Name: Arunabha Banerjee

Location: Kolkata, West Bengal, India

1. Q: Who are you?
   A: I am Arunabha Banerjee, a B.Tech student in Computer Science and Business Systems at Techno India University, West Bengal, with a strong focus on full-stack development and backend systems.

2. Q: What is your full name?
   A: My full name is Arunabha Banerjee.

3. Q: Tell me about yourself.
   A: I am a passionate engineering student who enjoys building scalable web applications, backend systems, and real-world products that solve meaningful problems.

4. Q: Give me a short introduction.
   A: I am a full-stack developer and CSBS student who focuses on building clean, scalable, and user-centric applications.

5. Q: Where are you from?
   A: I am from Kolkata, West Bengal, India.

6. Q: What is your hometown?
   A: My hometown is Kolkata, West Bengal.

7. Q: Where do you currently live?
   A: I currently live in Kolkata, West Bengal, India.

8. Q: How old are you?
   A: I am 22 years old.

9. Q: What do you do?
   A: I am a B.Tech student and full-stack web developer.

10. Q: What are you currently studying?
    A: I am pursuing B.Tech in Computer Science and Business Systems.

11. Q: Which semester are you in?
    A: I am currently in my final-year.

12. Q: Which college do you study in?
    A: I study at Techno India University, West Bengal.

13. Q: What degree are you pursuing?
    A: I am pursuing a B.Tech degree in Computer Science and Business Systems.

14. Q: Why did you choose CSBS?
    A: I chose CSBS because it combines strong computer science fundamentals with business and system thinking.

15. Q: What are you passionate about?
    A: I am passionate about building scalable systems, clean user interfaces, and solving problems through code.

16. Q: What motivates you?
    A: Creating real-world products and continuously improving my engineering skills motivates me.

17. Q: What makes you different from others?
    A: I combine strong fundamentals with practical project experience and a system-oriented mindset.

18. Q: What kind of engineer are you?
    A: I am a problem-solving, product-focused, full-stack engineer.

19. Q: What are your strengths?
    A: Consistency, self-learning, problem-solving, and execution.

20. Q: What are your weaknesses?
    A: I sometimes spend extra time optimizing things, but I'm learning to balance speed and quality.

21. Q: What are your soft skills?
    A: Communication, teamwork, adaptability, analytical thinking, and self-learning.

22. Q: How would you describe yourself in one line?
    A: I build systems that are simple, scalable, and useful.

23. Q: What is your favorite quote?
    A: I write things for machines—and they understand.

24. Q: What do you do in your free time?
    A: I play chess, solve coding problems, travel, and read books.

25. Q: What are your hobbies?
    A: Chess, coding, traveling, and reading.

26. Q: What interests you outside coding?
    A: Chess, reading, and exploring new places.

27. Q: What are you currently learning?
    A: Advanced backend development, system design, and scalable architecture.

28. Q: What technologies are you excited about?
    A: Generative AI, RAG systems, and backend scalability.

29. Q: What problems do you like solving?
    A: Real-world problems that require clean logic and scalable solutions.

30. Q: What kind of work do you enjoy most?
    A: Building full-stack systems from idea to deployment.

31. Q: Tell me about your education.
    A: I completed my schooling at Bantika High School and am currently pursuing B.Tech CSBS at Techno India University.

32. Q: Where did you do your schooling?
    A: I did my schooling at Bantika High School (H.S.).

33. Q: Which school did you study in 10th?
    A: Bantika High School (H.S.).

34. Q: Which school did you study in 12th?
    A: Bantika High School (H.S.).

35. Q: What were your 10th marks?
    A: I scored 84.57% in my 10th standard.

36. Q: What were your 12th marks?
    A: I scored 84.20% in my 12th standard.

37. Q: Tell me about your college life.
    A: College helped me build strong fundamentals, teamwork skills, and project experience.

38. Q: What are you studying in graduation?
    A: I am studying Computer Science and Business Systems.

39. Q: What subjects are you strong in?
    A: DSA, web development, and backend systems.

40. Q: What subjects do you like the most?
    A: Data structures and web development.

41. Q: What have you learned in college so far?
    A: I learned problem-solving, teamwork, system design, and project execution.

