**Introduction:**  
My name is omkar patil basically I am from Kolhapur but currently I am living in pune. I have completed my graduation in Bachelors of Science in computer science from shivaji university, Kolhapur and after that I have completed my post-graduation in master in computer application from Pune University, pune. I am a Java Full Stack Developer with over **3 years of experience**, specializing in **backend development using Java, Spring Boot, and Hibernate**. During my tenure at **[AMEDEA InfoTech]**, I worked on multiple projects, with a key focus on the **DSA (Direct Selling Agent) Management Module**.

**Project Overview – DSA Management Module:**  
The **DSA Management Module** is a system designed to efficiently manage **Direct Selling Agents (DSAs)**—individuals or entities that act as intermediaries between borrowers and financial institutions like **banks and NBFCs**. The primary goal of this project was to **streamline agent operations**, improve borrower-agent-bank coordination, and enhance overall loan application processing.

**2. Problem Statement (Why This Project Matters)**  
"In the lending ecosystem, banks and NBFCs rely heavily on DSAs to generate leads and process loan applications, especially in rural and urban areas. However, managing these agents manually can be chaotic—tracking their profiles, verifying their documents, assigning territories, and calculating their incentives often leads to delays, errors, and inefficiencies. Our project aimed to solve this by building an automated, scalable, and user-friendly platform to manage DSAs effectively."

**3. Key Features (What the Project Does)**  
"The DSA Management Module is divided into three main parts: Operations on DSAs and Operations on Agent Incentives and customer Let me briefly walk you through the key features:

* **Agent Management**:
  + Register new DSAs by collecting their details and performing KYC (Know Your Customer) verification, including document uploads and validation.
  + Update or deactivate agent profiles and reactivate them when needed.
  + Assign territories or regions to agents for better market coverage.
  + Retrieve agent information like performance metrics or sales history for analysis.
* **Incentive Management**:
  + Define incentive structures based on performance goals, such as commissions per loan sanctioned.
  + Track agent progress in real-time and calculate their earnings.
  + Disburse incentives seamlessly through the system.  
    This system not only simplifies agent management but also ensures transparency and efficiency for banks, NBFCs, and the agents themselves."

**4. Technical Implementation (How You Built It)**  
"Now, let me explain how I built this system using various technologies:

* **Spring Boot**: I used Spring Boot as the backend framework because it simplifies development with its microservices architecture and built-in tools. It helped me create RESTful APIs to handle requests like agent registration, profile updates, and incentive calculations.
* **Hibernate**: For database operations, I integrated Hibernate as the ORM (Object-Relational Mapping) tool. It allowed me to map Java objects to MySQL tables effortlessly and perform CRUD operations—like adding a new DSA or retrieving agent details—without writing complex SQL queries.
* **Core Java**: I leveraged Core Java for the business logic, such as validating KYC documents, calculating incentives based on loan amounts, and handling exceptions to ensure the system runs smoothly.
* **MySQL**: I designed a relational database in MySQL to store agent profiles, KYC documents, territory assignments, and incentive records. I optimized the schema with proper indexing to ensure fast retrieval of data.  
  The application follows a layered architecture—controller, service, and repository layers—which makes it modular and easy to maintain."

**5. My Contributions (What You Did)**  
"In this project, I was involved end-to-end—from designing the database schema to implementing the backend logic and testing the APIs. I created secure endpoints for agent registration and KYC verification, ensuring data integrity and user authentication. I also implemented a feature to generate reports on agent performance, which was a value-add for decision-making. Additionally, I wrote unit tests using JUnit to ensure the reliability of the system."

**6. Challenges and Solutions (Show Problem-Solving Skills)**  
"One challenge I faced was ensuring the KYC document verification process was both secure and efficient. Initially, handling large file uploads slowed down the system. To solve this, I implemented asynchronous file processing using Spring Boot’s @Async annotation and stored the files in a separate directory while linking their metadata in MySQL. This improved performance significantly. Another challenge was calculating incentives dynamically for different loan types—I tackled this by designing a flexible formula in Core Java that adapts to various commission percentages."

-----------------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------------

I have 3 years of professional experience working as a full stack developer, primarily using Java for backend development, Angular for frontend, and SQL Server for database management. Throughout my projects, I've been involved in designing, developing, and maintaining both the client-side and server-side of applications. So I’d say I have:

3 years of experience with Java (Spring Boot and REST APIs),

3 years with Angular (including components, services, routing, etc.),

and 3 years of experience with SQL Server — writing queries, stored procedures, and optimizing performance.

I’ve worked across the stack and am confident in handling both frontend and backend responsibilities as part of a full stack role."

### Tips:

If some technologies are more recent, you can slightly adjust:

"While I started with Java and SQL Server, I’ve been using Angular actively for over 2.5 years, so all three stack layers are part of my daily development work."

If you had more backend-focused roles, add:

"Even when my role leaned more towards backend, I regularly collaborated on frontend tasks, and I’m fully comfortable working across the full stack."