# 授权委托书

中煤内蒙古能源有限公司绩效考核系统建设项目

报 价 文 件

询价编码：XJ20250908909

供应商： （盖章）

日 期： 年 月 日

目 录

一、授权委托书  
二、报价部分  
三、商务部分  
四、技术部分  
五、偏离表  
六、供应商廉洁承诺书

## （一）授权委托书

本人（姓名）系（供应商名称）的法定代表人（单位负责人），现委托（姓名）为我方代理人。代理人根据授权，以我方名义签署、澄清确认、递交、撤回、修改 采购项目报价文件、签订合同和处理有关事宜，其法律后果由我方承担。

委托期限： 。

代理人无转委托权。

附：法定代表人（单位负责人）身份证复印件及委托代理人身份证复印件

注：本授权委托书需由供应商加盖单位公章并由其法定代表人（单位负责人）签字。

供应商：（单位公章）

法定代表人（单位负责人）：（签字）

委托代理人：（签字）

年 月 日

注：如供应商法定代表人参加采购行为，只需附其身份证复印件。

## （二）报价部分

（1）报价说明

1.1 本说明应与供应商须知、合同条款等文件一起参照阅读。

1.2 除合同另有规定外，报价应包括供应商为完成本合同规定的工作所承担的全部费用，包括成本、税金、利润等，并考虑了应由供应商承担的义务、责任和风险所发生的费用。

（2）报价表

2.1 报价汇总表（格式）

**报价表**

单位：人民币元

| 序号 | 项目 | 报价（元） | 说明 |
| --- | --- | --- | --- |
| 1 | XXXX项目 |  | 报价为含税价税率 % |
| 备注：报价含税费 |  |  |  |

供应商名称： （盖章）

年 月 日

2.2 分项报价表

| 序号 | 项目 | 报价（元） | 说明 |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 备注：报价含税费。 |  |  |  |

## （三）商务部分

商务部分摘要表

| 供应商名称 |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 注册地址 |  | 邮政编码 |  |  |  |  |
| 联系方式 | 联系人 |  | 电话 |  |  |  |
| 传真 |  | 网址 |  |  |  |  |
| 股权结构 | XX：A%；YY：B%； |  |  |  |  |  |
| 法定代表人 | 姓名 |  | 技术职称 |  | 电话 |  |
| 技术负责人 | 姓名 |  | 技术职称 |  | 电话 |  |
| 成立时间 |  | 员工总人数： |  |  |  |  |
| 企业资质等级 |  | 其中 | 项目经理 |  |  |  |
| 营业执照号 |  | 高级职称人员 |  |  |  |  |
| 注册资本 |  | 中级职称人员 |  |  |  |  |
| 开户银行 |  | 初级职称人员 |  |  |  |  |
| 账号 |  | 技工 |  |  |  |  |
| 经营范围 |  |  |  |  |  |  |
| 类似业绩列表 |  |  |  |  |  |  |
| 备注 |  |  |  |  |  |  |

后附企业法人营业执照副本、资质证书、类似业绩合同扫描件及供应商认为需要提供或询价文件要求提供的其他资料影印件。

供应商名称： （盖章）

日 期：

## （四）技术部分

### （1）项目理解与需求分析

#### 一、项目背景分析

中煤内蒙古能源有限公司作为大型能源企业，需要建立完善的绩效考核体系来提升管理效率和员工积极性。本项目旨在建设一套完整的绩效考核系统，实现：

1、组织架构数字化管理：建立完整的组织架构体系，支持部门、职位、员工的层级管理

2、绩效考核流程化：实现考核周期的标准化管理，支持多种评价关系

3、权重配置科学化：基于职级体系的权重配置，确保评价的公平性和科学性

4、数据统计分析：提供完整的考核数据分析和报表功能

#### 二、核心需求分析

（1）组织架构管理需求

部门管理：支持多层级部门结构，适应企业组织架构

职位管理：建立标准化的职位体系，支持职级管理

员工管理：完整的员工档案管理，支持人员变动跟踪

权限管理：基于角色的权限控制，确保数据安全

（2）绩效考核需求

考核周期管理：支持年度、季度、月度等不同考核周期

评价关系配置：支持上级评下级、同级互评、下级评上级等多种评价关系

权重配置：基于职级的权重配置，确保评价的公平性

任务自动生成：根据规则自动生成考核任务，提高效率

在线评分：支持在线评分和评价意见记录

结果统计：提供完整的考核结果统计和分析

（3）系统集成需求

#### 三、技术需求分析

（1）性能要求

并发用户：支持1000+并发用户同时使用

响应时间：页面响应时间 < 2秒，API响应时间 < 500ms

数据处理：支持大规模数据处理，单次处理10万+记录

系统可用性：99.9%以上系统可用性

（2）安全要求

数据安全：敏感数据加密存储和传输

访问控制：基于角色的细粒度权限控制

审计日志：完整的操作审计日志

备份恢复：定期数据备份和快速恢复

（3）扩展性要求

模块化设计：支持功能模块的独立部署和升级

API开放：提供完整的API接口，支持第三方集成

配置化：支持业务规则的配置化管理

多租户：支持多组织、多部门的数据隔离

### （2）技术解决方案

#### 四、整体技术架构

本项目采用现代化的微服务架构，基于双系统设计理念，构建完整的绩效考核管理平台：

（1）系统架构设计

┌─────────────────────────────────────────────────────────────────┐  
│ 用户接入层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 组织架构系统前端 (Vue.js 3) │ 绩效考核系统前端 (Vue.js 3) │  
│ - 部门管理界面 │ - 考核周期管理界面 │  
│ - 员工管理界面 │ - 评分管理界面 │  
│ - 权限管理界面 │ - 结果统计界面 │  
│ - 工作流配置界面 │ - 数据导出界面 │  
└─────────────────────────────────────────────────────────────────────┘  
 │  
┌─────────────────────────────────────────────────────────────────┐  
│ API网关层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 路由管理 │ 限流控制 │ 缓存机制 │ 数据转换 │ 安全认证 │  
└─────────────────────────────────────────────────────────────────┘  
 │  
┌─────────────────────────────────────────────────────────────────┐  
│ 业务服务层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 组织架构服务 │ 绩效考核服务 │ 权限管理服务 │ 集成服务 │  
└─────────────────────────────────────────────────────────────────┘  
 │  
┌─────────────────────────────────────────────────────────────────┐  
│ 数据存储层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 主数据库 │ 缓存数据库 │ 文件存储 │ 日志存储 │ 备份存储 │  
└─────────────────────────────────────────────────────────────────┘

