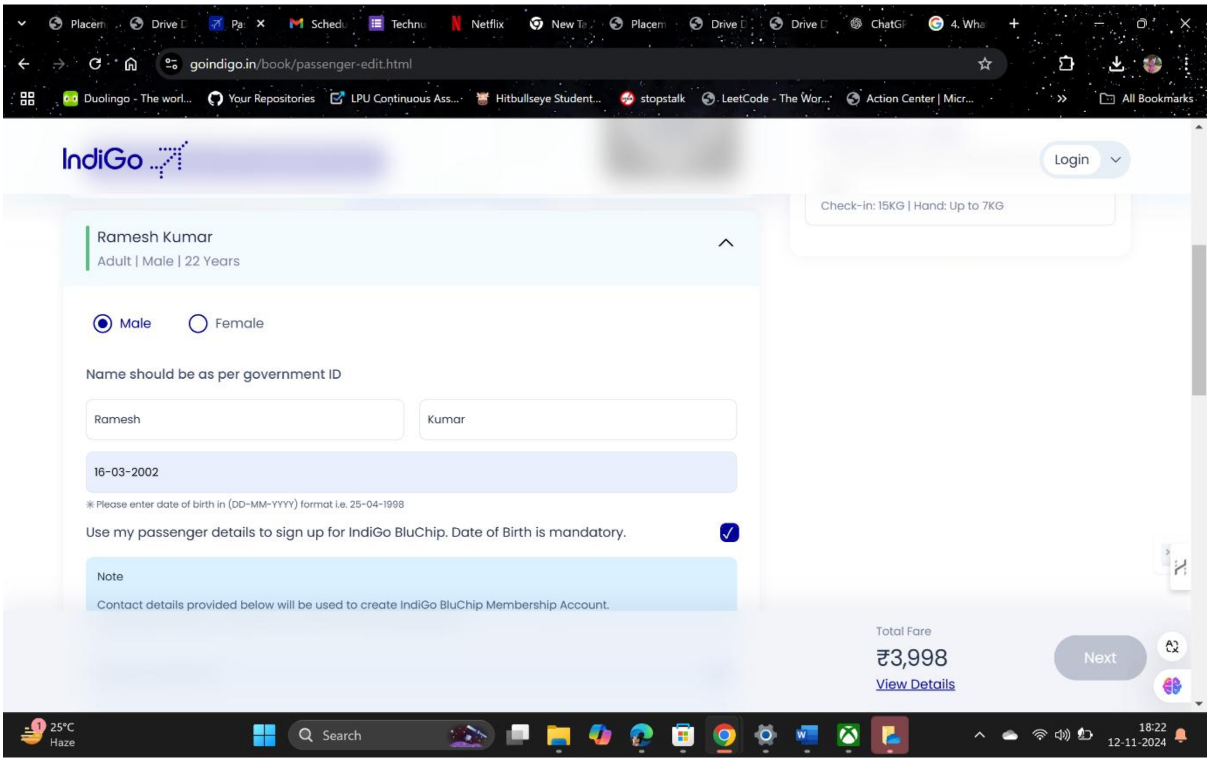
HYD —— DEL

Recommendations

Best Value · 6E 424	Fastest · 6E 424	Highest IndiGo BluChips · 6E 5384 · 6E 359
06:00 HYD · 02h 10m · 08:10 DEL, T1D	06:00 HYD · 02h 10m · 08:10 DEL, T1D	12:15 HYD · 06h · 1 Stops · 18:15 DEL, T1D
Saver fare ₹3,998 + Earn 317 IndiGo BluChips · Selected	Saver fare ₹3,998 + Earn 317 IndiGo BluChips · Selected	Stretch+ ₹26,868 + Earn 2741 IndiGo BluChips · Select

TOTAL FARE ₹3,998

Step5. Fill the passenger details



goindigo.in/book/passenger-edit.html

Check-in: 15KG | Hand: Up to 7KG

Ramesh Kumar
Adult | Male | 22 Years

● Male ○ Female

Name should be as per government ID

Ramesh

Kumar

16-03-2002

⁎ Please enter date of birth in (DD-MM-YYYY) format i.e. 25-04-1998

Use my passenger details to sign up for IndiGo BluChip. Date of Birth is mandatory. ☑

Note

Contact details provided below will be used to create IndiGo BluChip Membership Account.

