

中期汇报演讲稿（含翻页提示）

第1页 封面

- 各位老师同学好，我们团队今天汇报铁路12306系统的中期进展，围绕完成度、成功/失败案例与协作展望展开，时长7分钟，最后3分钟答疑。
- 我会在每页末加入翻页提示，便于把控节奏。
- 翻页提示：请切到第2页

第2页 议程

- 完成情况与需求树：当前实现与端到端完成度
- 成功案例：模型+测试驱动如何产出可执行代码
- 失败案例：生成失败的工程问题与手动修复
- 团队协作与未来展望
- 演示与Q&A
- 翻页提示：请切到第3页

第3页 完成情况总览（需求树+计数）

- 技术需求总览（依据 /中期汇报/需求树.md）：
 - 技术需求总数：34项（功能性15、非功能性13、数据要求6）
 - 严格按PRD规格完全实现：2项
 - 部分实现：19项
 - 未实现/未验证：13项
- 场景与需求树（完成度标注）：
 - 用户认证模块
 - 支持手机号/邮箱注册：已实现（backend/src/controllers/registerController.js；backend/src/services/registrationDbService.js）
 - 密码强度验证（≥8位含字母+数字）：部分实现（当前规则≥6且字符集组合，backend/src/controllers/registerController.js）
 - 图形验证码与短信验证码：部分实现（短信验证码已实现；图形验证码未实现，backend/src/controllers/authController.js）
 - 登录失败次数限制与账户锁定：未实现（backend/src/services/authService.js）
 - 会话管理与自动登录：部分实现（会话基础有；自动登录未实现，backend/src/services/sessionService.js）
 - 车票查询模块
 - 城市/车站模糊搜索：已实现（frontend/src/components/StationPicker.tsx）
 - 日期选择限制（当前起30天内）：部分实现（当前14天限制，frontend/src/components/CalendarPopover.tsx:41-60,106-120）
 - 多条件筛选（车次类型、时间段、席别等）：部分实现（类型/席别可筛；时间段未见，frontend/src/pages/TrainsPageContainer.tsx）
 - 车次信息实时展示与排序：部分实现（展示有；排序未见，frontend/src/components/TrainListTable.tsx）
 - 余票实时查询与更新：部分实现（静态/推断，backend/src/controllers/trainController.js）

- 订单管理模块
 - 常用乘车人信息管理（最多15人）：未实现（仅订单内乘客记录，`backend/src/controllers/orderController.js`）
 - 订单信息验证与重复提交防护：部分实现（唯一订单号但无幂等键，`backend/src/services/dbService.js` 建表）
 - 订单状态跟踪（待支付/已支付/已取消/已完成）：未实现（仅 pending，`backend/src/services/dbService.js`）
 - 支付超时自动取消：未实现
 - 历史订单查询与分页展示：未实现（列表无分页，`backend/src/controllers/orderController.js`）
- 非功能性需求
 - 性能（首屏<3s、查询<2s、并发≥100、索引优化）：未验证/未实现（SQLite未见索引）
 - 安全（密码哈希、HTTPS、SQL注入/XSS、输入验证、失败次数/IP限制）：
 - 密码哈希：部分实现（采用 bcrypt，强于MD5/SHA256但与规范不一致，`backend/src/services/registrationDbService.js`）
 - HTTPS：未实现（`backend/src/app.js` 未启用）
 - SQL注入防护：已实现（参数化查询，`backend/src/services/dbService.js`）
 - XSS防护：部分实现（helmet 安全头、React默认转义；未见输入清洗，`backend/src/app.js`）
 - 输入验证与过滤：部分实现（手写校验为主，`backend/src/controllers/registerController.js`）
 - 登录失败/IP限制：未实现
 - 可用性（界面清晰、错误提示、浏览器支持、确认机制）：部分实现（UI与提示存在；跨浏览器与确认机制未验证）

- 数据要求
 - 数据完整性（用户/乘客、车次/余票、订单/状态一致性）：部分实现（验证与结构存在但缺状态机与实时余票）
 - 数据安全性（敏感信息加密、备份恢复、操作日志与审计）：
 - 加密存储：部分实现（bcrypt）
 - 备份/恢复：未实现
 - 日志/审计：部分实现（console 为主，无结构化/审计链）
- 依据材料：`/e:/LQiu/cs3604/12306_Follow/中期汇报/需求树.md` 与现有代码映射
- 翻页提示：请切到第4页

第4页 首页完成情况

- 简单说，用户一进来就能用：上面有导航，中间是轮播，右侧一张“查询卡片”。
- 地址就是网站根路径 `/`，路由已经指到 `HomePage` (`frontend/src/App.tsx:12-15`)。
- 查询卡片该有的都有：出发地、到达地、日期；支持单程、往返、中转换乘、退改签；站点有选择器，日期是双月日历，还能勾“学生票”“高铁/动车”。
- 点一下“查询”，我们带着参数直接跳到 `/trains` (`frontend/src/pages/HomePage.tsx:80-109`)，不让用户迷路。
- 现在还没接“首页内容接口”（公告、功能卡片），后端已准备好 (`backend/src/routes/auth.js:15-16`;
`backend/src/controllers/authController.js:137-160`)，不影响演示，收尾阶段再补。

