



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پروژه کارشناسی

گرایش معماری سیستم‌های کامپیوتری

طراحی و پیاده‌سازی سامانه اطلاعات کالا در فروشگاه‌ها
و انبارها

نگارش

امین رشیدبیگی

استاد راهنما

دکتر بهادر بخشی سراسکانرود

مهرماه ۱۳۹۸

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تأیید و تصویب پایان نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- را قرار دهید.

نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.

به نام خدا

تاریخ: مهرماه ۱۳۹۸

تعهدنامه اصالت اثر



اینجانب امین رشیدبیگی متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

امین رشیدبیگی

امضا

تقدیم بہ
خالق بی نہایت،
پرستگارِ خلاق،
و غواصانِ گشتی ناپذیرِ دریایِ پانچ ماہ۔

سپاسگزاری

از پدرم که چون کوهی استوار و مادرم که چون دریای محبت در فراز و نشیب زندگی دلسوزانه
همراهم بوده‌اند؛
از اساتید بزرگوار جناب آقای دکتر بهادر بخشی که در کمال سعه صدر، با حسن خلق و فروتنی،
رهنمون من شده‌اند؛
و از سایر عزیزانی که در کنارشان این نتیجه حاصل آمد کمال تشکر و قدردانی را دارم.

امین رشیدی
مهرماه ۱۳۹۸

چکیده

مدیریت و یافتن کالاها در فروشگاه‌ها و انبارهای بزرگ یکی از دغدغه‌های فروشندگان و مدیران انبارهاست. هدف این پروژه پیاده‌سازی سیستم سامانه‌ی مدیریت و دریافت اطلاعات کالاها در این محیط‌هاست. این سامانه متشکل از قطعه‌های سخت‌افزاری و همچنین نرم‌افزار لازم برای تعامل با کاربر است. این سامانه از طریق بوردها و صفحه‌های نمایش لمسی قرار گرفته بر روی آنها در قسمت‌ها و دسته‌های مختلف انبار و فروشگاه، این اجازه را به کارکنان انبار و فروشگاه می‌دهد که موجودی کالاهای خود را به روزرسانی کنند و از طریق نرم‌افزار مربوطه، مکان، موجودی و سایر مشخصات کالا را جست‌وجو کنند. بوردها این اطلاعات را از طریق یک پایگاه‌داده دریافت می‌کنند و پس از هرگونه عملیات جمع‌آوری، این اطلاعات در پایگاه‌داده بروزرسانی می‌شود. خروجی این سامانه، نرم‌افزاری تحت وب با رابط کاربری مناسب می‌باشد که محل قرارگیری کالاهای موجود در انبار و فروشگاه را مشخص می‌کند و قابلیت جست‌وجو در بین این کالاها را به کاربر می‌دهد.

واژه‌های کلیدی:

انبارداری، هوشمند، فروشگاه، کالا

فهرست مطالب

صفحه

عنوان

۴	تعریف مسئله، راه حل پیشنهادی و تعریف نیازمندی‌ها
۵	۱-۲ مقدمه
۵	۲-۲ تعریف مسئله
۵	۳-۲ راه حل پیشنهادی
۶	۴-۲ نیازمندی‌های پروژه
۶	۱-۴-۲ رابط کاربری مدیریت
۶	۲-۴-۲ برنامه جمع‌آوری محصولات سفارش‌ها
۷	۳-۴-۲ برنامه انتخاب محصولات بخش‌ها
۷	۵-۲ نتیجه‌گیری
۸	۳ اجزا و تکنولوژی‌ها
۹	۱-۳ مقدمه
۹	۲-۳ رزبری پای مدل 3A
۱۰	۳-۳ نمایشگر لمسی 3.5 اینچی وِیوشر
۱۲	۴-۳ سیستم‌عامل رزبین
۱۳	۵-۳ چهارچوب نرم‌افزاری جنگو
۱۳	۱-۵-۳ مزیت‌های جنگو
۱۵	۲-۵-۳ تاثیر پایتون بر روی جنگو
۱۶	۳-۵-۳ بخش‌های جنگو
۱۷	۴-۵-۳ انطباق با میزبان‌های مختلف
۱۷	۶-۳ چهارچوب نرم‌افزاری REST جنگو
۱۸	۱-۶-۳ رابط برنامه‌نویسی نرم‌افزار چیست؟
۱۹	۲-۶-۳ REST چیست؟
۲۰	۳-۶-۳ جنگو REST
۲۱	۷-۳ چهارچوب نرم‌افزاری Vue.js
۲۱	۱-۷-۳ اجزاء