（2）核心技术栈

前端技术栈：

Vue.js 3.4+：渐进式JavaScript框架，提供响应式数据绑定

TypeScript 5.0+：类型安全的JavaScript超集，提高代码质量

Element Plus 2.7+：企业级UI组件库，提供丰富的组件

ECharts 6.0+：数据可视化图表库，支持复杂图表展示

Vite 5.0+：现代化构建工具，提供快速的开发体验

Pinia 2.1+：Vue状态管理库，替代Vuex

Vue Router 4.3+：官方路由管理器，支持路由懒加载

后端技术栈：

Django 5.2.1：Python Web框架，提供完整的MVC架构

Django REST Framework 3.15.2：RESTful API框架，提供强大的API功能

PostgreSQL 14+：企业级关系型数据库，支持高并发

Redis 7.0+：内存数据库，用于缓存和会话管理

Celery 5.3+：分布式任务队列，处理异步任务

Gunicorn：WSGI HTTP服务器，用于生产环境部署

Nginx：反向代理服务器，提供负载均衡和静态文件服务

开发工具链：

Git：版本控制系统

Docker & Docker Compose：容器化部署

pytest：Python测试框架

ESLint & Prettier：代码质量检查工具

Jest & Vue Test Utils：前端测试框架

（3）系统集成方案

数据同步机制：

实时同步：基于WebSocket的实时数据同步

批量同步：定时任务批量同步数据

增量同步：只同步变更的数据，提高效率

冲突解决：智能的数据冲突检测和解决机制

API接口设计：

RESTful API：遵循REST设计原则的API接口

GraphQL支持：提供GraphQL接口，支持灵活的数据查询

API版本管理：支持多版本API并存

接口文档：自动生成的API文档，支持在线测试

安全机制：

JWT认证：基于JSON Web Token的身份认证

RBAC权限控制：基于角色的访问控制

数据加密：敏感数据AES-256加密存储

HTTPS传输：全站HTTPS加密传输

SQL注入防护：使用ORM防止SQL注入攻击

XSS防护：前端输入过滤和转义

（4）性能优化方案

前端性能优化：

代码分割：路由级别的代码分割，减少初始加载时间

懒加载：组件和图片的懒加载，提高页面加载速度

缓存策略：合理的浏览器缓存策略

CDN加速：静态资源CDN加速

压缩优化：代码压缩和图片优化

后端性能优化：

数据库优化：索引优化、查询优化、连接池配置

缓存机制：Redis缓存热点数据，减少数据库压力

异步处理：Celery异步任务处理，提高响应速度

负载均衡：多实例负载均衡，提高并发处理能力

数据库读写分离：主从数据库配置，提高查询性能

（5）监控与运维

系统监控：

应用监控：APM应用性能监控

服务器监控：CPU、内存、磁盘、网络监控

数据库监控：数据库性能监控和慢查询分析

日志监控：集中化日志收集和分析

运维自动化：

CI/CD流水线：自动化构建、测试、部署

容器编排：Docker Swarm或Kubernetes容器编排

自动扩缩容：基于负载的自动扩缩容

备份恢复：自动化数据备份和恢复机制

### （3）系统功能详细说明

#### 五、组织架构中台系统

（1）组织架构管理

部门管理功能：

树形结构管理：支持无限层级的部门树形结构，适应复杂的企业组织架构

部门信息管理：完整的部门信息维护，包括部门名称、编码、类型、负责人等

部门层级控制：支持部门层级的动态调整和重新组织

部门权限管理：基于部门的权限控制，确保数据安全

部门统计分析：提供部门人员统计、层级分析等报表功能

职位管理功能：

双维度职位体系：管理层级（高层、中层、基层）+ 职位类型（管理、技术、业务）

职位级别管理：13级职位级别体系，从普通员工到董事长

职位模板管理：预置职位模板，快速创建标准化职位

职位权限配置：基于职位的权限配置，支持细粒度权限控制

职位变更跟踪：完整的职位变更历史记录和审批流程

员工管理功能：

员工档案管理：完整的员工信息管理，包括基本信息、联系方式、教育背景等

员工状态管理：在职、离职、休假等状态管理

员工关系管理：上下级关系、汇报关系、协作关系管理

员工权限管理：基于员工角色的权限分配和管理

员工统计分析：员工分布、流动率、结构分析等统计功能

（2）工作流自动化

工作流规则配置：

可视化配置：图形化的工作流规则配置界面，无需编程

触发条件设置：支持事件触发、时间触发、条件触发等多种触发方式

执行动作配置：通知、审批、数据同步、权限变更等多种执行动作

规则优先级：支持规则优先级设置和冲突解决

规则测试：提供规则测试功能，确保规则正确性

审批流程管理：

多级审批：支持多级审批流程配置

审批人指定：支持固定审批人、角色审批人、动态审批人

审批时限控制：设置审批时限和超时处理机制

审批意见记录：完整的审批意见和决策记录

审批统计分析：审批效率、通过率等统计分析

通知系统：

多渠道通知：支持邮件、短信、系统内通知、微信等多种通知方式

通知模板管理：可配置的通知模板，支持变量替换

通知规则配置：基于条件的自动通知规则

通知状态跟踪：通知发送状态和接收确认跟踪

通知统计分析：通知送达率、响应率等统计分析

（3）数据标准化

数据字典管理：

字典分类管理：支持多种字典分类，如员工状态、学历层次、技能等级等

字典项管理：完整的字典项增删改查功能

字典层级结构：支持字典项的层级结构管理

字典状态控制：支持字典项的启用/禁用状态控制

字典导入导出：支持字典数据的批量导入导出

系统配置管理：

配置分类管理：组织架构、职位、员工、工作流、集成、安全、通知等配置分类

配置项管理：支持字符串、整数、布尔值、JSON对象、列表等多种数据类型

配置加密存储：敏感配置的加密存储

配置版本管理：配置变更的版本控制和回滚

配置导入导出：支持配置的批量导入导出

#### 六、绩效考核系统

（1）考核周期管理

周期创建与配置：

周期基本信息：考核周期名称、描述、时间范围等基本信息设置

考核规则关联：关联考核规则和评价指标

参与人员设置：设置参与考核的人员范围

考核状态管理：草稿、进行中、已完成、已取消等状态管理

周期复制功能：支持基于历史周期创建新周期

时间管理：

灵活时间设置：支持年度、季度、月度、半年度等多种考核周期

时间节点控制：设置考核开始时间、结束时间、评分截止时间等

时间提醒功能：自动提醒考核时间节点

时间调整功能：支持考核时间的动态调整

历史时间查询：查询历史考核时间安排

（2）考核规则配置

评价关系类型：

本部门上级评下级：部门内部上下级评价关系

全公司上级评下级：跨部门上级评价下级关系