- 翻页提示：请切到第5页

第5页 登录页完成情况

- 登录我们做成“两步走”：第一步账号+密码，第二步短信验证码。
- 用户提交表单，我们先创建会话（`POST /api/auth/login`, `frontend/src/pages/LoginPage.tsx:20-37`），随后弹出短信验证码弹窗；6位数字输入对了就算登录成功（`POST /api/auth/verify-login`, `frontend/src/pages/LoginPage.tsx:60-88`）。
- 两件事还在收尾：登录成功后的自动跳转留了个 TODO（`frontend/src/pages/LoginPage.tsx:73-77`）；“忘记密码”按钮有了，页面还没接，但后端接口已准备好（`backend/src/routes/auth.js:18-19`, `backend/src/controllers/authController.js:162-187`）。
- 总体来说，登录链路能跑通、错误提示也完整，体验上就差最后一步导航。
- 翻页提示：请切到第6页

第6页 注册页完成情况

- 注册我们按“标准三步”来：先提交资料拿到 `sessionId`（`POST /api/register`, `frontend/src/pages/RegisterPage.tsx:32-77`），再发验证码并弹出验证框（`POST /api/register/send-verification-code`, `frontend/src/pages/RegisterPage.tsx:48-69`），最后输入验证码完成注册（`POST /api/register/complete`, `frontend/src/pages/RegisterPage.tsx:91-105`）。
- 成功后会有提示，2秒自动带你回登录页继续流程；字段校验和错误提示都在表单里，体验不迷路。
- 后端这条线的路由和校验都齐了（`backend/src/routes/register.js:11-23`），所以这是目前唯一一个“端到端完成”的场景。
- 翻页提示：请切到第7页

第7页 成功案例：模型+测试驱动（CalendarPopover）

- 我们挑了一个最能体现“测试驱动+模型”的小而美案例：首页的日期选择器。
- 先把要求说人话：只能选“今天起14天内”；两个月并排展示不能挤成一团；数字要等宽，“今天”要醒目；不管点输入框还是右边的小日历，都能打开同一个面板。
- 做法很直接：我们先列验收清单当“测试”，让模型根据这些边界生成最小代码；然后按清单一条条核对，把红灯一个个变成绿灯。
- 对应位置也好找：
 - 范围与“今天”：`frontend/src/components/CalendarPopover.tsx:41-61`
 - 双月并排与面板结构：`frontend/src/components/CalendarPopover.tsx:62-116`
 - 输入框/图标双入口触发：`frontend/src/pages/HomePage.tsx:185-199`、`190-198`
 - 样式标识（等宽数字、“今天”标签）：`frontend/src/components/CalendarPopover.tsx:91-96`
- 为什么能让模型“写得准”？因为边界清楚、组件粒度小、反馈快——它只需要把“可通过的最小实现”拼出来，我们再抛光细节。
- 翻页提示：请切到第8页

第8页 失败案例：生成失败与手动修复

- 我们一开始想“偷懒”：直接用一整张首页的截图，让模型一次性把页面生出来。结果很不理想——布局错位、间距不一致、字体粗细和颜色都跑偏，细节完全不达标。
- 后来换了思路：把首页拆成模块，一块一块来复刻。
 - 顶部导航 `TopNavigation`、主导航 `MainNavigation`、轮播区域、右侧查询卡片分别提供单独截图和验收点（间距、字号、颜色、hover、对齐）。
 - 每个模块给模型明确提示和约束，让它只改这一块；我们叠加对比图逐项核对，发现问题就继续小步修

正。

- 这个“模块截图 → 验收要点 → 迭代修改”的节奏更适合：
 - 约束更清晰，模型不容易“跑题”；
 - 影响范围可控，样式不互相干扰；
 - 复验更高效，逐块达到 1:1 复刻。
- 最终效果：主页达到像素级一致，样式也顺手整理出统一的设计变量（颜色、字号、间距），后续维护更轻松。
- 经验教训：整页大图“一把梭”的成功率很低；改用模块化复刻+明确验收标准，能稳定收敛到目标。
- 翻页提示：请切到第9页

第9页 团队协作与展望

- 现状：需求→接口契约→后端→前端→测试的协作流程基本成形；后端覆盖查询/详情/订单/鉴权，前端首页/登录/注册完成度较好
- 待改进：
 - 登录成功跳转与忘记密码页面落地
 - 首页接入 `GET /api/auth/homepage`
 - 个人中心API与页面实现
- 近期计划（至11/17）：
 - 完成登录成功跳转（`frontend/src/pages/LoginPage.tsx:73-77`）
 - 首页接入后端首页内容接口
 - 落地“忘记密码”页面并消费接口
 - 规划个人中心API（资料、订单汇总）并实现
- 翻页提示：请切到第10页

第11页 Q&A 引导

- 预设问答：
 - 为什么首页未接入后端首页内容接口？答：不影响主流程，属增强项，已排期补齐
 - 登录后未自动跳转的影响？答：流程已通，只需在既定位置补导航逻辑
 - 实时余票与排序如何保障？答：当前为示例数据，后续通过数据源与索引优化实现
 - 个人中心为何缺失？答：契约先行，分阶段补齐资料与订单汇总
- 翻页提示：可返回第3页或第12页备份视图

第12页 备份页（索引与素材）

- 路由入口：`frontend/src/App.tsx:12-16`
- 首页：`frontend/src/pages/HomePage.tsx:80-109`、`115-131`、`150-228`
- 登录：`frontend/src/pages/LoginPage.tsx:20-37`、`60-88`、`73-77`、`43-46`
- 注册：`frontend/src/pages/RegisterPage.tsx:32-77`、`48-69`、`91-105`、`170-181`
- 导航修复：`frontend/src/components/MainNavigation.tsx:9`
- 日期组件：`frontend/src/components/CalendarPopover.tsx:41-61`、`62-116`、`91-96`
- 后端接口：`backend/src/routes/auth.js:6-19`，`backend/src/controllers/authController.js:137-160`、`162-187`；`backend/src/routes/register.js:11-23`
- 翻页提示：演讲结束，感谢聆听