

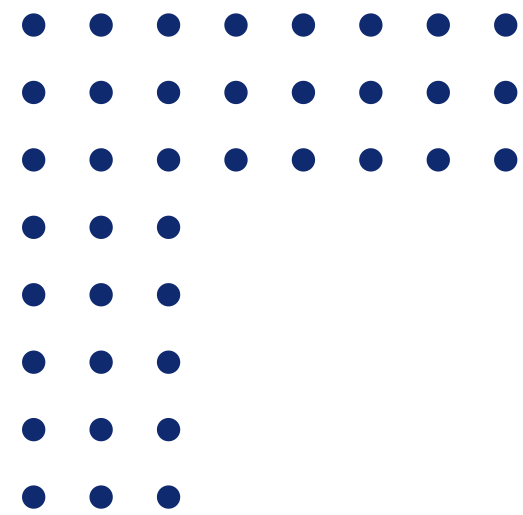


**FLIPKART**

**JEDI BOOTCAMP PROJECT**

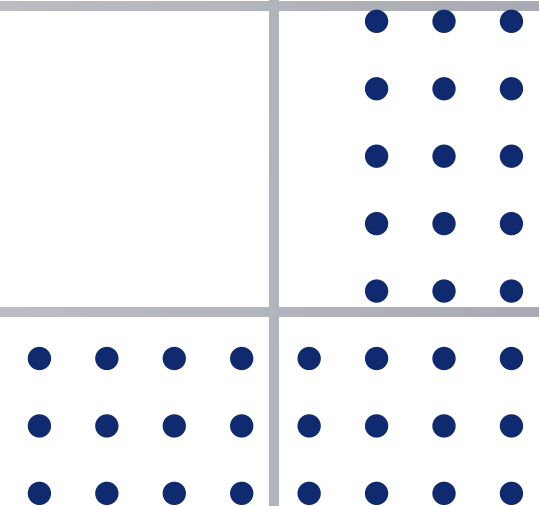
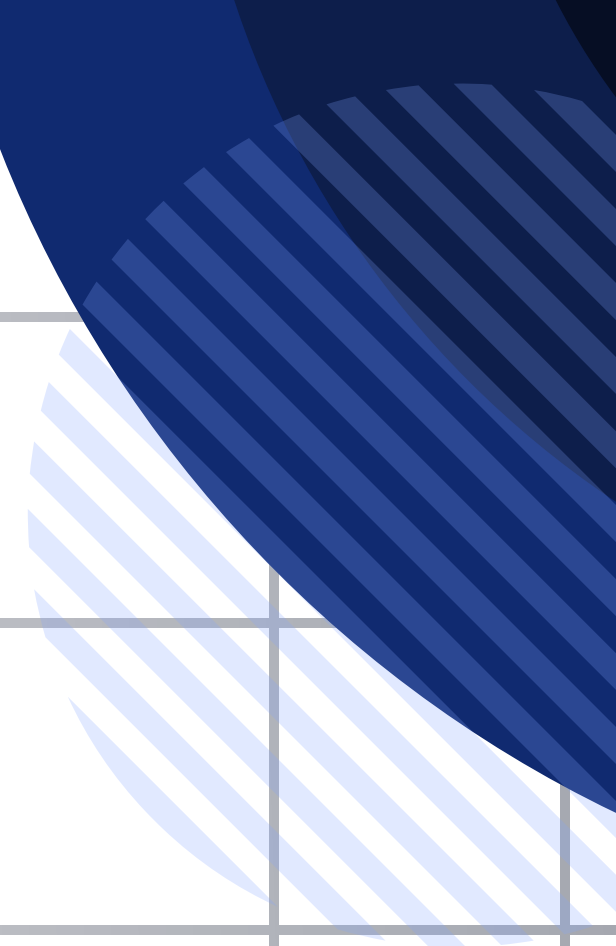
**TEAM IRIS**