۲۱	۲-۷-۳ قالب‌ها
۲۲	۳-۷-۳ انتقال‌ها
۲۲	۴-۷-۳ چهارچوب نرم‌افزاری Nuxt.js
۲۲	۸-۳ پایگاه‌داده رابطه‌ای Postgress
۲۳	۱-۸-۳ پایگاه داده رابطه‌ای
۲۵	۲-۸-۳ پایگاه‌داده Postgress
۲۶	۹-۳ ماشین مجازی
۲۷	۱-۹-۳ دلایل استفاده از میزبان مجازی شخصی
۲۸	۲-۹-۳ موارد استفاده از میزبان مجازی شخصی
۲۸	۱۰-۳ نتیجه‌گیری
۳۰	۴ طراحی و پیاده‌سازی
۳۱	۱-۴ مقدمه
۳۱	۲-۴ نمایشگر لمسی
۳۲	۱-۲-۴ نحوه قرارگیری بر روی پین‌های GPIO
۳۲	۲-۲-۴ راه‌اندازی نمایشگر بر روی رزبری پای
۳۴	۳-۲-۴ تنظیم صفحه لمسی
۳۵	۳-۴ طراحی و پیاده‌سازی پایگاه‌داده
۳۵	۱-۳-۴ محصولات
۳۶	۲-۳-۴ سفارش‌ها
۳۷	۳-۳-۴ بخش‌ها
۳۷	۴-۳-۴ دسته‌بندی‌ها
۳۷	۵-۳-۴ جداول میانی
۳۸	۴-۴ رابط‌های برنامه‌نویسی نرم‌افزار
۳۸	۱-۴-۴ دریافت محصولات فعال
۳۹	۲-۴-۴ لیست سفارش‌ها
۴۰	۳-۴-۴ دریافت محصولات یک سفارش
۴۱	۴-۴-۴ لیست بخش‌ها

۴۲	۵-۴-۴ ثبت جایگاه یک محصول
۴۲	۶-۴-۴ اطلاع اتمام جمع‌آوری محصولات سفارش
۴۲	۵-۴ برنامه انتخاب محصولات بخش‌ها
۴۴	۶-۴ برنامه جمع‌آوری محصول
۴۵	۷-۴ رابط کاربری مدیریت
۴۶	۱-۷-۴ احراز هویت و دسترسی
۴۷	۲-۷-۴ ایجاد و ویرایش عناصر پایگاه‌داده
۴۷	۸-۴ نتیجه‌گیری
۴۹	۵ جمع‌بندی، نتیجه‌گیری و پیشنهادات
۵۰	۱-۵ جمع‌بندی و نتیجه‌گیری
۵۰	۲-۵ کارهای آینده
۵۲	منابع و مراجع

شکل	فهرست اشکال	صفحه
۱-۳	اجزای رزبری پای مدل A۳	۱۱
۲-۳	نمایشگر لمسی ۵.۳ اینچی	۱۱
۳-۳	نحوه ارتباط لایه‌های جنگو	۱۴
۱-۴	معماری کلی پروژه	۳۲
۲-۴	قرارگیری نمایشگر لمسی بر روی رزبری پای	۳۳
۳-۴	تنظیم صفحه نمایش لمسی	۳۵
۴-۴	طراحی پایگاه داده	۳۶
۵-۴	نمایش بخش‌ها	۴۳
۶-۴	نمایش محصولات جهت ثبت در بخش مورد نظر	۴۳
۷-۴	نمایش لیست محصولات برنامه انتخاب محصولات بخش‌ها بر روی رزبری پای	۴۴
۸-۴	نمایش گزینه تایید برنامه انتخاب محصولات بخش‌ها بر روی رزبری پای	۴۴
۹-۴	لیست سفارش‌ها	۴۵
۱۰-۴	لیست محصولات سفارش	۴۶
۱۱-۴	صفحه احراز هویت	۴۷
۱۲-۴	ایجاد محصول	۴۸
۱۳-۴	ویرایش محصول	۴۸

فصل اول

مقدمه

۱-۱ مقدمه

انبارها ساختمان یا محوطه‌ای هستند که برای نگهداری کالاها استفاده می‌گردند. صنعت‌گران، واردکنندگان، صادرکنندگان، عمده‌فروشان و گمرک، استفاده‌کنندگان انبارها می‌باشند. انبارها معمولاً در شهرها، شهرک‌های صنعتی و کارخانجات ساخته می‌شوند ولی ممکن است جهت سهولت در دریافت و صدور کالا، در کنار راه‌های اصلی، فرودگاه و یا بنادر نیز ساخته شوند تا کالاها مستقیماً به آنجا وارد و یا خارج گردد.

