

به نام خدا

پروژه سیستم‌های نهفته

Embedded Web Server

اعضای تیم:

سارا خسروی ۹۷۱۰۱۵۸۶

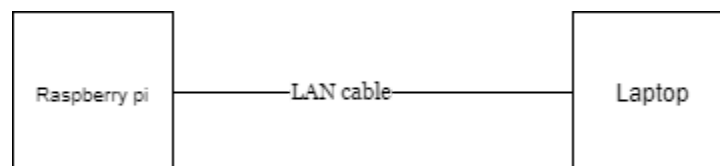
غزل شناور ۹۷۱۰۱۸۹۷

سپهر صفری ۹۷۱۰۸۲۶۳

شرح پروژه:

هدف این پروژه پیاده سازی یک embedded web-server بر روی یک پردازنده ARM با استفاده از Linux embedded است. این webserver باید قابلیت میزبانی از صفحات وب را به داشته باشد. برای تست، یک صفحه وب روی آن قرار داده شود که بتوان یک تصویر را در آن آپلود کرد، embedded server تصویر را بر روی یک حافظه خارجی ذخیره کند و در هر زمان دیگر که یک کاربر به آن صفحه متصل شد اجازه دانلود آن تصویر را بدهد.

بلوک دیاگرام سیستم:



سخت افزار مورد نیاز:

۱ - برد Raspberry pi

۲ - کابل LAN

۳ - مودم wifi (که اگر نبود با هات اسپات گوشی هم می توان برنامه را اجرا کرد.)

نرم افزار مورد نیاز:

۱- سیستم عامل raspberry pi

۲- فرانت: html, css, js

۳- بک: django framework

۴- پایگاه داده: MySql

کارهای انجام شده:

نصب سیستم عامل بر روی برد raspberry pi

یافتن ip و اتصال برد به اینترنت

نصب کتابخانه های مربوط به backend

پیاده سازی backend

پیاده سازی frontend

نحوه پیاده‌سازی:

ابتدا سیستم عامل بر روی برد raspberry pi را نصب کردیم. سپس روی آن کتابخانه‌های مورد نیاز مانند جنگو را ریختیم. دیتابیس مدنظر که mysql بود را نیز روی آن نصب کردیم. همچنین یک venv روی آن ساختیم. سپس برای آن که بتوان یک شبکه درست کرد، هر دوی لپتاپ و برد را به یک شبکه اینترنت متصل کردیم. سپس ip این شبکه را یافته و روی آن ip با استفاده از جنگو یک سایت بالا آوردیم. لازم به ذکر است که در صورتی که بخواهیم شبکه را با استفاده از ip جدیدی راه بیندازیم، کافی است آن ip را به لیست ipهای مجاز در فایل settings.py بیفزاییم.

در این سایت ابتدا به کاربر قابلیت ساخت اکانت داده شد. یعنی کاربر میتواند با انتخاب یک یوزرنیم و پسورد، صفحه شخصی خودش را داشته باشد. سپس کاربر به صفحه‌اش در این سایت رفته و میتواند با انتخاب یک فولدر در آن، محل ذخیره عکس را انتخاب کند. همچنین میتواند با ساخت فولدر جدید در اکانتش، یک عکس را در آنجا آپلود و یا دانلود نماید. لازم به ذکر است که میتوان عملیات آپلود یا دانلود را روی هر یک از لپتاپ یا برد رزپبری انجام داد و سپس روی دیگری تغییرات را مشاهده کرد و عکس جدید را دریافت نمود.

برای بالا آوردن سایت با جنگو، میدانیم که جنگو برای هر موجودیتی یک app درست میکند. برای مثال appهای جنگو برنامه ما یوزرها هستند. همچنین هر اپ درون خود init , admin, urls, tests, views, models, forms , ... دارد که اینها از ساختار فریمورک جنگو تبعیت میکنند.

سرور نیز یک تابع دارد که هر ریکوئستی به آن زدیم متوجه شده و به آن پاسخ دهد و سایت را بالا بیاورد. این سایت با دستور

```
python manage.py runserver 0.0.0.0:8000
```

بالا میاید. و حال روی آدرس 127.0.0.1:8000 میتوان سایت را مشاهده کرد و عملیات مورد نظر را انجام دادو لازم به ذکر است که این آدرس بسته به آدرس ip ها هر بار فرق میکند که با دستور ifconfig در لینوکس (رزپبری) یا دستور ipconfig در ویندوز میتوان آدرس را یافت و سایت را روی همان بالا آورد.

در نهایت میتوان به دیتاست رفت و با کوئریهای زبان mysql، اطلاعات ذخیره شده در دیتابیس را مشاهده کرد.

برای ساخت آن از دستورات زیر استفاده میکنیم:

نخست یوزر را میسازیم:

```
sudo mysql
```

```
'CREATE USER 'rasp_user'@'localhost' IDENTIFIED BY 'rasp_user
```

```
'GRANT ALL PRIVILEGES ON * . * TO 'rasp_user'@'localhost
```

```
FLUSH PRIVILEGES
```

```
Exit
```

سپس دیتابیس را ایجاد میکنیم:

```
mysql -u rasp_user -p
```

```
CREATE DATABASE `rasp_db` CHARACTER SET utf8
```

```
exit
```

برای آپدیت کردن آن نیز میتوان از دستور

```
python manage.py makemigrations
```

و سپس

```
python manage.py migrate
```

استفاده کرد.

برای این کار ابتدا با دستور `mysql -u rasp_user -p` وارد دیتابیس میشویم. لازم به ذکر است که نام کاربری و پسور دیتابیس هر دو `rasp_user` هستند. سپس با دستور `USE rasp_db` میتوانیم دیتابیس مدنظرمان را انتخاب کنیم و با `SHOW TABLES` جدولهای موجود در آن را ببینیم. سپس باید کوئریها را بسته به چیزی که میخواهیم چک کنیم بنویسیم. برای مثال میتوانیم برای مشاهده ی یوزرهای ساخته شده در دیتابیس،

```
SELECT * FROM auth_user
```

را میزنیم.

لازم به ذکر است برای بهبود ظاهر برنامه، فایلهای HTML CSS برنامه را بهبود دادیم.

برای مشاهده ی خروجی ها نیز ویدئوی کامل دموی این برنامه به آقای سیادت زاده فرستاده شده است.

موارد پیاده شده به صورت امتیازی:

استفاده از جنگو

ساخت اکانت

امکان انتخاب دایرکتوری

اتصال به سرور از خارج

بهبود دادن فرانت اند نسبت به جنگوی اولیه