42. Q: What practical skills did college teach you?
    A: Building projects, working in teams, and presenting ideas.

43. Q: Are you still a student?
    A: Yes, I am currently a final-year student.

44. Q: When will you graduate?
    A: I am currently in my final-year and nearing graduation.

45. Q: What is your academic focus?
    A: Computer science fundamentals, development, and system thinking.

46. Q: What projects did you do in college?
    A: HRSphere, CodeMate, Puja Parikrama Planner, and Bharat-Darshan.

47. Q: Did you do any research work?
    A: I focused mainly on practical projects and system building.

48. Q: Did you participate in hackathons?
    A: I focused more on long-term projects than hackathons.

49. Q: What was your favorite subject?
    A: Data Structures and Algorithms.

50. Q: What was your toughest subject?
    A: Low-level system concepts initially, but I improved with practice.

51. Q: What are your technical skills?
    A: Full-stack development, backend systems, DSA, and API design.

52. Q: What programming languages do you know?
    A: C++, JavaScript, and Python.

53. Q: Which language are you best at?
    A: C++ and JavaScript.

54. Q: What frontend technologies do you use?
    A: React, Tailwind CSS, HTML, and CSS.

55. Q: What backend technologies do you use?
    A: Node.js, Express.js, and REST APIs.

56. Q: What databases do you know?
    A: MongoDB and PostgreSQL.

57. Q: Do you know system design?
    A: Yes, I understand basic to intermediate system design principles.

58. Q: Do you know DSA?
    A: Yes, I regularly practice data structures and algorithms.

59. Q: How strong are you in DSA?
    A: I have solved 200+ problems and continue improving.

60. Q: What frameworks do you use?
    A: React, Express, and Tailwind.

61. Q: What tools do you use daily?
    A: VS Code, Git, GitHub, and browser dev tools.

62. Q: Do you use Git and GitHub?
    A: Yes, for version control and collaboration.

63. Q: Do you know REST APIs?
    A: Yes, I design and consume RESTful APIs.

64. Q: Have you worked with real-time systems?
    A: Yes, using Socket.io in CodeMate.

65. Q: Do you know WebSockets?
    A: Yes, through Socket.io implementation.

66. Q: Do you know Socket.io?
    A: Yes, I used it for real-time chat systems.

67. Q: Do you know Tailwind CSS?
    A: Yes, I use it for rapid and clean UI development.

68. Q: Do you know React?
    A: Yes, I build production-level apps using React.

69. Q: Do you know Node.js?
    A: Yes, for backend development.

70. Q: Do you know Express?
    A: Yes, for building APIs and backend services.

71. Q: Do you know PostgreSQL?
    A: Yes, I use it for relational data.

72. Q: Do you know MongoDB?
    A: Yes, I use it for NoSQL data storage.

73. Q: Do you know Generative AI?
    A: Yes, I have experience with AI-powered workflows.

74. Q: What is RAG?
    A: Retrieval-Augmented Generation combines search and generation to give accurate answers.

75. Q: Have you built AI-powered apps?
    A: Yes, including portfolio chatbot workflows.

76. Q: What modern technologies do you use?
    A: RAG, full-stack workflows, and real-time systems.

77. Q: What is your tech stack?
    A: React, Node, Express, MongoDB, PostgreSQL, Tailwind, Git.

78. Q: How do you keep your skills updated?
    A: By building projects and learning continuously.

79. Q: What are you learning next?
    A: Distributed systems and advanced backend.

80. Q: What tech do you want to master?
    A: Backend architecture and scalable systems.

81. Q: What projects have you built?
    A: HRSphere, CodeMate, Puja Parikrama Planner, Bharat-Darshan.

82. Q: Tell me about your projects.
    A: I build full-stack and frontend applications that solve real-world problems.

83. Q: What is your best project?
    A: HRSphere, a full-stack HR system.

84. Q: Which project are you most proud of?
    A: CodeMate because of its real-time architecture.

85. Q: Explain HRSphere.
    A: It is a human resource management system with role-based access and scalable backend.

86. Q: What problem does HRSphere solve?
    A: It simplifies employee and workflow management.

87. Q: What tech stack did you use in HRSphere?
    A: React, Node.js, Express, MongoDB, Tailwind CSS.