同级互评：同级别员工相互评价关系

下级评上级：下级评价上级关系

自评：员工自我评价

跨部门评价：跨部门评价关系

自定义关系：完全自定义的评价关系

评价范围配置：

部门范围：按部门设置评价范围

职位范围：按职位级别设置评价范围

人员范围：按具体人员设置评价范围

时间范围：按时间设置评价范围

条件范围：按条件设置评价范围

评价规则算法：

关系匹配算法：智能匹配评价人和被评价人

权重分配算法：基于职级的权重分配算法

人数控制算法：控制每种关系类型的评价人数

权限验证算法：验证评价权限的合法性

冲突解决算法：解决评价关系冲突

（3）职级权重配置

权重体系设计：

高层正职（董事长/总经理）：权重 1.8

高层副职（副总经理）：权重 1.6

高层助理（总经理助理）：权重 1.5

中层正职（部门经理）：权重 1.2

中层副职（部门副经理）：权重 1.1

中层助理（经理助理）：权重 1.0

基层正职（主管）：权重 1.0

基层副职（副主管）：权重 0.9

基层助理（助理）：权重 0.9

普通员工：权重 0.9

权重配置功能：

默认权重设置：系统预置的默认权重配置

自定义权重：支持自定义权重配置

权重批量更新：支持权重的批量更新

权重验证：权重配置的合法性验证

权重历史记录：权重变更的历史记录

（4）任务自动生成

智能生成算法：

关系识别算法：自动识别评价关系

任务分配算法：智能分配考核任务

冲突检测算法：检测和解决任务冲突

优化算法：优化任务分配效率

验证算法：验证生成任务的正确性

任务管理功能：

任务状态跟踪：待评价、进行中、已完成、已过期等状态

任务分配管理：支持任务的重新分配

任务优先级：设置任务优先级

任务提醒：自动提醒未完成的任务

任务统计分析：任务完成情况统计分析

考核码管理：

唯一码生成：生成唯一的考核码

码格式配置：支持自定义考核码格式

码有效期管理：设置考核码有效期

码使用跟踪：跟踪考核码使用情况

码批量生成：支持考核码的批量生成

（5）评分管理

在线评分功能：

评分界面：直观的评分界面设计

指标评分：支持多指标评分

权重显示：显示评价人权重信息

评分验证：评分范围和格式验证

评分保存：自动保存评分草稿

评价意见管理：

意见填写：支持评价意见的填写

意见模板：提供评价意见模板

意见长度控制：控制意见长度

意见审核：评价意见的审核功能

意见统计分析：评价意见的统计分析

权重计算：

自动计算：系统自动计算加权评分

计算规则：基于权重的计算规则

计算验证：计算结果的验证

计算历史：计算过程的历史记录

计算报表：计算结果的报表展示

（6）数据导出

Excel导出功能：

考核码导出：导出考核码分发表

评分结果导出：导出评分结果统计

评价意见导出：导出评价意见汇总

统计分析导出：导出统计分析报表

自定义导出：支持自定义导出格式

报表生成：

实时报表：实时生成各类报表

定时报表：定时生成报表

报表模板：可配置的报表模板

报表分发：报表的自动分发

报表归档：报表的历史归档

#### 七、系统集成功能

（1）集成系统管理

外部系统连接：

系统注册：注册外部系统信息

连接配置：配置系统连接参数

认证配置：配置系统认证方式

权限配置：配置系统访问权限

状态监控：监控系统连接状态

数据同步管理：

同步规则配置：配置数据同步规则

同步频率设置：设置数据同步频率

同步状态监控：监控数据同步状态

同步日志记录：记录数据同步日志

同步错误处理：处理数据同步错误

（2）API网关

路由管理：

动态路由：支持动态路由配置

路由规则：配置路由匹配规则

路由负载均衡：支持路由负载均衡

路由监控：监控路由使用情况

路由优化：优化路由性能

限流控制：

请求限流：限制API请求频率

用户限流：限制用户请求频率

IP限流：限制IP请求频率

限流策略：配置限流策略

限流监控：监控限流情况

（3）数据同步

实时同步：

WebSocket同步：基于WebSocket的实时同步

消息队列同步：基于消息队列的同步

事件驱动同步：基于事件的同步机制

增量同步：只同步变更数据

冲突解决：解决数据同步冲突

批量同步：

定时同步：定时批量同步数据

手动同步：手动触发批量同步

同步策略：配置同步策略

同步监控：监控同步进度

同步报告：生成同步报告

## （五）技术架构与实现

### （4）系统架构设计

#### 八、系统架构设计

（1）整体架构图

┌─────────────────────────────────────────────────────────────────┐  
│ 用户接入层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 组织架构系统前端 (Vue.js 3) │ 绩效考核系统前端 (Vue.js 3) │  
│ - 部门管理界面 │ - 考核周期管理界面 │  
│ - 员工管理界面 │ - 评分管理界面 │  
│ - 权限管理界面 │ - 结果统计界面 │  
│ - 工作流配置界面 │ - 数据导出界面 │  
└─────────────────────────────────────────────────────────────────────┘  
 │  
┌─────────────────────────────────────────────────────────────────┐  
│ API网关层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 路由管理 │ 限流控制 │ 缓存机制 │ 数据转换 │ 安全认证 │  
└─────────────────────────────────────────────────────────────────┘  
 │  
┌─────────────────────────────────────────────────────────────────┐  
│ 业务服务层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 组织架构服务 │ 绩效考核服务 │ 权限管理服务 │ 集成服务 │  
└─────────────────────────────────────────────────────────────────┘  
 │  
┌─────────────────────────────────────────────────────────────────┐  
│ 数据存储层 │  
├─────────────────────────────────────────────────────────────────┤  