Total Fare
₹3,998
View Details

Next

Here is the drive link of this procedure in a video format:

https://drive.google.com/file/d/1MDYuf1rCQiN_B1KGcp3YltpAeEi6u_Z_/view?usp=drive_link

SECTION B

2. Choose a case study off our website. Which testing obstacles were addressed, and what creative solutions did we apply to get beyond them? How might you use an equivalent approach in a different testing situation?

- Challenge: Making apps work across devices and browsers

- Apps need to work well on iOS, Android, and web browsers like Chrome, Firefox, and Safari. They provide a variety of tools to help identify issues specific to each platform to test your app. This will ensure that the app looks and works well everywhere. Since it was designed by users, testing is required to make sure everything looks and works correctly in both languages. Some questions. This provides a good experience for English and Arabic users. Case study. They also provide daily and weekly reports that help them track issues and fix them quickly without slowing down the work. Managing all this information using data and 200 bug tests can be confusing. This makes it easier for both the testing team and the customer to understand what's going on, help them quickly, and solve the most important issues.

- ## Testing strategies:

- 1. Cross-Platform Testing: When an app needs to work on different devices (like phones, tablets, or web browsers), it's essential to test it on as many as possible to see how it behaves. Making a list of issues for each device helps focus on the biggest problems and ensures a good experience across all platforms.

- 2. Testing for Multiple Languages or Regions: If your app is used by people who speak different languages or live in different places, it's important to test each version to make sure everything works smoothly. This way, every user has a good experience, no matter where they are or what language they speak.

- 3. Handling Frequent Updates: When updates or new features are added often, quick tests after each change keep things running smoothly. Regular check-ins with the team (like daily or weekly updates) help everyone stay informed and avoid misunderstandings. This keeps quality high even when things change frequently.

- 4. Organizing Large Amounts of Data: For projects with many tests and bugs, a dashboard or tracking tool makes it easy to keep track of everything. It visually shows what's been tested and what issues still need fixing, making the entire process easier to manage and more organized.

3. Review an article or blog on our website that discusses software testing techniques. What important insights did you gain, and how do you think these techniques could be used to enhance a project in the real world?

- ## Key Insights

Automation of Repetitive Tasks

Natural Language Interaction

Improved Test Coverage and Consistency

Enhanced Regression Testing

Global Adaptability

- **Real-World Applications**

1. Cross-Platform and Device Testing: For apps that need to function seamlessly on various devices and operating systems, ChatGPT generates device-specific test cases, promoting consistent user experiences across all platforms.

2. Data-Driven Testing: ChatGPT quickly creates diverse test data—ideal for industries like e-commerce, finance, and healthcare where complex data scenarios are essential.

3. Automated Bug Reporting and Documentation: By generating detailed bug reports, ChatGPT streamlines communication between developers and testers, enhancing efficiency and the feedback process.

4. Test Coverage Analysis: ChatGPT aids in identifying test coverage gaps, ensuring thorough testing and reducing the risk of critical issues in production.

4. Describe a time when you worked on a project or in an academic context and had to deal with a difficult issue related to software testing or development.

1. What lessons did you take away from the event and how did you go about solving it?

- In one of my early academic projects, I was tasked with testing a simple web application that allowed users to sign up and log in. The project was straightforward, but I encountered an issue where the login functionality was not working as expected. Some users could log in successfully, while others could not, and the problem seemed random.
- **How I Solved It:**

- **Understanding the Issue:** I first reproduced the issue by trying to log in with different test accounts. I noted which ones worked and which ones didn't.
- **Testing Different Scenarios:** I decided to test the login process with different types of input, like usernames with special characters or very long passwords, and tried using different browsers. This helped me narrow down that the issue occurred when the password was longer than a certain number of characters.
- **Collaboration with Developers:** After narrowing down the issue, I communicated with the development team, where we discovered that there was a bug in the way the backend was processing long passwords. We worked together to fix the issue.
- **Testing the Fix:** Once the issue was fixed, I wrote additional test cases to ensure that the problem would not reoccur in the future.