انبارها از لحاظ کاربری ممکن است انواع گوناگونی داشته باشند. انبار محصول، انبار مواد اولیه، انبار قطعات نیم‌ساخته، انبار قطعات یدکی، انبار ابزار آلات، انبار غلات و مخازن تنها تعداد اندکی از انواع انبارها می‌باشند. با توجه به کاربری و تنوع زیاد انواع انبارها، ماشین آلات، سیستم‌ها و تجهیزات سخت‌افزاری و نرم‌افزاری بسیاری متناسب با نوع کاربری انبارها توسعه یافته‌اند. از آنجا که انبارها در فرآیند تجارت هیچ‌گونه ارزش افزوده‌ای ایجاد نمی‌نمایند، مکان‌یابی، جابجایی و خروج کالاها از انبار بسیار خطیر می‌باشد و لازم است تا عملیات انبارها با حداقل هزینه و حداکثر بهره‌وری انجام پذیرد.

سیستم اداره انبارها با توجه به نوع کاربری و جامعه مورد استفاده بسیار متفاوت است. در برخی از جوامع، انبارها هنوز به صورت سنتی اداره می‌گردند، حال آنکه برخی از انبارها کاملاً خودکار و مکانیزه می‌باشند؛ بدون اینکه نیاز به نیروی کار انسانی داشته باشند و از طریق سیستم‌های دریافت و انتقال خودکار کالاها و نرم افزارهای لجستیکی^۱ مدیریت می‌شوند.

انبارداری به صورت سنتی، در صورت افزایش مقیاس انبار، مشکلات زیادی را به وجود می‌آورد. در این روش هرگونه مشکل پیش آمده برای کارکنان انبار باعث ضرر و زیان مالی می‌شود. از این مشکلات می‌توان به خستگی، چندکاری، محیط پر سروصدا و کاهش تمرکز کارکنان اشاره کرد. مشکل پیدا کردن و جمع‌آوری کالاها مورد نیاز در فروشگاه‌ها و انبارهای بزرگ، همواره دغدغه کارخانه‌ها و شرکت‌های توزیع‌کننده و صاحبان فروشگاه‌ها بوده‌است. این مشکل زمانی حیاتی می‌شود که جایگاه کالاها ثابت نباشد و در طول زمان تغییر کند. در این صورت جمع‌آوری کالا به گلوگاه مجموعه تبدیل می‌شود. در در جامعه کنونی و در شرایط فعلی، این روندها بیشتر به صورت انسانی و بدون کمک رایانه انجام می‌شوند. از موارد مشکل‌ساز روند فعلی، می‌توان به افزایش هزینه‌های چیدمان و نگهداری کالاها اشاره کرد. چون چیدمان نامناسب کالاها و

^۱Logistic

جمع‌آوری آن‌ها در انبارها، به نیروی انسانی بیشتری نیاز دارد. این پروژه، امکانی را برای کارکنان فراهم می‌کند که جایگاه اجناس را به وسیله قطعه‌های سخت افزاری بروزرسانی کنند و همچنین نرم‌افزاری در اختیار آنان قرار می‌دهد تا به وسیله آن، بتوانند بسیار راحت‌تر و سریع‌تر موارد مورد نیاز خود را پیدا کنند. همچنین این پروژه به مشتریان فروشگاه‌های بزرگ و زنجیره‌ای کمک می‌کند که اطلاعات و جایگاه کالاهای موجود در لیست خریدشان را دریافت کنند. علاوه بر این، رابطی برای مدیریت مجموعه فراهم شده‌است تا بتوانند وضعیت فعلی مجموعه را زیر نظر داشته باشند و در صورت نیاز، تغییراتی در سیستم اعمال کنند.

در این نوشته، ابتدا به تعریف مسئله و راه حل پیشنهادی و همچنین نیازمندی‌های مسئله پرداخته می‌شود. سپس تکنولوژی‌های نرم‌افزاری و همچنین قطعات سخت‌افزاری انتخاب شده معرفی می‌گردند. در قسمت بعد به شرح مفصل پیاده‌سازی راه‌حل پرداخته می‌شود و در نهایت به جمع‌بندی کارهای انجام شده و همچنین کارهایی که در آینده برای این پروژه می‌توان انجام داد پرداخته می‌شود.