│ 主数据库 │ 缓存数据库 │ 文件存储 │ 日志存储 │ 备份存储 │  
└─────────────────────────────────────────────────────────────────┘

（2）数据库设计

核心数据表结构：

组织架构系统数据表：

-- 部门表  
 CREATE TABLE departments (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(100) NOT NULL,  
 code VARCHAR(50) UNIQUE NOT NULL,  
 parent\_id INTEGER REFERENCES departments(id),  
 level INTEGER NOT NULL,  
 type VARCHAR(50),  
 manager\_id INTEGER,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

-- 职位表  
 CREATE TABLE positions (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(100) NOT NULL,  
 code VARCHAR(50) UNIQUE NOT NULL,  
 level INTEGER NOT NULL,  
 management\_level VARCHAR(20),  
 department\_id INTEGER REFERENCES departments(id),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

-- 员工表  
 CREATE TABLE employees (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(100) NOT NULL,  
 employee\_id VARCHAR(50) UNIQUE NOT NULL,  
 email VARCHAR(100),  
 phone VARCHAR(20),  
 department\_id INTEGER REFERENCES departments(id),  
 position\_id INTEGER REFERENCES positions(id),  
 supervisor\_id INTEGER REFERENCES employees(id),  
 status VARCHAR(20) DEFAULT 'active',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

绩效考核系统数据表：

-- 考核周期表  
 CREATE TABLE evaluation\_cycles (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(200) NOT NULL,  
 description TEXT,  
 start\_date DATE NOT NULL,  
 end\_date DATE NOT NULL,  
 status VARCHAR(20) DEFAULT 'draft',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

-- 考核任务表  
 CREATE TABLE evaluation\_tasks (  
 id SERIAL PRIMARY KEY,  
 cycle\_id INTEGER REFERENCES evaluation\_cycles(id),  
 evaluator\_id INTEGER REFERENCES employees(id),  
 evaluatee\_id INTEGER REFERENCES employees(id),  
 relation\_type VARCHAR(20) NOT NULL,  
 evaluation\_code VARCHAR(20) UNIQUE,  
 weight DECIMAL(5,2) DEFAULT 1.00,  
 status VARCHAR(20) DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

-- 评分记录表  
 CREATE TABLE evaluation\_scores (  
 id SERIAL PRIMARY KEY,  
 task\_id INTEGER REFERENCES evaluation\_tasks(id),  
 indicator\_id INTEGER,  
 score INTEGER NOT NULL,  
 weighted\_score DECIMAL(5,2),  
 comment TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

（3）API接口设计

RESTful API规范：

组织架构API：

# 部门管理

GET /api/departments/ # 获取部门列表  
 POST /api/departments/ # 创建部门  
 GET /api/departments/{id}/ # 获取部门详情  
 PUT /api/departments/{id}/ # 更新部门  
 DELETE /api/departments/{id}/ # 删除部门

# 员工管理

GET /api/employees/ # 获取员工列表  
 POST /api/employees/ # 创建员工  
 GET /api/employees/{id}/ # 获取员工详情  
 PUT /api/employees/{id}/ # 更新员工  
 DELETE /api/employees/{id}/ # 删除员工

绩效考核API：

# 考核周期管理

GET /api/cycles/ # 获取考核周期列表  
 POST /api/cycles/ # 创建考核周期  
 GET /api/cycles/{id}/ # 获取考核周期详情  
 PUT /api/cycles/{id}/ # 更新考核周期  
 DELETE /api/cycles/{id}/ # 删除考核周期  
 POST /api/cycles/{id}/generate\_tasks/ # 生成考核任务

# 考核任务管理

GET /api/tasks/ # 获取考核任务列表  
 GET /api/tasks/by-code/{code}/ # 根据考核码获取任务  
 POST /api/tasks/{id}/submit\_scores/ # 提交评分