  Lessons Learned:
- Patience and Persistence
- Importance of Communication

2. In what ways does this role as an SDET assist you in improving your problem-solving abilities?

- As an SDET, I continuously refine my problem-solving abilities by gaining a deeper understanding of how to test effectively and collaborate with developers. Some ways this role helps me improve my skills include:
  Automation Skills
  Systematic Approach
  Collaboration

5. **What actions do you usually take to find and fix a problem when you encounter an unexpected problem or bug during a project?**

- When encountering an unexpected problem or bug during a project, my approach is systematic and focused on finding the root cause, fixing it, and preventing it from recurring in the future. Here's how I typically handle it:

  Reproduce the Issue: Test the bug under different conditions to understand it.
  Analyze Logs: Check error logs and use debugging tools to pinpoint the issue.
  Isolate the Cause: Break down the code to find where the problem originates.
  Collaborate: Discuss with the team if the issue is complex.
  Fix and Test: Apply the fix and thoroughly test to ensure the bug is resolved.

1. **How can you make sure that problems like these never happen again?**

- To Prevent Future Issues:

- Root Cause Analysis: Understand why it happened and address weaknesses.
- Write Test Cases: Add tests to catch similar bugs in the future.
- Code Reviews: Use peer reviews to catch issues early.
- Automated Testing: Implement tests to verify core features continuously.
- Monitoring: Ensure proper logging to detect issues quickly.
- Post-Mortem: Review the issue with the team to improve future processes.

6. **Consider a project where you successfully used your technical skills to achieve a particular objective.**
   1. **Which particular skills did you bring to the table, and how did they contribute to the project's success?**
      - I applied skills in data analysis, problem-solving, and development (e.g., Python, JavaScript) to automate processes, improve efficiency, and provide actionable insights. My ability to communicate complex technical concepts to non-technical stakeholders also helped align the team, contributing to the project's success.
   2. **On the flip side, outline an obstacle you faced as a result of your inexperience and how you want to get better at it going forward.**
      - I struggled with time management due to inexperience in balancing tasks, which led to delays. To improve, I plan to use project management tools, break tasks into smaller chunks, and better estimate time requirements, seeking mentorship to enhance my planning skills.

7. **Write about a time you worked on a software project as a team member. What difficulties did the team face, and what part did you play in getting them over?**
   - In a software project, the team struggled with syncing the frontend and backend, leading to bugs and data not displaying correctly. As a beginner, I helped by testing the integration between both parts and identifying where the issue was. I created simple test cases, reported the problems, and worked with the team to fix the data flow. My contribution helped the team resolve the issue and move the project forward.
8. **What motivates you to pursue a career in software testing, and how do you plan to stay engaged and continuously improve in this field?**
   - I'm motivated by the problem-solving aspect of software testing and the satisfaction of ensuring software works well. As a beginner, I plan to improve by learning new testing tools, reading industry blogs, and

practicing test cases. I also want to explore automation testing to stay updated and grow my skills.

9. **How will the growth of DevOps and cloud computing affect software testing? What are the main benefits and challenges of implementing these technologies in testing?**
    - DevOps and cloud computing will make software testing faster and more scalable by enabling automation and testing across different environments. The benefits include quicker testing and easier scalability, while the challenges are managing complexity and ensuring security in the cloud. These technologies will improve efficiency but require new tools and skills.

REFERENCE : CHATGPT, https://www.frugaltesting.com/blog/how-chatgpt-enhances-automated-testing-a-guide-with-real-world-use-cases

