### **课程报告：个人数字助理系统开发**

#### **课程目标**

本课程的目标是掌握分布式web应用开发技术基础，了解web前端和分布式后端的运行环境，掌握ECMAScript/JavaScript核心语法，使用Node.js和Vue进行应用开发。

#### **项目概述**

项目名称：个人数字助理系统

该系统旨在帮助用户管理个人信息、笔记和待办事项，提高日常生活和工作的效率。项目采用前后端分离的架构，前端使用Vue3进行开发，后端使用Node.js进行开发，数据存储采用SQLite。系统包含用户登录/注册、用户信息管理、笔记管理和待办事项管理四个主要模块。

#### **项目开发思路与步骤**

开发此系统的思路主要包括三个阶段：需求分析、系统设计和系统实现。

在需求分析阶段，通过调查了解用户对个人数字助理系统的需求，确定系统需要实现的功能。功能需求包括用户注册和登录、用户信息管理、笔记管理和待办事项管理。

在系统设计阶段，设计了系统的整体架构和模块划分。前端部分使用Vue3框架，通过Vue Router实现页面跳转，使用Pinia进行状态管理，利用TailwindCSS进行界面风格渲染，确保系统具备良好的用户体验和响应式设计。后端部分采用Node.js框架，基于MVC架构设计，包含User、Note和Todo三个模型及对应的路由模块，使用SQLite进行数据存储。

在系统实现阶段，分步骤开发各个功能模块，并进行集成和测试。具体步骤如下：

1. **前端实现**：
   * 使用Vue CLI创建Vue3项目，配置Vue Router和Pinia。
   * 开发用户登录/注册模块，包括登录和注册表单，表单验证和API请求处理。
   * 开发用户信息模块，展示用户的基本信息，并实现信息编辑和保存功能。
   * 开发笔记管理模块，允许用户创建、查看、编辑和删除笔记。利用Pinia进行笔记状态管理，确保笔记的创建、更新和删除操作实时更新。
   * 开发待办事项管理模块，允许用户添加、查看、标记完成和删除待办事项。使用Pinia管理待办事项的状态。
   * 使用TailwindCSS对各个模块进行样式设计，确保界面美观且响应迅速。
   * 实现响应式设计，使系统在不同设备上都能有良好的显示效果。
2. **后端实现**：
   * 使用Express框架搭建Node.js服务器，配置中间件以处理JSON数据和跨域请求。
   * 基于MVC架构设计User、Note和Todo三个模型，定义数据库表结构和关联关系。
   * 实现用户注册、登录、信息查询和更新的API接。
   * 实现笔记管理的API接口，包括创建、查询、更新和删除笔记的功能。确保API的安全性和数据完整性。
   * 实现待办事项管理的API接口，包括添加、查询、标记口完成和删除待办事项的功能，确保操作的准确性和数据的实时更新。
   * 使用SQLite作为数据存储，配置数据库连接和模型的映射。

#### **系统分析**

系统设计中，前端和后端各自实现了不同的功能模块，彼此通过API进行数据交互。在前端，通过Vue Router实现页面之间的跳转，通过Pinia管理应用的全局状态，确保各模块的数据一致性和操作的实时性。通过TailwindCSS进行样式设计，提高了系统的用户体验，并且实现了响应式设计，使得系统在不同设备上都有良好的表现。

在后端，通过Node.js和Express框架搭建服务器，基于MVC架构设计了系统的核心模型和路由，确保了代码的结构化和模块化。用户管理部分使用bcrypt进行密码加密存储，确保用户数据的安全性。使用JWT进行身份验证，确保API请求的安全。笔记和待办事项管理部分，通过设计完善的API接口，实现了数据的增删改查功能，并确保数据的实时性和一致性。

在整个系统开发过程中，通过需求分析、系统设计和功能实现，掌握了分布式web应用开发的技术基础，了解了前后端的运行环境，熟练使用了ECMAScript/JavaScript核心语法，并通过Node.js和Vue的应用开发，实现了一个功能完整的个人数字助理系统。