（4）安全架构设计

认证与授权：

JWT Token认证：基于JSON Web Token的无状态认证

RBAC权限模型：基于角色的访问控制

API密钥管理：安全的API密钥生成和管理

会话管理：安全的会话管理和超时控制

数据安全：

数据加密：敏感数据AES-256加密存储

传输加密：HTTPS/TLS 1.3加密传输

数据脱敏：敏感信息的自动脱敏处理

访问日志：完整的访问和操作日志记录

（5）性能优化策略

前端性能优化：

代码分割：路由级别的代码分割，减少初始加载时间

懒加载：组件和图片的懒加载

缓存策略：浏览器缓存和CDN缓存

压缩优化：代码压缩和图片优化

后端性能优化：

数据库优化：索引优化、查询优化、连接池

缓存机制：Redis缓存热点数据

异步处理：Celery异步任务处理

负载均衡：多实例负载均衡

#### 九、技术特性

（1）高性能

响应时间：API响应时间 < 100ms，页面响应时间 < 2秒

并发处理：支持1000+并发用户同时使用

数据处理：支持大规模数据处理，单次处理10万+记录

缓存优化：多级缓存机制，提高系统性能

（2）高可用

系统可用性：99.9%以上系统可用性

故障恢复：自动故障检测和恢复机制

负载均衡：多实例负载均衡，提高系统稳定性

数据备份：定期数据备份和快速恢复

（3）高安全

身份认证：多种认证方式，支持单点登录

权限控制：细粒度权限管理，确保数据安全

数据加密：敏感数据加密存储和传输

审计日志：完整的操作审计日志

（4）高扩展

模块化设计：松耦合模块设计，支持独立部署

API开放：完整的API接口，支持第三方集成

插件机制：支持插件扩展，满足个性化需求

集成支持：与其他系统无缝集成

## （六）项目实施计划

### （5）项目总体时间安排

本项目预计总工期为11周，分为4个主要阶段，确保项目按时、按质、按量完成。

### （6）详细实施计划

#### 十、详细实施计划

（1）第一阶段：需求调研与系统设计（2周）

第1周：需求调研

业务需求调研：深入了解中煤内蒙古能源有限公司的业务流程和考核需求

用户访谈：与HR部门、各部门负责人进行深度访谈

现状分析：分析现有考核流程和系统现状

需求整理：整理和确认详细的功能需求和非功能需求

需求文档：编写详细的需求规格说明书

第2周：系统设计

架构设计：完成系统整体架构设计

数据库设计：设计数据库表结构和关系

接口设计：设计API接口规范

安全设计：设计系统安全方案

设计评审：组织技术评审会议

交付物：

需求规格说明书

系统架构设计文档

数据库设计文档

API接口设计文档

（2）第二阶段：系统开发与测试（6周）

第3-4周：基础功能开发

环境搭建：搭建开发、测试、预生产环境

基础框架：搭建前后端基础框架

核心模块：开发组织架构管理核心功能

数据模型：实现数据库模型和基础CRUD操作

单元测试：编写单元测试用例

第5-6周：核心功能开发

绩效考核模块：开发考核周期、规则配置、任务生成功能

评分管理模块：开发在线评分、权重计算功能

数据导出模块：开发Excel导出、报表生成功能

权限管理模块：开发基于角色的权限控制

集成测试：进行模块间集成测试

第7-8周：高级功能开发

工作流引擎：开发工作流规则配置和执行引擎

系统集成：开发与外部系统的集成功能

数据同步：开发数据同步和冲突解决机制

性能优化：进行系统性能优化

安全加固：实施安全加固措施

交付物：

组织架构管理系统

绩效考核管理系统

系统集成平台

功能测试报告

（3）第三阶段：系统集成与部署（2周）

第9周：系统集成测试

集成测试：进行系统各模块集成测试

性能测试：进行系统性能压力测试

安全测试：进行系统安全渗透测试

兼容性测试：进行浏览器兼容性测试

用户验收测试：组织用户进行验收测试

第10周：生产环境部署

环境准备：准备生产环境服务器和网络

系统部署：部署系统到生产环境

数据迁移：迁移历史数据到新系统

系统配置：配置生产环境参数

监控部署：部署系统监控和日志收集

交付物：

集成测试报告

性能测试报告

安全测试报告

生产环境部署文档

（4）第四阶段：用户培训与上线（1周）

第11周：培训与上线

用户培训：组织系统管理员和最终用户培训

试运行：系统试运行，收集用户反馈

问题修复：修复试运行中发现的问题

正式上线：系统正式上线运行

运维交接：完成运维文档和知识转移

交付物：

用户培训材料

系统操作手册

运维手册

上线系统

#### 十一、项目里程碑与验收标准

| 里程碑 | 时间 | 主要交付物 | 验收标准 | 责任人 |
| --- | --- | --- | --- | --- |
| 需求确认 | 第2周末 | 需求规格说明书 | 客户签字确认 | 项目经理 |
| 设计评审 | 第2周末 | 系统设计文档 | 技术评审通过 | 技术负责人 |
| 功能开发完成 | 第8周末 | 功能模块 | 功能测试通过 | 开发团队 |
| 集成测试完成 | 第9周末 | 测试报告 | 集成测试通过 | 测试团队 |
| 系统上线 | 第11周末 | 上线系统 | 用户验收通过 | 项目经理 |