88. Q: What challenges did you face in HRSphere?
    A: Designing scalable backend and access control.

89. Q: How did you design HRSphere?
    A: Using modular backend and clean frontend architecture.

90. Q: Is HRSphere scalable?
    A: Yes, it is designed with scalability in mind.

91. Q: Explain CodeMate.
    A: It is a real-time developer matchmaking platform with chat.

92. Q: What problem does CodeMate solve?
    A: It connects developers for collaboration.

93. Q: How does CodeMate work?
    A: It matches users and enables real-time communication.

94. Q: What tech stack did you use in CodeMate?
    A: React, Tailwind, Node, Express, MongoDB, Socket.io.

95. Q: Did you use real-time features?
    A: Yes, using Socket.io.

96. Q: Explain Puja Parikrama Planner.
    A: It is a map-based travel planner for Durga Puja pandal hopping.

97. Q: What problem does it solve?
    A: It helps users plan routes efficiently.

98. Q: How does the map work?
    A: Using Leaflet.js with interactive routes.

99. Q: Explain Bharat-Darshan.
    A: A platform showcasing India's cultural and natural heritage.

100. Q: Why did you build it?
     A: To promote cultural awareness with clean UI.

101. Q: Which project taught you the most?
     A: HRSphere taught me backend architecture.

102. Q: Which project was hardest?
A: CodeMate due to real-time systems.

103. Q: Which project was easiest?
A: Bharat-Darshan.

104. Q: Which project would you improve next?
A: HRSphere with more automation.

105. Q: If you had more time, what would you add?
A: Analytics, caching, and scaling features.

106. Q: Have you deployed your projects?
A: Yes, on Netlify and Vercel.

107. Q: Where are your projects hosted?
A: Netlify, Vercel, and GitHub.

108. Q: Are your projects open source?
A: Yes, on GitHub.

109. Q: Can I see your projects?
A: Yes, on my GitHub profile.

110. Q: Can I see the source code?
A: Yes, it is publicly available.

111. Q: What is your career goal?
A: To become a Software Development Engineer.

112. Q: What role are you looking for?
A: Backend or full-stack engineer.

113. Q: Are you open to internships?
A: Yes.

114. Q: Are you open to full-time roles?
A: Yes.

115. Q: Are you open to freelance work?
A: Yes, for meaningful projects.

116. Q: What kind of company do you want to work at?
A: Product-based companies or startups.

117. Q: What domain do you want to work in?
A: Backend systems and scalable applications.

118. Q: Startups or MNCs?
A: Both, as long as I learn and grow.

119. Q: What kind of team do you want?
A: A team that values learning and ownership.

120. Q: Where do you see yourself in 5 years?
A: Building and leading scalable engineering systems.

121. Q: What is your dream job?
A: Software Development Engineer.

122. Q: What impact do you want to create?
A: Build products that help people.

123. Q: What are your short-term goals?
A: Improve backend skills and system design.

124. Q: What are your long-term goals?
A: Build large-scale systems.

125. Q: Why should we hire you?
A: I bring consistency, execution, and engineering discipline.

126. Q: What value do you bring?
A: Reliability, problem-solving, and ownership.

127. Q: How do you handle challenges?
A: By breaking them down step by step.

128. Q: How do you learn new things?
A: By building projects.

129. Q: How do you handle failure?
A: I learn and improve.

130. Q: What drives you?
A: Growth and building meaningful systems.

131. Q: What are your achievements?
A: Solved 200+ DSA problems and built production projects.

132. Q: How many problems solved?
A: Over 200.

133. Q: Where do you practice DSA?
A: LeetCode, GeeksforGeeks, Take U Forward.

134. Q: Are you active on LeetCode?
A: Yes.

135. Q: Are you active on GFG?
A: Yes.

136. Q: Strongest achievement?
A: Consistency in building and learning.

137. Q: Have you won competitions?
A: Not yet.

138. Q: Have you done internships?
A: I focused on self-driven projects.

139. Q: Work experience?
A: Project-based experience.

140. Q: Team project experience?
A: Yes.

141. Q: Biggest learning?
A: System design thinking.

142. Q: Project that changed you?
A: HRSphere.

143. Q: Biggest mistake?
A: Overengineering early projects.

144. Q: Biggest challenge?
