深 圳 大 学 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 岗位实践 | | |
| 实验名称 | 基础功能开发 | | |
| 学院 | 计算机与软件学院 | | |
| 专业 | 软件工程 | | |
| 指导教师 | 黄耀东 | | |
| 报告人 | 江浩 | 学号 | 2022150083 |
| 实验时间 | 2024.10.27 | | |
| 提交时间 | 2024.10.31 | | |

教务处制

|  |
| --- |
| 实验目的与要求：完成基本功能的迭代，我负责后端部分：项目管理模块 项目管理逻辑 - 包含项目任务的增删查改功能，组长发布任务，管理员可查看所有任务。 代码仓库管理 - 实现GIT集成，提供API支持用户组长创建仓库、用户查看代码。 网盘资源管理 - 实现文件的上传、下载、删除和权限控制，为项目和课程资料提供存储支持。 配合重点： 确保项目管理页面和代码仓库功能的互通，支持用户分配任务、查看任务进展及代码仓库的使用权限，管理员可以随时监控任务和进度。 |
| 实验过程及内容：  **1. 项目管理模块**  **功能：**   * **增删查改任务** * **组长发布任务** * **管理员查看所有任务**   **示例代码（使用 Go 和 Gin 框架）**  **package main**  **import (**  **"net/http"**  **"github.com/gin-gonic/gin"**  **)**  **// Task 结构体定义**  **type Task struct {**  **ID int `json:"id"`**  **Title string `json:"title"`**  **Description string `json:"description"`**  **Status string `json:"status"` // e.g., "pending", "in\_progress", "completed"**  **}**  **// 存储任务**  **var tasks []Task**  **var nextID = 1**  **// 增加任务**  **func createTask(c \*gin.Context) {**  **var newTask Task**  **if err := c.ShouldBindJSON(&newTask); err != nil {**  **c.JSON(http.StatusBadRequest, gin.H{"error": err.Error()})**  **return**  **}**  **newTask.ID = nextID**  **nextID++**  **tasks = append(tasks, newTask)**  **c.JSON(http.StatusCreated, newTask)**  **}**  **// 查看所有任务**  **func getTasks(c \*gin.Context) {**  **c.JSON(http.StatusOK, tasks)**  **}**  **// 删除任务**  **func deleteTask(c \*gin.Context) {**  **id := c.Param("id")**  **for i, task := range tasks {**  **if task.ID == id {**  **tasks = append(tasks[:i], tasks[i+1:]...)**  **c.JSON(http.StatusOK, gin.H{"message": "task deleted"})**  **return**  **}**  **}**  **c.JSON(http.StatusNotFound, gin.H{"error": "task not found"})**  **}**  **// 主函数**  **func main() {**  **r := gin.Default()**  **r.POST("/tasks", createTask)**  **r.GET("/tasks", getTasks)**  **r.DELETE("/tasks/:id", deleteTask)**  **r.Run(":8080")**  **}**  **2. 代码仓库管理**  **功能：**   * **用户组长创建仓库** * **用户查看代码**   **代码（使用 GitHub API 进行集成）**  **import (**  **"net/http"**  **"encoding/json"**  **"bytes"**  **)**  **func createRepo(c \*gin.Context) {**  **type Repo struct {**  **Name string `json:"bill"`**  **Description string `json:"homework description"`**  **}**  **var repo Repo**  **if err := c.ShouldBindJSON(&repo); err != nil {**  **c.JSON(http.StatusBadRequest, gin.H{"error": err.Error()})**  **return**  **}**  **// 调用 GitHub API 创建仓库**  **url := "https://api.github.com/user/repos"**  **jsonData, \_ := json.Marshal(repo)**  **req, \_ := http.NewRequest("POST", url, bytes.NewBuffer(jsonData))**  **req.Header.Set("Authorization", "token YOUR\_GITHUB\_TOKEN")**  **client := &http.Client{}**  **resp, err := client.Do(req)**  **if err != nil {**  **c.JSON(http.StatusInternalServerError, gin.H{"error": "Unable to create repo"})**  **return**  **}**  **defer resp.Body.Close()**  **if resp.StatusCode == http.StatusCreated {**  **c.JSON(http.StatusCreated, repo)**  **} else {**  **c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to create repo"})**  **}**  **}**  **3. 网盘资源管理**  **功能：**   * **文件上传、下载、删除和权限控制**   **示例代码：**  **import (**  **"github.com/gin-gonic/gin"**  **"net/http"**  **"os"**  **)**  **func uploadFile(c \*gin.Context) {**  **file, \_ := c.FormFile("file")**  **c.SaveUploadedFile(file, "./uploads/"+file.Filename)**  **c.JSON(http.StatusOK, gin.H{"status": "file uploaded"})**  **}**  **func downloadFile(c \*gin.Context) {**  **filename := c.Param("filename")**  **c.File("./uploads/" + filename)**  **}**  **func deleteFile(c \*gin.Context) {**  **filename := c.Param("filename")**  **os.Remove("./uploads/" + filename)**  **c.JSON(http.StatusOK, gin.H{"status": "file deleted"})**  **}**  **func main() {**  **r := gin.Default()**  **r.POST("/upload", uploadFile)**  **r.GET("/download/:filename", downloadFile)**  **r.DELETE("/delete/:filename", deleteFile)**  **r.Run(":8080")**  **}**  **4. 功能互通**   * **任务和代码仓库的互通：在任务结构体中可以添加字段来存储相关代码仓库的信息，通过任务 ID 查询相关代码仓库状态。** * **权限控制：在任务和文件的管理中增加权限检查，确保只有组长和管理员可以进行特定操作。** |

深圳大学学生实验报告用纸

|  |
| --- |
| 实验结论：  **由于小组还未具体进行职能分组，这里简单展示了如何搭建一个基本的项目管理模块及相关功能，后续下次课我们组具体细分完职能后我会根据具体需求扩展和完善这些功能，如添加数据库支持、用户认证等等的迭代** |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。