فصل دوم

تعریف مسئله، راه حل پیشنهادی و تعریف نیازمندی‌ها

۱-۲ مقدمه

در این قسمت ابتدا به تعریف و طرح مسئله پرداخته می‌شود. سپس راه‌حلی پیشنهاد داده می‌شود و در ادامه نیازمندی‌های پروژه بر اساس راه حل پیشنهادی تعریف می‌شود. پیش از آغاز پیاده‌سازی نرم‌افزار لازم است درک جامع و کاملی از مسئله پیدا کنیم و فهرست دقیقی از نیازمندی‌های آن تهیه نماییم. در این فصل سعی شده است پس از بیان مسئله، نیازمندی‌ها و روش حل آن مشخص شوند.

۲-۲ تعریف مسئله

مدیریت و یافتن کالاها در فروشگاه‌ها و انبارهای بزرگ، یکی از دغدغه‌های اصلی فروشندگان و انباردارهاست. هنگامی که سفارشی ثبت می‌شود، جمع‌آورندگان محصولات در انبارها وظیفه دارند در انبوهی از کالاها و بخش‌های انبار، کالاهای ثبت‌شده در سفارش را جمع‌آوری کنند. اگر انبار مربوطه تعداد زیادی کالا داشته باشد و یا انباری مربوط به کسب و کارهایی با محصولات پرمصرف باشد، این موضوع به گلوگاه تبدیل می‌شود. همچنین مشتریان فروشگاه‌های زنجیره‌ای همواره با پیدا کردن لیست محصولات مورد نیاز خود در بین انبوهی از کالاها مشکل دارند. کاهش زمان صرف‌شده برای جمع‌آوری کالاها، برای مدیران انبارها، از اهمیت ویژه‌ای برخوردار است.

۳-۲ راه حل پیشنهادی

راه حل پیشنهادی برای حل این مسئله، طراحی و پیاده‌سازی سامانه‌ای جهت مدیریت، ایجاد تغییرات و کنترل کالاها در جایگاه‌های متفاوت در این محیط‌هاست. این سامانه متشکل از قطعه‌های سخت‌افزاری و همچنین نرم‌افزار لازم برای تعامل با کاربر است. این سامانه از طریق بوردها و صفحه‌های نمایش لمسی قرار گرفته بر روی آن، در قسمت‌ها و دسته‌های مختلف انبار و فروشگاه، این اجازه را به کارکنان انبارها و فروشگاه‌ها می‌دهد که موجودی کالاهای خود را به روزرسانی کنند و از طریق نرم‌افزار مربوطه، مکان، موجودی و سایر مشخصات کالا را جست‌وجو کنند. همچنین به یک برنامه سمت کاربر جهت مدیریت داده‌ها و کاربران نیاز است.

۴-۲ نیازمندی‌های پروژه

با توجه به موارد مطرح شده در قسمت تعریف مسئله و همچنین در قسمت راه حل پیشنهادی، نیازمندی‌هایی به وجود آمد که در ادامه به آن‌ها پرداخته می‌شود.

۱-۴-۲ رابط کاربری مدیریت

پروژه شامل کاربرها و محصولات، بخش‌ها^۱، سفارش‌ها و دسته‌بندی‌های مختلف است. برای مدیریت و کنترل این قسمت‌ها، به یک محیطی نیاز است که مدیر، دسترسی تغییر بخش‌های مختلف را بر اساس نیاز داشته باشد.

مدیر در محیط مدیریت^۲ باید بتواند اعمال زیر را انجام دهد:

- اضافه کردن، ویرایش و یا حذف یک محصول
- اضافه کردن، ویرایش و یا حذف یک دسته‌بندی
- اضافه کردن، ویرایش و یا حذف یک بخش
- مشاهده و ویرایش سفارش‌های ثبت شده

با توجه به این که ممکن است بر اساس نیاز، لازم شود افراد دیگری غیر از مدیر اصلی توانایی تغییر بخش‌های ذکر شده را داشته باشند، باید امکان دادن دسترسی به کاربران دیگر نیز داده شود. همچنین نیاز است سطح دسترسی کاربران تحت عنوان نقش‌های مختلف نیز تعریف شوند. از وظایف مدیر اصلی رابط کاربری مدیریت، تعیین دسترسی‌های کاربران و نقش‌های مختلف برنامه است.

