Absolutely! Here's the slightly more elaborated version of each response—still under 75 seconds each so they fit into your video interview smoothly, while adding just a little more storytelling, impact, and reflection.

---

1. Ethical Dilemma

"In our DBMS mini-project on a Parking Lot Management System, one of my teammates proposed copying the ER diagram from a similar project online since we were running behind. I felt uncomfortable with it because even though it seemed minor, it was still plagiarism. I brought it up in our group call and explained how it could hurt our learning and possibly get us flagged for copying. After a discussion, we agreed to create our own diagram based on real requirements we defined. It took more time, but it was original and we learned how to structure a system from scratch. This reinforced the importance of standing up for academic integrity, even in a team setting."

---

2. Explaining Technical Info to a Non-Technical Person

"During our project on predicting student performance using machine learning, I had to explain the core model to a mentor from a non-technical background. Instead of going into technical terms like logistic regression or feature scaling, I said—'Imagine a teacher who sees student behavior like attendance, assignments, and past marks, and based on that, tries to predict their final grade.' I also used visuals and avoided code talk. By relating it to real-life examples, I could explain the model in a way that made sense to them. It helped me improve my own understanding too."

---

3. Why You're a Good Fit

"Being a 2nd-year Data Science student, I've built a strong foundation in coding, problem-solving, and working with data. I've worked on projects like the Parking Lot Management System using DBMS and a small ML-based Student Performance Predictor. I enjoy learning new tools, recently picked up GitHub, and regularly take part in workshops and hackathons. I'm good at working in teams, adapting quickly, and approaching problems with a learner's mindset. I believe these qualities make me a good fit for this role where both technical curiosity and teamwork are valued."

---

4. Collaborating with Multiple People

"In our DBMS project, we built a Parking Lot Management System as a team of four. I was responsible for database design and writing queries, while others handled the UI and logic. We faced initial coordination issues, so I suggested dividing tasks using a Trello board and scheduling brief calls every evening. This helped us stay in sync. One major challenge was connecting the backend with our database, but we debugged it together by testing one module at a time. Our final project was appreciated for its smooth functionality, and I realized how important regular communication is in group work."

---

5. Steps Taken to Learn About the Program

"I started by going through the official website to understand the structure, goals, and opportunities of the program. Then I looked up testimonials on LinkedIn and read experiences shared by past participants. I also reached out to a senior who was selected last year—she gave me insights into the kind of projects expected and how to prepare.

Additionally, I followed related webinars and student communities on Instagram to stay informed. These steps helped me understand how I could align my skills with what the program looks for."

---

6. Presenting Complex Info to Someone With Limited Knowledge

"During a Python basics workshop for juniors, I had to explain concepts like loops and functions to students with no programming experience. Instead of jumping into code, I compared loops to real-life routines—like brushing teeth or making tea every day. I used Google Colab to give live examples, and kept checking in to make sure they were following. Many of them said the analogies helped them understand better and gave them the confidence to start coding. It showed me how powerful simple explanations can be."

---

7. Ethical Challenge in School

"During an online exam in my first year, a classmate asked if I could send him answers over WhatsApp. I felt conflicted because he was a close friend, but I knew it would be wrong to help cheat. I politely declined and explained the risks—both ethical and academic. After the exam, I offered to help him understand the topic better. He appreciated it and started joining our study sessions. It was a tough call, but it reminded me that doing the right thing might feel uncomfortable in the moment, but it always pays off in the long run."

---

8. Solving a Project With Missing Information

“While working on a Smart Water Usage Analysis project, we couldn’t get real data from our campus. I proposed we use open datasets from Kaggle and simulate realistic values for the local context. We documented all assumptions clearly and adapted our analysis accordingly. We also noted how the results might differ with actual data. This approach helped us stay on track while being honest about limitations. Our faculty appreciated our effort to work around the issue without compromising on transparency.”

---

9. Identifying and Solving a Root Cause

“In our DBMS project, we had a bug where user data wasn’t updating in the system. At first, we thought it was an issue with the frontend, but after testing each module, I found the real problem—a missing commit statement in our Python backend. Without it, changes weren’t getting saved to the database. Once I fixed it, the issue was resolved and everything worked fine. That experience taught me how important it is to break down a problem and check each component step by step.”

---

10. Learning a New Skill & Facing Challenges

“I started learning Python through online courses and coding platforms. Initially, it was tough—I’d get syntax errors, and logic building didn’t come naturally. But I kept practicing on HackerRank, joined a peer coding group, and watched videos to understand how others solve problems. I also built small projects like a calculator and quiz app, which gave me confidence. It took effort and patience, but now I feel comfortable using Python in my coursework and mini-projects.”

---

11. Resolving Conflict With a Colleague

“During a college fest, I was on the design team and had a disagreement with another member about the color scheme. Rather than arguing, I asked her to present her version and showed mine too. We realized both ideas had strengths, so we merged her layout with my color choices. The final design looked great, and we got good feedback. That experience taught me that being open-minded and collaborative helps resolve conflicts and leads to better results.”

---

12. Situation With Poor Communication

“In our Python workshop team, I once assumed everyone understood the plan. One member didn’t complete his part because I hadn’t clearly explained his task. It caused last-minute stress. I immediately called a team meeting, clarified roles, and made a checklist to track progress. Since then, I always confirm understanding in group work, and follow up with written notes or messages to avoid miscommunication.”

---

13. Decision Made Without Full Info

“During a 24-hour hackathon, we chose the theme ‘Education’ without checking if enough data was available. Halfway through, we realized we couldn’t access real student data. I suggested narrowing the scope to analyzing quiz feedback using text sentiment, and we built a basic analyzer using open feedback samples. It turned out well, and we got appreciation for creativity. I learned that you can’t always wait for perfect info, but you can adapt and still deliver something impactful.”

---

14. Almost Missed Deadline

“In the second semester, I had a math exam and our DBMS project submission in the same week. I misjudged how long the documentation would take. With just one day left, I listed out pending sections, asked my teammate to handle screenshots while I wrote the schema and queries part. We worked late but submitted just 5 minutes before the deadline. That experience pushed me to start using reminders and to-do apps so I can manage time better going forward.”

Would you like these as a downloadable doc, or maybe a mobile-friendly PDF for practice before recording?