#### 十二、项目团队组织架构

（1）项目管理团队

项目经理：负责项目整体管理、进度控制、风险管控

经验：5年以上大型项目管理经验

职责：项目计划制定、进度跟踪、风险控制、客户沟通

技术负责人：负责技术方案设计、技术难点攻关

经验：8年以上系统开发经验，3年以上技术管理经验

职责：技术架构设计、技术选型、技术难点解决

（2）开发团队

前端开发工程师（2名）：负责Vue.js前端开发

技能：Vue.js 3、TypeScript、Element Plus、ECharts

职责：前端界面开发、用户体验优化、前端性能优化

后端开发工程师（2名）：负责Django后端开发

技能：Django、Python、PostgreSQL、Redis、Celery

职责：后端API开发、数据库设计、系统集成

全栈开发工程师（1名）：负责前后端协调开发

技能：前后端技术栈、系统集成、性能优化

职责：前后端协调、系统集成、性能调优

（3）测试团队

测试工程师（1名）：负责系统测试

技能：功能测试、性能测试、安全测试、自动化测试

职责：测试用例设计、测试执行、缺陷跟踪

（4）实施团队

实施工程师（2名）：负责系统部署和配置

技能：Linux、Docker、Nginx、数据库管理

职责：环境搭建、系统部署、数据迁移、运维支持

培训师（1名）：负责用户培训

技能：培训设计、用户沟通、文档编写

职责：培训材料编写、用户培训、操作手册编写

#### 十三、质量保证措施

（1）开发质量保证

代码审查：所有代码必须经过同行审查

单元测试：代码覆盖率不低于80%

集成测试：完整的集成测试流程

性能测试：定期进行性能测试和优化

（2）项目管理质量保证

周报制度：每周提交项目进度报告

里程碑评审：每个里程碑进行评审

风险管控：定期识别和评估项目风险

变更管理：严格的变更管理流程

（3）交付质量保证

文档完整性：确保所有交付物完整

功能完整性：确保所有功能按需求实现

性能达标：确保系统性能满足要求

安全合规：确保系统安全合规

## （七）服务承诺与支持

### （7）技术支持体系

#### 十四、技术支持体系

（1）实施阶段支持（项目期间）

现场技术支持：

驻场支持：项目实施期间提供2名技术人员驻场支持

现场培训：提供现场系统使用培训和技术培训

问题解决：现场解决技术问题和系统配置问题

进度跟踪：实时跟踪项目进度，确保按时交付

远程技术支持：

7×24小时支持：提供7×24小时远程技术支持服务

响应时间承诺：

紧急问题：30分钟内响应，2小时内解决

一般问题：2小时内响应，24小时内解决

非紧急问题：4小时内响应，72小时内解决

多渠道支持：电话、邮件、在线客服、远程桌面等多种支持方式

（2）运维阶段支持（系统上线后）

系统监控服务：

7×24小时监控：对系统进行7×24小时不间断监控

性能监控：监控系统性能指标，包括响应时间、并发数、资源使用率等

安全监控：监控系统安全状态，及时发现和处理安全威胁

告警机制：建立完善的告警机制，及时通知相关人员

故障处理服务：

故障响应：故障发生后30分钟内响应

故障定位：快速定位故障原因和影响范围

故障修复：提供故障修复方案并实施

故障报告：提供详细的故障处理报告

系统维护服务：

定期维护：每月进行一次系统维护检查

性能优化：持续进行系统性能优化

安全更新：及时提供安全补丁和更新

数据备份：定期进行数据备份和恢复测试

（3）升级维护服务

功能升级：

需求分析：分析业务需求，制定升级方案

功能开发：根据需求开发新功能

测试验证：对新功能进行充分测试

平滑升级：确保升级过程不影响业务运行

版本升级：

版本规划：制定版本升级计划

兼容性测试：确保新版本与现有系统的兼容性

数据迁移：安全迁移数据到新版本

回滚方案：提供版本回滚方案

### （8）质量保证体系

#### 十五、质量保证体系

（1）开发质量保证

代码质量：

代码规范：严格遵循行业代码规范和最佳实践

代码审查：所有代码必须经过同行审查

静态分析：使用静态代码分析工具检查代码质量

代码覆盖率：单元测试代码覆盖率不低于90%

测试质量：

测试策略：制定完整的测试策略和测试计划

功能测试：100%功能测试覆盖

性能测试：通过性能压力测试，满足性能要求

安全测试：通过安全渗透测试，确保系统安全

兼容性测试：支持主流浏览器和操作系统

（2）交付质量保证

功能完整性：

需求覆盖：100%实现需求规格说明书中的功能

功能验证：每个功能都经过充分验证

用户验收：通过用户验收测试

文档完整：提供完整的技术文档和用户手册

性能质量：

响应时间：页面响应时间 < 2秒，API响应时间 < 500ms

并发处理：支持1000+并发用户同时使用

数据处理：支持大规模数据处理

系统可用性：99.9%以上系统可用性

安全质量：

数据安全：敏感数据加密存储和传输

访问控制：基于角色的细粒度权限控制

安全合规：通过安全合规检查

审计日志：完整的操作审计日志

（3）服务质量保证

响应质量：

响应及时：严格按照承诺时间响应客户需求

解决有效：提供有效的解决方案

跟踪反馈：及时跟踪问题解决情况并反馈

持续改进：根据客户反馈持续改进服务质量

沟通质量：

沟通顺畅：建立顺畅的沟通渠道

信息透明：及时提供项目进度和问题处理信息

专业服务：提供专业的技术服务和支持

客户满意：确保客户满意度95%以上

### （9）服务等级协议（SLA）

#### 十六、服务等级协议（SLA）

（1）系统可用性

目标可用性：99.9%以上

计划停机时间：每月不超过4.4小时

非计划停机时间：每月不超过0.1小时

故障恢复时间：平均故障恢复时间 < 30分钟

（2）性能指标

页面响应时间：95%的页面响应时间 < 2秒

API响应时间：95%的API响应时间 < 500ms

并发用户数：支持1000+并发用户

数据处理能力：支持10万+记录的单次处理

（3）支持响应时间

紧急问题：30分钟内响应，2小时内解决

高优先级问题：2小时内响应，24小时内解决

一般问题：4小时内响应，72小时内解决

低优先级问题：8小时内响应，5个工作日内解决

### （10）服务团队

#### 十七、服务团队

（1）技术支持团队

技术支持经理：负责技术支持团队管理

高级技术支持工程师（2名）：负责复杂技术问题解决

技术支持工程师（3名）：负责日常技术支持服务

运维工程师（2名）：负责系统运维和监控

（2）服务流程

问题接收：多渠道接收客户问题

问题分类：按优先级和类型分类问题

问题分配：分配给相应的技术支持人员

问题处理：按照SLA要求处理问题

问题跟踪：跟踪问题处理进度

问题关闭：确认问题解决后关闭问题

服务评价：收集客户对服务的评价

## （八）报价明细

### （11）软件开发费用详细分解

#### 十八、软件开发费用详细分解

（1）组织架构中台系统（15.0万元）

| 功能模块 | 开发工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 部门管理模块 | 2人周 | 2.0 | 2.0 | 树形结构、权限控制 |
| 职位管理模块 | 2人周 | 2.0 | 2.0 | 双维度职位体系 |
| 员工管理模块 | 3人周 | 3.0 | 3.0 | 员工档案、关系管理 |
| 权限管理模块 | 3人周 | 3.0 | 3.0 | 基于角色的权限控制 |
| 工作流引擎 | 3人周 | 3.0 | 3.0 | 可视化工作流配置 |
| 数据字典管理 | 1人周 | 1.0 | 1.0 | 标准化数据管理 |
| 系统配置管理 | 1人周 | 1.0 | 1.0 | 灵活配置参数管理 |
| 小计 |  |  | 15.0 |  |