# AGENDA

1. Project Goals
2. Our Vision
3. Our Journey
4. Our Team
5. Engineering Practices
6. Tech Stack
7. Development Lifecycle
8. Challenges & Learnings
9. Demo
10. Questions






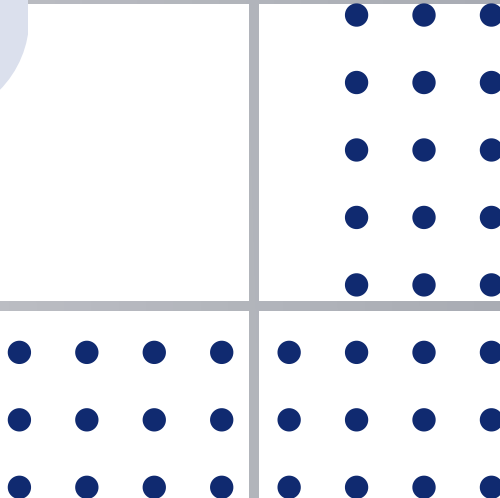
# THE ASK



## **Problem Statement:**

Design a system for "FlipFit," an enterprise fitness application for Flipkart.


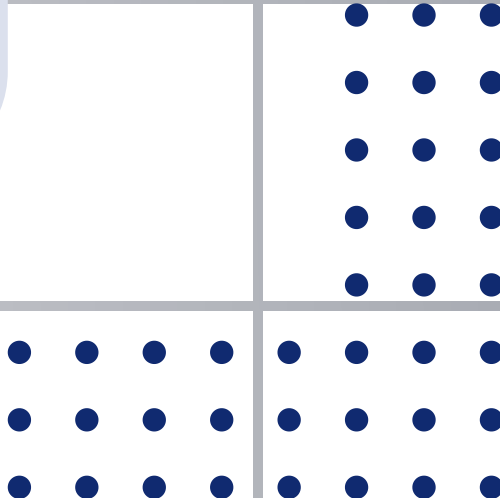
## **Key Requirements:**

- Partnering with gyms across Bangalore.
  - Manage multiple centers with fixed hourly slots (n slots/day).
  - Fixed seat capacity per slot.
  - Operations: User registration, view availability, book workouts.
- 
- 



# OUR VISION



- **Quality:** Scalable microservices architecture.
  - **Security:** Robust user authentication and role-based access.
  - **Speed:** High-performance backend using Java 17 and MySQL.
  - **Interactivity:** Intuitive UI for seamless gym discovery and booking.
- 
- 



# OUR TEAM



- Sakshi Hingane (Group Lead)
  - Nandini Gupta (Sub Group Leader)
  - Aditya Hansraj
  - Akshat Kumar Nayak
  - Biraj Sanghai
  - Deepkumar Patel
  - Diya Kailash
  - Ishan Datta
  - Mohim Mahajan
  - Pranav Bhutada
- 
- 