10. Please share the following details in your profile:

1. LinkedIn Profile Link : www.linkedin.com/in/anamika-pandey76

2. Resume (PDF file) :
https://drive.google.com/file/d/12xnmVBpyEHNGIGc2S7hyPbsHJA-phZ2J/view?usp=drive_link

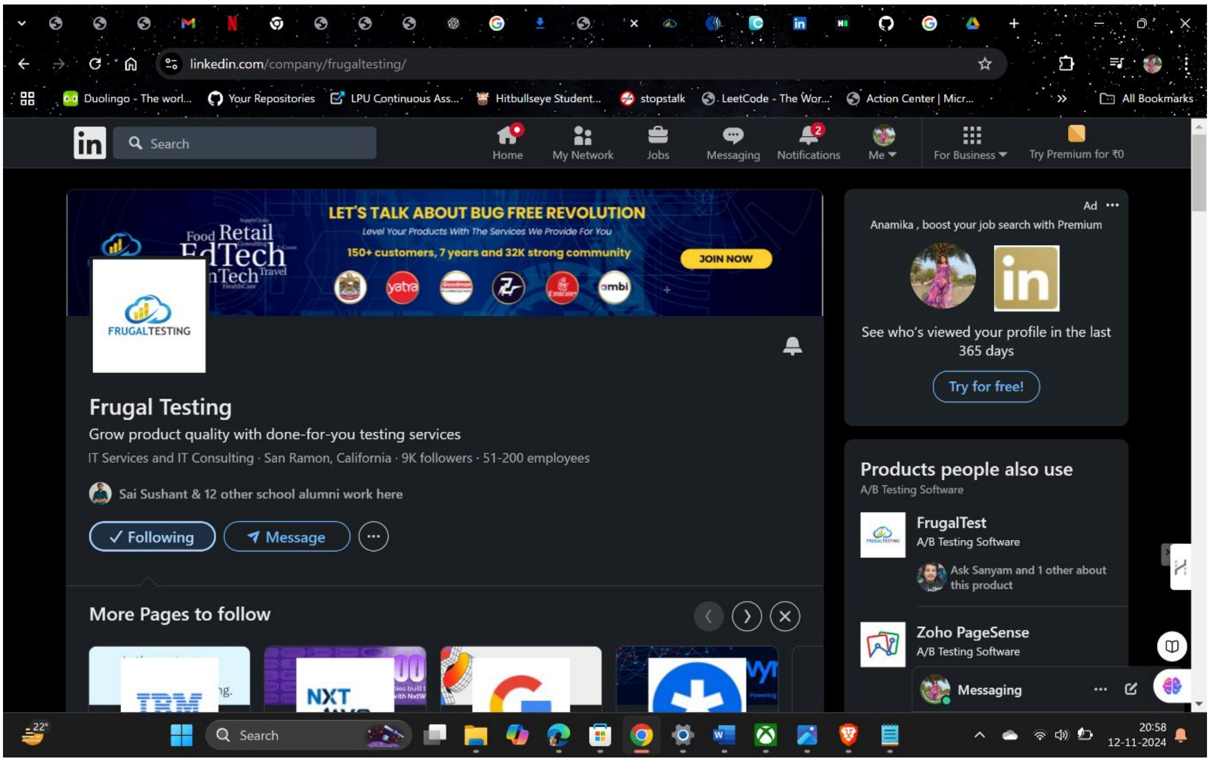
3. Any other Technical profiles like :

Hackerrank: https://www.hackerrank.com/profile/anupandey1177

Github: https://github.com/Anamika4260

3. Any other Project links about which you are really proud of?
https://github.com/Anamika4260/CRM-Kalvium

11. **Visit and follow the Frugal Testing page on all the social media platforms. Like/Share the posts and also share your views on the posts in the comment section.**



linkedin.com/company/frugaltesting/

Frugal Testing

Grow product quality with done-for-you testing services

IT Services and IT Consulting · San Ramon, California · 9K followers · 51-200 employees

Sai Sushant & 12 other school alumni work here

Anamika , boost your job search with Premium

See who's viewed your profile in the last 365 days

Products people also use

A/B Testing Software

FrugalTest

A/B Testing Software

Ask Sanyam and 1 other about this product

Zoho PageSense

A/B Testing Software

12. . How do you use ChatGPT to solve problems in your daily work?

- I use ChatGPT in my daily work to assist with troubleshooting issues, generate test cases, and clarify technical concepts. It helps me by suggesting solutions to coding problems, explaining difficult topics, and offering examples of how to implement certain features. I also use it to draft reports, review best practices, or brainstorm ideas for improving processes. It's a great tool for quickly gathering information, staying updated on new technologies, and improving my overall productivity.