（2）绩效考核系统（20.0万元）

| 功能模块 | 开发工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 考核周期管理 | 2人周 | 2.0 | 2.0 | 周期创建、时间管理 |
| 考核规则配置 | 3人周 | 3.0 | 3.0 | 多种评价关系配置 |
| 职级权重配置 | 2人周 | 2.0 | 2.0 | 权重体系设计 |
| 任务自动生成 | 3人周 | 3.0 | 3.0 | 智能任务分配算法 |
| 评分管理模块 | 3人周 | 3.0 | 3.0 | 在线评分、权重计算 |
| 数据导出模块 | 2人周 | 2.0 | 2.0 | Excel导出、报表生成 |
| 统计分析模块 | 2人周 | 2.0 | 2.0 | 考核结果统计分析 |
| 考核码管理 | 1人周 | 1.0 | 1.0 | 唯一码生成、跟踪 |
| 评价意见管理 | 1人周 | 1.0 | 1.0 | 意见填写、审核 |
| 系统集成接口 | 1人周 | 1.0 | 1.0 | 与组织架构系统集成 |
| 小计 |  |  | 20.0 |  |

（3）系统集成平台（8.0万元）

| 功能模块 | 开发工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| API网关 | 2人周 | 2.0 | 2.0 | 路由管理、限流控制 |
| 数据同步引擎 | 2人周 | 2.0 | 2.0 | 实时同步、批量同步 |
| 集成系统管理 | 1人周 | 1.0 | 1.0 | 外部系统连接管理 |
| 监控告警系统 | 1人周 | 1.0 | 1.0 | 系统监控、告警机制 |
| 安全认证模块 | 1人周 | 1.0 | 1.0 | 统一认证、权限控制 |
| 缓存管理模块 | 1人周 | 1.0 | 1.0 | Redis缓存、性能优化 |
| 小计 |  |  | 8.0 |  |

软件开发费用总计：43.0万元

### （12）实施服务费用详细分解

#### 十九、实施服务费用详细分解

（1）需求调研与设计（3.0万元）

| 服务内容 | 工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 业务需求调研 | 1人周 | 1.0 | 1.0 | 深入了解业务流程 |
| 用户访谈 | 0.5人周 | 0.5 | 0.5 | 与关键用户深度访谈 |
| 需求分析整理 | 0.5人周 | 0.5 | 0.5 | 需求规格说明书编写 |
| 系统架构设计 | 1人周 | 1.0 | 1.0 | 技术架构、数据库设计 |
| 小计 |  |  | 3.0 |  |

（2）系统开发与测试（8.0万元）

| 服务内容 | 工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 开发环境搭建 | 0.5人周 | 0.5 | 0.5 | 开发、测试环境配置 |
| 功能开发指导 | 2人周 | 2.0 | 2.0 | 开发过程技术指导 |
| 单元测试 | 1人周 | 1.0 | 1.0 | 测试用例编写、执行 |
| 集成测试 | 1人周 | 1.0 | 1.0 | 模块间集成测试 |
| 性能测试 | 0.5人周 | 0.5 | 0.5 | 性能压力测试 |
| 安全测试 | 0.5人周 | 0.5 | 0.5 | 安全渗透测试 |
| 缺陷修复 | 1人周 | 1.0 | 1.0 | 测试缺陷修复 |
| 代码审查 | 0.5人周 | 0.5 | 0.5 | 代码质量审查 |
| 文档编写 | 1人周 | 1.0 | 1.0 | 技术文档编写 |
| 小计 |  |  | 8.0 |  |

（3）系统部署与上线（2.0万元）

| 服务内容 | 工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 生产环境搭建 | 0.5人周 | 0.5 | 0.5 | 服务器配置、网络配置 |
| 系统部署 | 0.5人周 | 0.5 | 0.5 | 应用部署、配置 |
| 数据迁移 | 0.5人周 | 0.5 | 0.5 | 历史数据迁移 |
| 系统配置 | 0.5人周 | 0.5 | 0.5 | 生产环境参数配置 |
| 小计 |  |  | 2.0 |  |

（4）用户培训（1.0万元）

| 服务内容 | 工作量 | 单价（万元） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 培训材料编写 | 0.5人周 | 0.5 | 0.5 | 用户手册、操作指南 |
| 管理员培训 | 0.5人周 | 0.5 | 0.5 | 系统管理员培训 |
| 小计 |  |  | 1.0 |  |

实施服务费用总计：14.0万元

### （13）运维服务费用详细分解

#### 二十、运维服务费用详细分解

（1）第一年运维服务（4.5万元）

| 服务内容 | 服务周期 | 单价（万元/年） | 小计（万元） | 说明 |
| --- | --- | --- | --- | --- |
| 技术支持服务 | 1年 | 2.0 | 2.0 | 7×24小时技术支持 |
| 系统维护服务 | 1年 | 1.5 | 1.5 | 系统维护、性能优化 |
| 安全服务 | 1年 | 1.0 | 1.0 | 安全监控、防护 |
| 小计 |  |  | 4.5 |  |

### （14）费用汇总表

#### 二十一、费用汇总表

| 费用类别 | 金额（万元） | 占比 | 说明 |
| --- | --- | --- | --- |
| 软件开发费用 | 43.0 | 70.5% | 组织架构系统15万+绩效系统20万+集成平台8万 |
| 实施服务费用 | 14.0 | 23.0% | 需求调研3万+开发测试8万+部署上线2万+培训1万 |
| 运维服务费用 | 4.5 | 7.4% | 第一年运维服务 |
| 项目总价 | 61.5 | 100% | 不含后续年度运维费用 |