# OUR JOURNEY



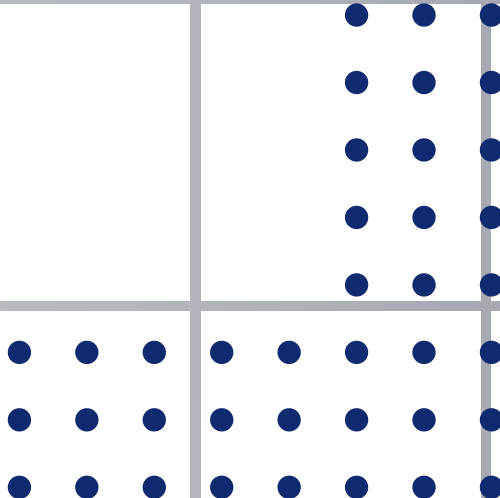
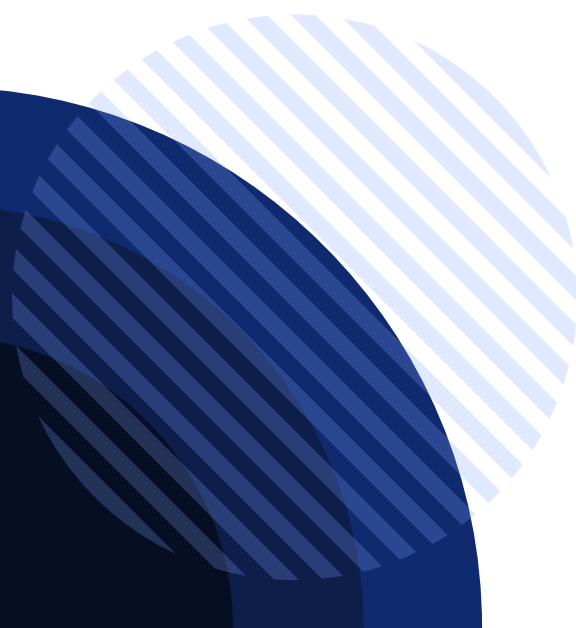
Phase 1: Lab Setup & Git Lifecycle  
Configuration of Java 17, MySQL, STS/IntelliJ, and Maven.

Phase 2: Design Thinking  
Creating UML Artifacts (Use Case, Activity, and Class Diagrams).

Phase 3: Java Development (POS Phase)  
Implementing core logic with Collections (List/Map/Set) for  
hardcoded data.

Phase 4: Persistence & Database  
MySQL Schema design and JDBC (DAO) implementation with  
Prepared Statements.


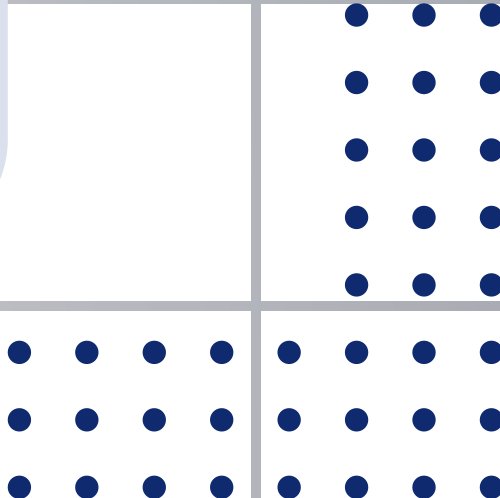
Phase 5: Advanced Features & Web API  
Exception handling, Java 8+ features (Streams/Lambda), and  
Dropwizard integration.





# ENGINEERING PRACTICES


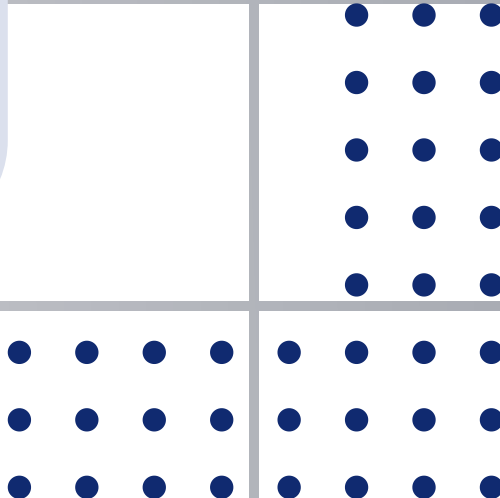


- **Agile Methodology:** Sprint planning, grooming, and implementation.
  - **TDD (Test Driven Development):** Ensuring code reliability.
  - **SDLC Excellence:** Git commits with PR reviews and Confluence documentation.
  - **Static Analysis:** Sonar analysis for code quality.
  - **Observability:** Metrics and logs emitted from the code.
- 
- 



# TECH STACK



- **Backend:** Java 17, Dropwizard.
  - **Database:** MySQL (JDBC for connectivity).
  - **Tools:** Maven, Git (SCM), JIRA (Story Management).
  - **Frontend/UI:** Figma for design, full-stack integration on FK platform.
  - **Infrastructure:** Microservices architecture.
- 
- 