13. What are the most effective ways to ensure quality in a software project?

- To ensure quality in a software project, start by making sure the project has clear requirements. Use simple tests to check if the code works as expected, and automate repetitive tests to save time. Code reviews help keep the code clean and error-free. Set up continuous testing to catch issues early, and encourage good communication between team members. Also, test the software's performance and get feedback from users to make sure it meets their needs.

14. How can ChatGPT help in improving the testing process?

- ChatGPT can make the testing process easier by helping create test cases, writing automation scripts, and drafting detailed test plans. It can also help organize bug reports and pinpoint important areas to test. By analyzing test results, ChatGPT suggests improvements and ensures all parts of the system are thoroughly tested. Plus, it helps with documenting and summarizing test outcomes, making the process more efficient. This allows testers to focus on what matters most and improves the overall workflow.

15. **What does exploratory testing mean, and how would you explain it using a daily life example?**
    - Exploratory testing is like using a new app or gadget by experimenting with it, rather than following a set of instructions. You explore different features and behaviors, discovering bugs or issues along the way through hands-on interaction.
16. **What is the purpose of a bug report, and how would you explain it to someone who's not in tech?**
    - A bug report is like a note you leave to let someone know that something isn't working as it should. It explains the problem, how to reproduce it, and what went wrong, so the team can fix it. It's like telling a mechanic what's wrong with your car so they can repair it properly.
17. **How can ChatGPT or similar AI tools enhance test case generation, bug reporting, or test result analysis?**
    - ChatGPT or similar AI tools can make testing easier by quickly generating test cases based on software features, helping write clear bug reports, and analyzing test results to find patterns or suggest improvements. They can save time by automating repetitive tasks, so testers can focus on more important work, like improving the software.
18. **How do you decide which test cases to automate and which to leave manual?**
    - Test cases that are repetitive, time-consuming, or need to be run frequently (like login or data entry tests) are ideal for automation, as they save time and reduce errors. More complex, one-off, or exploratory tests, where human intuition is important, are usually better done manually.
19. **How would you explain the importance of regression testing to a non-technical person?**
    - Regression testing is like double-checking that new changes or fixes in a product haven't accidentally broken anything that was already working well. It's similar to making sure that fixing one part of a car hasn't messed up other parts, so everything still runs smoothly.
20. **Please write an article on any ONE of the below topics.**
    - **The Impact of AI on Software Testing: Opportunities and** Challenges

Artificial intelligence is changing the way we do software testing, it's faster and more efficient. A big advantage is the automation of test case generation. Instead of creating tests manually, AI can generate a wide variety of tests based on user behavior patterns and past data, saving time and helping testers catch more bugs. Another advantage is predictive analysis for error detection. By analyzing previous error patterns, AI can predict where problems are likely to occur, helping testers focus on areas that need the most attention and improve overall quality.

AI also helps with test maintenance. In traditional testing, software updates can break test scripts and manual correction can take time. AI tools can detect these changes and adjust tests automatically, so testers don't have to. For example, if the site layout changes, the AI tool can detect and update the test to match the new layout. In addition, AI can facilitate regression testing by selecting only the most relevant tests, focusing on the parts of the software that have changed, so testing is faster and still thorough.

However, the use of artificial intelligence in testing also has its pitfalls. AI needs a lot of quality data to work well, which can be hard to get, especially for newer projects without a lot of historical data. There is also the issue of transparency; sometimes it's hard to understand how the AI makes certain decisions, which can make testers unsure whether to trust the results. Also, the cost and setup can be high, as AI tools need special software, training and infrastructure that is not always available, especially for smaller teams.

Finally, there is a shortage of skilled professionals who understand both AI and software testing, making it difficult for some companies to fully implement AI into their testing process. The best approach currently seems to be a mix of AI and human testing – using AI for repetitive tasks and data-intensive analytics while letting testers focus on creative and complex testing. This way we get the best of both worlds: the speed and accuracy of artificial intelligence combined with the critical thinking and intuition of human testers.

REFERENCE : CHATGPT , BLACKBOXAI, frugal testing , software testing websites,