۲-۴-۲ برنامه جمع‌آوری محصولات سفارش‌ها

جمع‌آوردندگان^۳ داخل انبار و فروشگاه نیاز دارند تا از سفارش‌های ثبت شده مطلع شوند تا به جمع‌آوری محصولات آن سفارش بپردازند. همچنین پس از جمع‌آوری محصولات، به مدیر و بقیه جمع‌آوردندگان باید اطلاع دهند که سفارش جمع‌آوری گردید.

¹Segments

²Admin Panel

³Picker

این برنامه شامل موارد زیر است:

- مشاهده سفارش‌های ثبت شده و وضعیت آن‌ها
- مشاهده محصولات سفارش
- اطلاع دادن پس از جمع‌آوری محصولات سفارش

۳-۴-۲ برنامه انتخاب محصولات بخش‌ها

بخش‌های مختلف در انبارها و فروشگاه‌ها دارای محصولات متفاوتی هستند. این محصولات در طول زمان اضافه، کم و یا جابجا می‌شوند.

در این پروژه به قطعات سخت‌افزاری جهت قرارگیری بر روی بخش‌های مختلف نیاز است تا کاربر بتواند به راحتی با آن تعامل برقرار کند و محصولات فعلی بخش و موجودی آن‌ها را ثبت کنند. برای ثبت محصولات، علاوه بر قطعات سخت‌افزاری، به برنامه‌ای نیاز است تا محصولات موجود را نمایش دهد و پس از تغییر جایگاه محصولات، آن تغییر را در پایگاه داده ثبت کند.

۵-۲ نتیجه‌گیری

در این قسمت مسئله مورد نظر تعریف و شرح داده شد. همچنین راه‌حلی پیشنهادی برای مسئله ارائه شد. در نهایت و با توجه به راه حل، نیازمندی‌های مسئله تعریف شد. این نیازمندی‌ها شامل برنامه‌های سمت کاربر و همچنین مدیریت مجموعه است.

فصل سوم

اجزا و تکنولوژی‌ها

۱-۳ مقدمه

در این قسمت اجزا، قطعات و تکنولوژی‌های سخت‌افزاری و نرم‌افزاری مورد استفاده در پروژه معرفی خواهند شد.

۲-۳ رزبری پای مدل 3A

امروزه با پیشرفت روزافزون تکنولوژی و ظهور پردازنده‌های کم‌مصرف آرم^۱، حجم و وزن کامپیوترها هرروز کوچک‌تر و در عوض سرعت و قدرت آن‌ها افزایش می‌یابد. بردهای رزبری پای [۵] از جمله کوچک‌ترین و پرقدرت‌ترین بردهای کامپیوتری هستند که علاوه بر ابعاد کوچک و وزن کم، از سرعت و دقت بالایی برخوردارند. بردهای رزبری پای توسط بنیاد خیریه رزبری^۲ و با هدف گسترش و توسعه علوم رایانه برای دانش‌آموزان ساخته شده است، تا نسل آینده بهتر بتواند روش عملکرد رایانه‌ها را درک کند. این دستگاه کوچک با توانایی‌های بسیار بالا برای تمام سنین و علاقه‌مندان به مباحث برنامه‌نویسی و اینترنت‌اشیاء^۴ کاربرد دارد. برد رزبری پای ۳ دارای یک درگاه^۵ استاندارد رابط چندرسانه‌ای وضوح بالا^۶ است که این برد را قادر می‌سازد به انواع تلویزیون‌ها و نمایشگرها متصل شود. همچنین این برد برخلاف نسل‌های قبلی خود، دارای قابلیت اتصال به اینترنت بی‌سیم^۷ و بلوتوث^۸ داخلی است که در کنار یک درگاه شبکه اترنت^۹ در قسمت شبکه و اتصال به اینترنت و سایر شبکه‌ها چیزی کم ندارد. رزبری پای ۳ در قسمت پردازنده مرکزی از چیپ چهار هسته‌ای Cortex-A53 BCM2837 شرکت برادکام^{۱۰} استفاده می‌کند که بر اساس گفته بنیاد رزبری، تا ۶۰ درصد سریع‌تر از نسل پیشین، یعنی رزبری پای ۲ است. انطباق‌دهنده^{۱۱} مورد نیاز برای این برد باید دارای ولتاژ ۵ ولت و جریان ۲