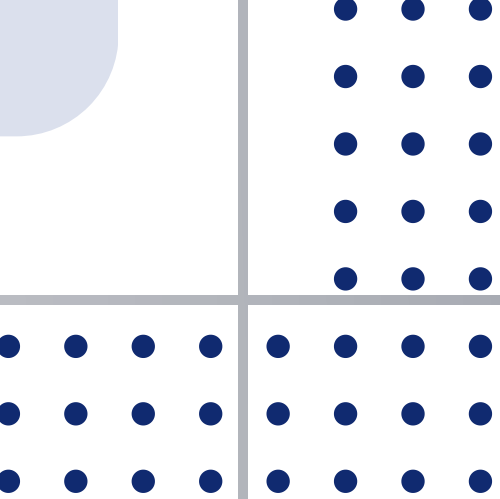
# DEVELOPMENT & DESIGN



## UML Artifacts:

- **Use Case:** Identify Actors (Admin, Customer, Gym Owner).
  - **Activity:** Flow of booking and slot management.
  - **Class Diagram:** Relationship between User, Role, Center, Slot, and Booking.
- 


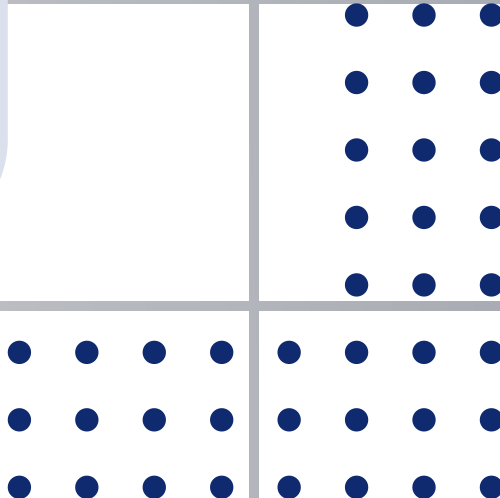
## Key Modules:

- `com.flipfit.dao`: Data Access Layer.
  - `com.flipfit.service`: Business Logic with Stream API.
  - `com.flipfit.exception`: Custom error handling (e.g. `UserNotFoundException`)
- 



# CHALLENGES & LEARNINGS




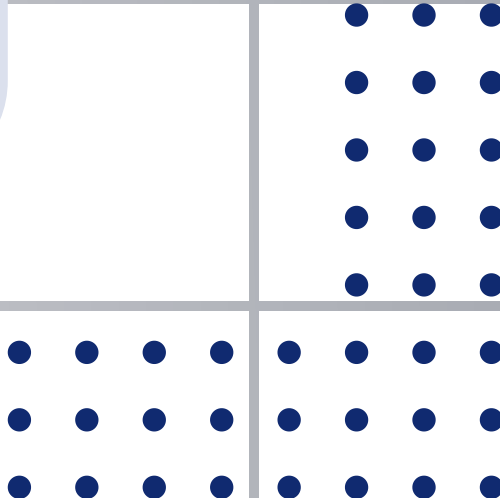
- **Concurrency:** Handling real-time slot availability to prevent overbooking.
  - **Transitioning:** Moving from Collections-based storage to persistent MySQL storage.
  - **New Tech:** Implementing Dropwizard and JAX-RS annotations for RESTful services.
  - **Collaboration:** Working as a 10-member team on a shared Git repository (JEDI-IRIS-DEVELOPMENT-FLIPKART).
- 
- 



# DEMO



## Live Walkthrough:

- User Login and Session Time logging.
  - Filtering Gym Owners (Approved vs. Not Approved) using Stream API.
  - The booking flow and "Bonus Stories" (Waitlist promotion/Cancellation).
- 
- 



ANY  
QUESTIONS?

THANK YOU