These projects not only utilize Spring Boot and a database but also incorporate various data structures and system design principles, making them ideal for showcasing your skills in a FAANG interview.

**1. Personal Finance Manager**

* **Description**: Develop an application to help users manage their personal finances by tracking expenses, setting budgets, and generating financial reports.
* **Key Features**:
  + **Data Structures**: Use hash maps for categorizing expenses, linked lists for transaction history, and graphs for visualizing financial trends.
  + **System Design**: Implement services for user authentication, expense tracking, budget management, and reporting.
  + **Technologies**: Spring Boot, Spring Security, JPA/Hibernate, PostgreSQL or MySQL.

**2. Event Management System**

* **Description**: Create a platform for managing events, including event creation, attendee registration, and schedule management.
* **Key Features**:
  + **Data Structures**: Use hash maps for event details and registration lists, linked lists for managing event schedules, and trees for categorizing events.
  + **System Design**: Develop services for user authentication, event creation, registration, and notification.
  + **Technologies**: Spring Boot, Spring Security, JPA/Hibernate, MongoDB or MySQL.

**3. Online Voting System**

* **Description**: Build a secure online voting platform for conducting elections and surveys.
* **Key Features**:
  + **Data Structures**: Use hash maps for storing voter information and vote counts, and trees for organizing candidates and election data.
  + **System Design**: Implement services for voter authentication, vote casting, and result tallying.
  + **Technologies**: Spring Boot, Spring Security, JPA/Hibernate, PostgreSQL or MySQL.

**4. Recipe Sharing Platform**

* **Description**: Create a community-driven platform for sharing and discovering recipes, with user ratings and comments.
* **Key Features**:
  + **Data Structures**: Use hash maps for recipe ingredients, linked lists for user comments and ratings, and trees for recipe categorization.
  + **System Design**: Develop services for user authentication, recipe submission, rating, and commenting.
  + **Technologies**: Spring Boot, Spring Security, JPA/Hibernate, MongoDB or MySQL.

**5. Workout and Fitness Tracker**

* **Description**: Develop an application for tracking workouts, fitness goals, and progress over time.
* **Key Features**:
  + **Data Structures**: Use hash maps for workout plans and progress tracking, linked lists for daily logs, and graphs for visualizing fitness trends.
  + **System Design**: Implement services for user authentication, workout logging, goal setting, and progress reporting.
  + **Technologies**: Spring Boot, Spring Security, JPA/Hibernate, PostgreSQL or MySQL.

**6. Inventory Management System for Small Businesses**

* **Description**: Build a system for small businesses to manage their inventory, track stock levels, and generate reports.
* **Key Features**:
  + **Data Structures**: Use hash maps for inventory items, linked lists for transaction history, and graphs for visualizing stock trends.
  + **System Design**: Develop services for user authentication, inventory tracking, and reporting.
  + **Technologies**: Spring Boot, Spring Security, JPA/Hibernate, MySQL.

**Personal Finance Manager**

While this project demonstrates a solid understanding of Spring Boot and database integration, showcasing it in a FAANG interview would be more impactful if it also included some advanced features and considerations. Here are some enhancements to make it more impressive:

1. **Advanced Features**:
   * **Analytics Dashboard**: Create a dashboard to show financial analytics using charts and graphs.
   * **Transaction Categorization**: Implement automatic categorization of expenses.
   * **Notifications**: Add email or SMS notifications for budget limits or unusual spending.
2. **System Design**:
   * **Scalability**: Discuss how you would scale the application to handle more users.
   * **Security**: Implement and highlight secure coding practices, data encryption, and OAuth2 for authentication.
   * **Microservices**: Consider breaking the application into microservices if time allows.
3. **Code Quality and Best Practices**:
   * **Unit Testing**: Ensure you have comprehensive unit tests and possibly integration tests.
   * **Documentation**: Provide clear documentation and API specifications.
4. **Deployment**:
   * Deploy the application to a cloud platform (e.g., AWS, GCP) and highlight this in your project.

**Conclusion**

With the above enhancements, this project can indeed be suitable for showcasing in a FAANG interview. It demonstrates your ability to build a full-stack application using Spring Boot, your understanding of data structures, and your knowledge of system design principles. The key is to clearly articulate your design decisions, the challenges you faced, and how you overcame them during the interview.