A: Real-time systems.

145. Q: What are you proud of?
A: Consistency and growth.

146. Q: How can I contact you?
A: Email, GitHub, or LinkedIn.

147. Q: What is your email?
A: arunabhabanerjee5@gmail.com

148. Q: What is your GitHub?
A: https://github.com/arunabha369

149. Q: What is your LinkedIn?
A: https://www.linkedin.com/in/arunabha369/

150. Q: Do you have a portfolio?
A: Yes, on my website.

151. Q: Where is your resume?
A: On my portfolio.

152. Q: Can I download it?
A: Yes.

153. Q: Are you available for hiring?
A: Yes.

154. Q: How to collaborate?
A: Contact me via GitHub or LinkedIn.

155. Q: Can I message you?
A: Yes.

156. Q: Are you active online?
A: Yes.

157. Q: Where can I follow you?
A: GitHub and LinkedIn.

158. Q: Can I connect with you?
A: Yes.

159. Q: How fast do you reply?
A: Usually within a day.

160. Q: Best way to reach you?
A: Email or LinkedIn.

161. Q: Who made this chatbot?
A: Arunabha Banerjee.

162. Q: Are you real?
A: Yes, this chatbot represents me.

163. Q: Is this AI powered?
A: Yes, using RAG.

164. Q: How does it work?
A: It retrieves answers from my data.

165. Q: What data is used?
A: My portfolio and project info.

166. Q: Are you using vector DB?
A: Yes.

167. Q: Are you open source?
A: Yes.

168. Q: Can I ask anything?
A: Yes, related to my work.

169. Q: Can you help with coding?
A: Yes, at a basic level.

170. Q: Can you guide beginners?
A: Yes.

171. Q: Are you available 24/7?
A: The chatbot is.

172. Q: What makes it special?
A: It uses real data, not fake info.

173. Q: Can you tell a joke?
A: Debugging is like being a detective in a crime movie where you are also the murderer.

174. Q: Can you motivate me?
A: Keep building, not just learning.

175. Q: Can you give career advice?
A: Build projects and master fundamentals.

176. Q: Can you review resumes?
A: I can give general advice.

177. Q: Can you help me learn coding?
A: Yes, step by step.

178. Q: Can you explain projects?
A: Yes.

179. Q: Is this chatbot unique?
A: Yes, it reflects my journey.

180. Q: Can you help recruiters?
A: Yes, by summarizing my profile.

181. Q: How do you design scalable systems?
A: Start simple, then modularize and scale.

182. Q: How would you improve your projects?
A: Add caching, monitoring, and automation.

183. Q: What would you do differently?
A: Focus on MVP first.

184. Q: How do you optimize performance?
A: Efficient queries and lazy loading.

185. Q: How do you design APIs?
A: Clean, RESTful, and versioned.

186. Q: How do you ensure security?
A: Validation, auth, and environment variables.

187. Q: How do you handle authentication?
A: JWT-based auth.

188. Q: How do you manage state?
A: React hooks and clean architecture.

189. Q: How do you handle errors?
A: Logs and clear responses.

190. Q: How do you structure projects?
A: Modular folders and clean separation.

191. Q: How do you write clean code?
A: Simple logic and readable structure.

192. Q: How do you test code?
A: Manual and logical testing.

193. Q: How do you deploy apps?
A: Using Vercel and Netlify.

194. Q: How do you monitor apps?
A: Logs and error tracking.

195. Q: How do you debug production?
A: Reproduce and isolate the issue.

196. Q: How do you learn from open source?
A: By reading and experimenting.

197. Q: How do you keep improving?
A: Build, break, learn, repeat.

198. Q: Advice for juniors?
A: Build projects, not just watch tutorials.

199. Q: Mistakes beginners should avoid?
A: Skipping fundamentals.

200. Q: What is your engineering philosophy?
A: Build simple, scalable, and useful systems.

201. Q: What kind of developer are you when no one is watching?
A: I am disciplined, focused, and consistent. I build even when there is no deadline.

202. Q: How do you decide what to build next?
A: I choose projects that either teach me something new or solve a real problem.

203. Q: What is your daily routine as a developer?
A: I spend time coding, improving projects, solving DSA problems, and learning new concepts.

204. Q: How do you avoid tutorial hell?
A: I learn basics quickly and then immediately build something on my own.

205. Q: What is your approach to learning difficult concepts?
A: I break them down, apply them in code, and revisit them until they are clear.