¹ARM²Raspberry PI³Raspberry Foundation⁴Internet of Things⁵Port⁶High-Definition Multimedia Interface (HDMI)⁷Wifi⁸Bluetooth⁹Ethernet¹⁰Broadcom Corporation¹¹Adaptor

آمپر باشد که می‌توان از درگاه گذرگاه سریال سراسری^{۱۲} برد رزبری پای نیز استفاده کرد ولی توصیه می‌شود تا با انطباق‌دهنده مخصوص راه‌اندازی گردد. این پردازنده قادر است در هر هسته با فرکانس ۲.۱ گیگاهرتز فعالیت کند. همچنین این برد در قسمت گرافیک و پردازنده گرافیکی رزبری پای پیشرفت چشمگیری داشته و با یک پردازنده‌ی گرافیکی ۴۰۰ مگاهرتز از نوع Dual Core VideoCore IV GPU همراه شده است. این پردازنده گرافیکی دوهسته‌ای اگرچه نمی‌تواند ویدئوها و بازی‌هایی با کیفیت 4K را پشتیبانی کند، ولی با این‌حال از پس اکثر کاربردهای روزمره بر می‌آید. حافظه داخلی رزبری پای ۳، یک گیگابایت است که برای سیستم‌عامل‌های کم‌حجم مانند رزبرین^{۱۳} و برخی نسخه‌های لینوکس و اندروید مناسب است. و اما تفاوت اصلی کامپیوترهای تک‌بردی مانند رزبری پای با سایر کامپیوترها، وجود تعداد نسبتاً زیادی درگاه ورودی و خروجی عام منظوره^{۱۴} است. این درگاه‌ها که تعداد آن در رزبری پای ۴۰ عدد است، کاربردهای بسیار زیادی در اینترنت اشیا و اتوماسیون صنعتی دارند. این درگاه توانایی اتصال رزبری پای را به سایر دستگاه‌ها و سنسورهای الکترونیکی و تعامل با آن‌ها را فراهم می‌کنند. اجزای رزبری پای استفاده شده در این پروژه در تصویر ۳-۱ نمایش داده شده است.

۳-۳ نمایشگر لمسی 3.5 اینچی وِیوشر

این نمایشگر لمسی ساخت شرکت وِیوشر^{۱۵} بوده و دارای رزولوشن 320x480 می‌باشد. این نمایشگر را می‌توان برای انواع بوردهای رزبری پای مورد استفاده قرار داد. درایورهای این نمایشگر به صورت آماده بوده و به صورت مستقیم با سیستم‌عامل رزبین هماهنگ می‌باشد. اندازه‌ی این نمایشگر کاملاً با انواع بوردهای رزبری پای مطابقت دارد و به راحتی بر روی آن قرار می‌گیرد. بر روی این نمایشگر می‌توان فیلم‌ها و عکس‌هایی با قالب^{۱۶} های مختلف و با کیفیتی بالا نمایش داد. این نمایشگر از نرم‌افزار صفحه‌کلید نیز پشتیبانی می‌کند که در این حالت نیازی به صفحه‌کلید خارجی نمی‌باشد. صفحه لمسی این نمایشگر از نوع مقاوم‌تی می‌باشد. این نمایشگر در تصویر ۳-۲ نشان داده شده است.

