### 1. **项目和团队简介：**

* **团队分工：**
  + 说明每个团队成员的角色和责任，如何协作完成不同任务。
  + 可使用 **团队组织结构图** 来展示分工。
* **工具使用：**
  + 列出项目中使用的开发工具、框架和技术栈（例如，Spring Boot, Vue.js, MySQL 等）。
  + 可使用 **技术栈图** 或 **工具使用的流程图** 来清晰展示工具的选择和使用流程。
* **项目背景：需求出发：**
  + 说明项目的背景、目标以及需求分析。
  + 可以使用 **需求分析图**，例如用户故事图或需求列表。
* **项目管理机制：**
  + 说明项目的管理方法（如敏捷开发），及如何进行迭代开发、任务分配。
  + 可以使用 **甘特图** 或 **任务分配图** 来展示进度和管理机制。
* **编码规范：**
  + 提供编码规范的主要原则，如命名规范、代码格式等。
  + 可用 **编码规范文档摘要** 或 **代码片段示例** 来展示。

### 2. **功能建模：**

* **用例图：**
  + 展示系统的功能和与外部实体（用户、其他系统等）的交互。
  + **用例图**：通过用例和参与者展示系统功能，描述核心操作和目标。
* **抽象：**
  + 对功能进行抽象，提炼出关键功能模块。
  + **抽象模型图**，如模块划分图，帮助阐述功能的层次结构。
* **类图：**
  + 通过类图展示系统的类结构及它们之间的关系。
  + **初步类图**，展示主要的类和它们的属性、方法及类之间的关联（继承、依赖等）。
* **动态建模：**
  + 描述系统中的动态行为，如对象如何协作、状态如何变化。
  + **状态图/活动图**，展示对象生命周期中的不同状态和状态之间的转换。
* **时序图：**
  + 展示不同对象之间的消息交换过程，突出时序关系。
  + **时序图**，描述系统中各个对象的互动顺序。
* **类图类精化：**
  + 在原有类图的基础上，进一步精化类结构，如添加细节、调整方法等。
  + **精化后的类图**，说明如何通过设计优化类结构，确保高内聚低耦合。
* **对象设计：**
  + 基于分析阶段的结果，完成类中属性的数据结构定义和操作设计。
  + **类设计表格或图**，展示每个类的详细设计，如属性类型、格式和方法的实现。
* **类间消息传递：**
  + 描述类之间如何协作以及消息的传递过程。
  + **消息传递图**，用图形方式描述类间的消息调用和顺序。
* **数据流图：**
  + 描述系统中数据流动的过程。
  + **数据流图**，展示数据输入、输出和转换过程。
* **流程图：**
  + 展示系统操作过程中的各个步骤，明确输入输出。
  + **流程图**，具体到业务流程、操作流程或控制流程。

### 3. **设计：**

* **设计模式：**
  + 介绍采用的设计模式，如工厂方法模式、单例模式、策略模式。
  + 可使用 **设计模式图解**（类图）来描述每个设计模式的实现。
  + 详细说明每种设计模式的适用场景、问题、解决方案和效果。
* **总体架构图：**
  + 展示整个系统的架构，如何划分为不同的模块，模块之间的交互。
  + **系统架构图**，可以使用组件图或部署图展示系统的结构。
* **数据库设计：**
  + 展示系统的数据库结构，包括表的设计、字段和关系。
  + **ER图（实体关系图）**，展示数据库中的实体、属性及实体间的关系。
* **UI设计：**
  + 展示系统的用户界面设计，展示各个界面的布局和交互。
  + 使用 **UI原型图** 或 **界面设计图**，可通过工具（如Figma、Sketch等）生成。

### 4. **测试：**

* **单元测试、集成测试、确认测试、系统测试：**
  + 说明如何进行不同阶段的测试，确保系统功能的正确性。
  + 展示 **测试流程图** 或 **测试用例表**，包含每个测试阶段的测试内容、目的和结果。
* **白盒测试：**
  + 关注代码内部逻辑的正确性，通常通过单元测试实现。
  + 使用 **单元测试代码片段** 或 **测试报告** 来展示。
* **黑盒测试：**
  + 关注系统外部接口和功能的正确性，通常通过集成测试实现。
  + 展示 **集成测试用例**，测试系统输入和输出的正确性。
* **回归测试：**
  + 确保新功能的添加不会影响系统的其他功能。
  + 展示 **回归测试报告**，描述回归测试的覆盖范围和结果。

### 5. **演示：**

* **现场演示，展示界面和功能：**
  + 实时展示系统的主要功能，突出展示用户交互部分和后台服务的操作流程。
  + 可以准备 **现场演示视频**，提前录制好项目功能展示。

### 6. **常见提问：**

* **数据安全：**
  + 说明项目如何保证数据的安全性（如加密、权限控制等）。
  + 可准备一个 **数据安全设计图**，展示如何处理用户数据和敏感信息。
* **功能完善：**
  + 说明系统是否已经满足所有功能需求，是否有任何改进空间。
  + 可使用 **需求满足度表**，展示已实现功能与需求的对比。
* **如何改进：**
  + 说明未来版本可能的改进方向，功能扩展或优化。
  + 可准备 **改进路线图**，展示未来的版本计划。

### 7. **老师总结不足：**

* **提前部署准备：**
  + 确保现场演示前完成部署，避免技术故障。
  + 记录部署过程及 **部署架构图**，说明部署过程和环境配置。
* **避免虚话：**
  + 具体说明技术选择的原因，可以准备 **技术选择对比表**，对比不同技术的优缺点。
* **非功能需求量化：**
  + 将非功能需求（如性能、安全、可扩展性等）具体化，提供量化指标。
  + 使用 **非功能需求指标图**，如响应时间、系统吞吐量等。
* **完整的测试：**
  + 确保所有测试用例都执行并记录，展示完整的 **测试覆盖率图** 或 **测试报告**。

这样的展示结构不仅能够全面覆盖项目的所有重要方面，还能清晰地传达每个环节的设计、实现和测试成果。每个部分的图表和内容将帮助评审理解项目的深度与广度，也有助于展示你对设计模式和系统架构的深入理解。