

微型博客系统 - 数据库课程设计任务1

这是一个微型博客系统，具备创建、删除、修改及查询帖子的功能。

在线演示地址: <http://47.93.240.135:5000/blog/>

开发环境与技术说明

开发语言

开发语言选用当前最火热的Python语言。Python是一个解释型脚本语言，由于具有规范的格式，使得Python代码非常美观、简洁。在加上Python自带的各种Web框架，如 **Flask**、**Fjango**，开发人员能在短时间内迅速开发出强大的Web应用。总之，使用Python写代码的感觉非常爽。

Web服务器

本系统选用Python的 **Django** 库作为后端框架，**Django** 自带了一个简单的Web服务器。因为该系统仅作为练习用，故没必要采用 **Ngnix** 等专业的Web服务器。

数据库管理系统

数据库采用 **SQLite** 数据库。**SQLite** 是一个轻量级的关系型数据库，由于数据保存在单个文件中，这使得SQLite无需安装或配置即可食用。另外，其底层实现采用C语言，因此体积小且性能强大。

数据库操作接口采用Python内置的 **sqlite3** 模块，它提供类似 **JDBC** 的API用

于访问 **SQLite** 数据库。

数据库表的定义

Table Name: posts

column	type	info	description
id	INTEGER	primary key	Post's id
title	TEXT		Post's title
text	TEXT		Post's text content
timestamp	TEXT		Post's create date time

SQL commands to create table:

```
CREATE TABLE posts (  
    id INTEGER PRIMARY KEY,  
    title TEXT,  
    text TEXT,  
    timestamp TEXT  
);
```

数据库连接与增删改查

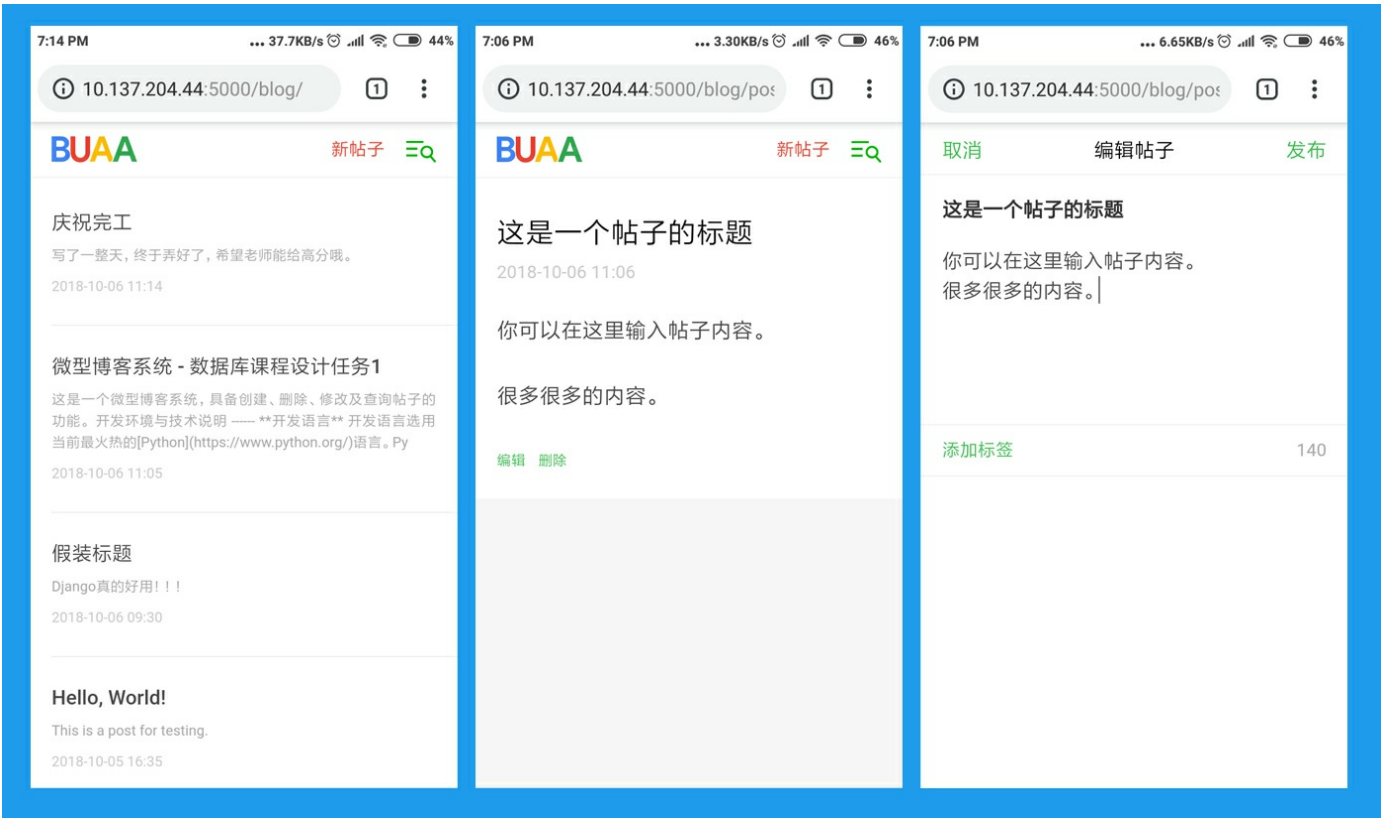
以下为Python语法示例：

```
import sqlite3
```

```
# connect
conn = sqlite.connect("my_database.sqlite3")
# get cursor
cursor = conn.cursor()
# create tables
sql_command = """
    CREATE TABLE posts (
        id INTEGER PRIMARY KEY,
        title TEXT,
        text TEXT,
        timestamp TEXT);
"""

cursor.execute(sql_command)
# Create
cursor.execute('INSERT INTO posts(title, text, timestamp)
values ("Title", "Text", "Time")')
# Read
result = cursor.execute('SELECT * FROM posts')
for post in result.fetchall():
    print(post)
# Update
cursor.execute('UPDATE posts SET title="A", text="B" where
id=1')
# Delete
cursor.execute("DELETE FROM posts where id=1")
```

功能截图



更多截图文件保存在 **screenshots** 目录下。