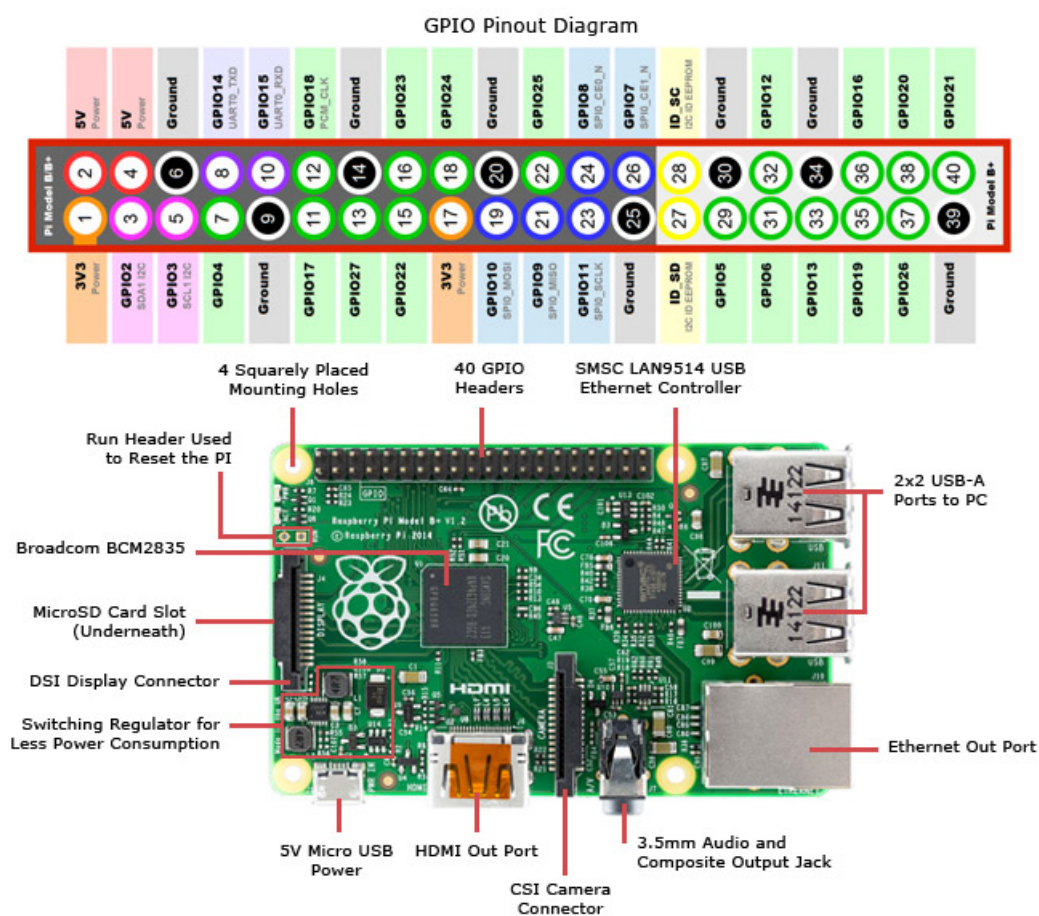
¹²Universal Serial Bus (USB)

¹³Raspbian

¹⁴General Purpose Input/Output (GPIO)

¹⁵Waveshare

¹⁶Format



شکل ۳-۱: اجزای رزبری پای مدل A۳



شکل ۳-۲: نمایشگر لمسی ۵.۳ اینچی

برخی دیگر از ویژگی‌های این نمایشگر:

- یک جایگزین مناسب برای مانیتورهایی با رابط چندرسانه‌ای وضوح بالا
- پشتیبانی به‌وسیله درایورهای مورد نیاز
- پشتیبانی از سیستم‌عامل رزبین
- پشتیبانی از تمام نسخه‌های رزبری پای
- عکس‌برداری لمسی (در ۱۷ حالت دوربین)
- کیفیت بالای ساخت با روکش طلایی
- نمایشگر از جنس ترانزیستور فیلم نازک^{۱۷}
- رابط محیطی سریال^{۱۸}
- درایور لمسی XPT2046
- پشتیبانی از ۶۵۵۳۶ رنگ مختلف
- نور پس‌زمینه از نوع دیودهای ساطع‌کننده نور^{۱۹}

۴-۳ سیستم‌عامل رزبین

سیستم‌عامل نصب شده بر روی بورد رزبری پای، رزبین نام دارد. رزبین یک سیستم‌عامل رایانه‌ای مبتنی بر دبین^{۲۰} برای رزبری پای است. نسخه‌های مختلفی از رزبین شامل Raspbian Buster و Raspbian Stretch وجود دارد. این سیستم‌عامل را بنیاد رزبری از سال ۲۰۱۵ به طور رسمی به‌عنوان سیستم‌عامل اصلی برای خانواده رایانه‌های رزبری پای ارائه داده است. رزبین توسط مایک تامپسون^{۲۱} و پیت گرین^{۲۲} به عنوان یک پروژه مستقل ایجاد شد. نسخه اولیه این سیستم‌عامل

¹⁷TFT LCD

¹⁸Serial Peripheral Interface (SPI)

¹⁹Light-Emitting Diode (LED)

²⁰Debian

²¹Mike Thompson

²²Peter Green

در ژوئن ۲۰۱۲ به پایان رسید. این سیستم عامل هنوز در حال توسعه است. رزبین برای پردازنده های آرم با توان پردازشی کم، بهینه شده است. رزبین در آخرین نسخه ارائه شده، از پیکسل^{۲۳} به عنوان محیط رومیزی^{۲۴} خود استفاده می کند.