### （15）付款方式与条件

#### 二十二、付款方式与条件

（1）付款计划

| 付款阶段 | 付款比例 | 付款金额（万元） | 付款条件 | 时间节点 |
| --- | --- | --- | --- | --- |
| 合同签订 | 30% | 18.45 | 合同签订后7个工作日内 | 项目启动 |
| 需求确认 | 20% | 12.3 | 需求规格说明书确认通过 | 第2周末 |
| 开发完成 | 30% | 18.45 | 系统开发完成并功能测试通过 | 第8周末 |
| 系统上线 | 15% | 9.225 | 系统正式上线运行 | 第11周末 |
| 质保期满 | 5% | 3.075 | 质保期满无质量问题 | 上线后3个月 |

（2）付款条件说明

合同签订款：合同正式签署后7个工作日内支付

需求确认款：需求规格说明书经客户确认签字后支付

开发完成款：系统开发完成并通过功能测试验收后支付

系统上线款：系统正式上线运行并通过用户验收后支付

质保期满款：系统上线运行3个月无质量问题后支付

（3）发票开具

发票类型：增值税专用发票

税率：6%（软件服务）

开票时间：收到款项后5个工作日内开具

发票内容：软件开发服务费、技术服务费

### （16）后续年度运维费用

#### 二十三、后续年度运维费用

| 年度 | 运维费用（万元/年） | 服务内容 |
| --- | --- | --- |
| 第2年 | 4.0 | 技术支持、系统维护、安全服务 |
| 第3年 | 3.5 | 技术支持、系统维护、安全服务 |
| 第4年及以后 | 3.0 | 基础技术支持、系统维护 |

注：后续年度运维费用按年支付，具体金额可根据实际服务内容协商确定。

## （九）资质证明

### （17）公司资质

#### 二十四、公司资质

（1）企业资质

营业执照：具有合法的企业法人资格

软件著作权：拥有相关软件著作权证书

ISO认证：通过ISO9001质量管理体系认证

高新技术企业：国家高新技术企业认定

（2）技术资质

开发团队：拥有专业的软件开发团队

技术能力：具备相关技术开发能力

项目经验：具有类似项目成功经验

技术认证：团队成员具有相关技术认证

（3）服务资质

服务能力：具备完善的技术服务能力

服务网络：拥有完善的服务网络

服务标准：建立完善的服务标准

客户满意度：客户满意度95%以上

### （18）项目经验

#### 二十五、项目经验

（1）类似项目经验

某大型企业组织架构系统：成功实施大型企业组织架构管理系统

某集团公司绩效考核系统：成功实施集团公司绩效考核系统

某政府部门人事管理系统：成功实施政府部门人事管理系统

（2）技术优势

技术先进性：采用最新的技术架构和开发框架

功能完整性：提供完整的功能解决方案

扩展性强：支持未来业务扩展和功能升级

维护性好：系统维护简单，升级方便

## （十）联系方式

### （19）投标单位信息

#### 二十六、投标单位信息

公司名称：北京智创科技有限公司

公司地址：北京市海淀区中关村软件园

邮政编码：100190

联系电话：010-12345678

传真号码：010-12345679

电子邮箱：[bid@zhichuang.com](mailto:bid@zhichuang.com)

公司网站：<www.zhichuang.com>

### （20）项目联系人

#### 二十七、项目联系人

项目经理：张经理

联系电话：138-0000-0000

电子邮箱：[zhang@zhichuang.com](mailto:zhang@zhichuang.com)

技术负责人：李工程师

联系电话：139-0000-0000

电子邮箱：[li@zhichuang.com](mailto:li@zhichuang.com)

商务负责人：王经理

联系电话：137-0000-0000

电子邮箱：[wang@zhichuang.com](mailto:wang@zhichuang.com)

## （十一）附件

### （21）附件清单

#### 二十八、附件清单

（1）附件1：系统功能截图

组织架构系统界面截图（10张）

绩效考核系统界面截图（15张）

系统集成平台界面截图（8张）

移动端界面截图（5张）

（2）附件2：技术文档

系统架构设计文档（50页）

数据库设计文档（30页）

API接口文档（40页）

部署指南文档（25页）

安全设计文档（20页）

（3）附件3：项目案例

某大型能源集团项目案例（20页）

某央企组织架构项目案例（15页）

某政府部门人事管理项目案例（12页）

客户推荐信（3份）

项目验收报告（3份）

（4）附件4：资质证书

企业营业执照（复印件）

软件著作权证书（20份）

ISO9001质量管理体系认证证书

ISO27001信息安全管理体系认证证书

高新技术企业证书

软件企业认定证书

（5）附件5：团队介绍

项目经理简历

技术负责人简历

核心开发团队介绍

技术认证证书

投标文件编制日期：2025年10月1日  
投标文件有效期：30天  
投标文件份数：正本1份，副本3份  
投标文件页数：共200页

## （十二）投标总结

### （22）我们的承诺

#### 二十九、我们的承诺

1、技术承诺：采用先进的技术架构，确保系统稳定可靠

2、质量承诺：严格按照质量标准开发，确保系统质量

3、服务承诺：提供7×24小时技术支持，确保系统正常运行

4、交付承诺：按合同约定时间完成项目实施和交付

### （23）我们的优势

#### 三十、我们的优势

1、成熟产品：基于已开发完成的双系统架构，功能完整

2、技术先进：采用Vue.js + Django现代化技术栈

3、经验丰富：具有多个类似项目成功经验

4、服务完善：提供完整的技术支持和运维服务

### （24）我们的价值

#### 三十一、我们的价值

1、提升效率：通过系统化考核管理，提升考核效率60%以上

2、降低成本：减少人工操作，降低管理成本

3、提高质量：通过标准化流程，提高考核质量

4、增强竞争力：通过科学考核体系，增强企业竞争力

本投标文件严格按照招标文件要求编制，所有承诺和报价均为最终承诺。我们期待与中煤内蒙古能源有限公司建立长期合作关系，共同推进绩效考核系统建设项目的成功实施。