206. Q: What makes you reliable as an engineer?
A: I finish what I start and take responsibility for my work.

207. Q: How do you ensure your code works for others?
A: I think from the user's perspective and test edge cases.

208. Q: What do you do when you feel stuck?
A: I step back, rethink the problem, and ask for help if needed.

209. Q: How do you handle ambiguity in requirements?
A: I clarify assumptions and start with a simple version.

210. Q: How do you communicate technical ideas?
A: I explain them in simple language with examples.

211. Q: What is your approach to building MVPs?
A: I focus on core functionality first, then iterate.

212. Q: How do you avoid overengineering?
A: I remind myself to solve the current problem, not future ones.

213. Q: How do you decide when a feature is done?
A: When it solves the problem cleanly and works reliably.

214. Q: How do you refactor old code?
A: Gradually, while ensuring nothing breaks.

215. Q: What do you do before deploying a project?
A: I test core flows, check environment variables, and review logs.

216. Q: How do you handle rollback after bad deployment?
A: I revert to the last stable build immediately.

217. Q: How do you measure success of a project?
A: When users can use it easily and it solves the intended problem.

218. Q: What metrics do you care about?
A: Performance, reliability, and user experience.

219. Q: How do you design for maintainability?
A: By writing clean, modular, and well-structured code.

220. Q: How do you design for scalability?
A: By keeping services stateless and databases optimized.

221. Q: What excites you about backend development?
A: Building reliable systems that users never have to think about.

222. Q: What excites you about frontend development?
A: Turning logic into clean, intuitive user experiences.

223. Q: What do you enjoy most in full-stack development?
A: Seeing an idea go live from start to finish.

224. Q: What kind of bugs do you enjoy fixing?
A: Logical bugs that require deep thinking.

225. Q: What kind of bugs do you hate?
A: Bugs caused by unclear requirements.

226. Q: How do you keep code readable?
A: By writing simple logic and meaningful names.

227. Q: How do you handle large codebases?
A: By understanding structure first, then making small changes.

228. Q: How do you onboard yourself to new projects?
A: By running the project locally and reading the code flow.

229. Q: How do you document your work?
A: Through README files and clear comments where needed.

230. Q: How do you ensure consistency across projects?
A: By following similar structure and naming conventions.

231. Q: What do you do when something breaks in production?
A: I identify the root cause and fix it calmly.

232. Q: How do you prioritize bugs vs features?
A: Critical bugs always come first.

233. Q: How do you test without a test framework?
A: By manually testing edge cases and flows.

234. Q: What is your attitude toward testing?
A: Testing is important, but clean code prevents most bugs.

235. Q: How do you handle performance bottlenecks?
A: By profiling and optimizing the slowest parts.

236. Q: How do you debug database issues?
A: By checking queries, indexes, and data consistency.

237. Q: How do you optimize database queries?
A: By indexing and reducing unnecessary queries.

238. Q: How do you handle large datasets?
A: By pagination and efficient querying.

239. Q: How do you handle API failures?
A: With retries and graceful error handling.

240. Q: How do you ensure reliability?
A: By designing simple, predictable systems.

241. Q: What do you expect from your first job?
A: Learning, mentorship, and real ownership.

242. Q: What do you expect from your teammates?
A: Honesty, collaboration, and accountability.

243. Q: What do you expect from your manager?
A: Clear direction and constructive feedback.

244. Q: What kind of culture do you prefer?
A: Learning-focused and transparent.

245. Q: What values do you bring to a company?
A: Ownership, consistency, and problem-solving.

246. Q: How do you handle feedback from seniors?
A: I listen carefully and apply improvements.

247. Q: How do you handle mistakes at work?
A: I own them and fix them quickly.

248. Q: How do you help your teammates?
A: By sharing knowledge and helping debug issues.

249. Q: What makes you a good teammate?
A: Reliability and clear communication.

250. Q: What kind of leader do you want to be?
A: A calm, supportive, and technically strong leader.

251. Q: How do you stay motivated long term?
A: By focusing on growth, not just results.

252. Q: What do you do when motivation is low?
A: I rely on discipline instead of motivation.

253. Q: What habits improved you the most?
A: Daily coding and consistent learning.

254. Q: What habit are you currently building?
A: Writing cleaner and more maintainable code.

255. Q: What habit did you remove?
A: Overthinking before starting.

256. Q: What is your biggest personal win?
A: Staying consistent over years.

257. Q: What are you most proud of today?
A: My progress and projects.

258. Q: What keeps you going?
A: The joy of building things.

259. Q: What is your definition of success?
A: Continuous growth and impact.

260. Q: What would you tell your future self?
A: Keep building and stay humble.

261. Q: What would you do if you had no fear?
A: Build bigger systems and take more risks.

262. Q: What would you build if money didn't matter?
A: Tools that help students learn better.

263. Q: What kind of projects do you want to build next?
A: Scalable backend systems and AI-powered apps.

264. Q: What kind of problems do you want to solve in life?
A: Problems that improve people's daily lives.

265. Q: What impact do you want to leave?
A: Useful, reliable software.

266. Q: What do you want to be known for?
A: Building clean and scalable systems.

267. Q: What do you want recruiters to remember about you?
A: That I can build, not just talk.

268. Q: What is your personal mission?
A: To build software that matters.

269. Q: Why should someone trust you?
A: Because I take responsibility seriously.

270. Q: What legacy do you want to leave?
A: Code that helps people.

PROMPT :-

You are a professional assistant that answers user questions about Arunabha Banerjee.

- Always expand short answers into detailed, well-structured responses.

- Use fluent grammar, professional wording, and engaging style.

- If the answer is factual (from vector DB), enhance it with smooth wording.

- Be precise, confident, and professional.