۳-۵ چهارچوب نرم افزاری جنگو

یک چهارچوب نرم افزاری^{۲۵} سطح بالا، بسیاری از موارد برنامه نویسی را بصورت خودکار فراهم کرده و در اختیار برنامه نویس قرار می دهد. همچنین روش هایی میانبر و واسط برای اجرای اعمال مختلف را دارا می باشد. این ویژگی باعث می شود برنامه نویس برای پیاده سازی بخش های مختلف کمتر کد بزند. جنگو^{[۱]۲۶} یک چارچوب نرم افزاری آزاد و متن باز^{۲۷} مبتنی بر پایتون^{۲۸} است که امکان طراحی و ایجاد بسیار سریع و آسان برنامه های تحت وب را فراهم می کند. جنگو از الگوی معماری MTV^{۲۹} پیروی می کند. این مؤسسه توسط بنیاد نرم افزاری مستقل جنگو^{۳۰} تأسیس شده و نگهداری می شود. هدف اصلی جنگو، سهولت در ایجاد وبسایت های پیچیده و با محوریت پایگاه داده است. این چارچوب بر ویژگی هایی از جمله استفاده مجدد^{۳۱} و تعداد خط کد کم^{۳۲}، وابستگی کم^{۳۳}، توسعه سریع^{۳۴} و اصل تکرار خود^{۳۵} تأکید دارد. در این پروژه برای پیاده سازی ساختار کلی Back-end و همچنین رابط کاربری مدیریت از جنگو استفاده شده است.

۳-۵-۱ مزیت های جنگو

در ادامه ویژگی های این چهارچوب قدرتمند توضیح داده خواهد شد.

²³Pi Improved X-Window Environment, Lightweight(PIXEL)

²⁴Desktop Enviornment

²⁵Software Framework

²⁶Django

²⁷Open Source

²⁸Python

²⁹Model Template View

³⁰Django Software Independent Foundation

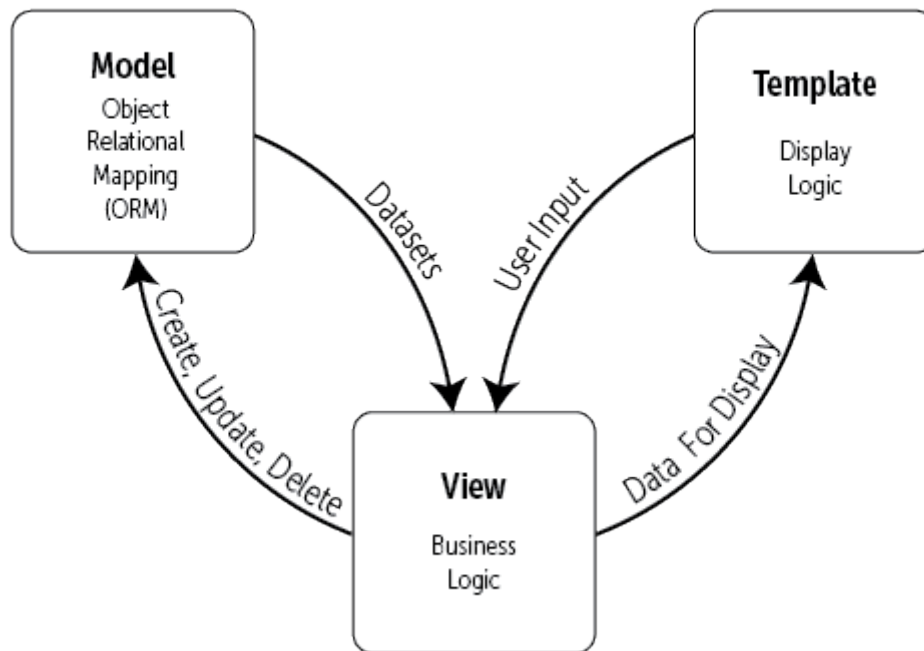
³¹Reusability

³²Less Code

³³Low Coupling

³⁴Rapid Development

³⁵Don't repeat yourself (DRY)



شکل ۳-۳: نحوه ارتباط لایه‌های جنگو

امکان جداسازی محتوا از ظاهر نمایشی

در اکثر زبان‌های برنامه‌نویسی، کدهای زبان نشانه‌گذاری ابرمتنی با کدها و محتوای سایت آمیخته می‌گردند، که باعث ایجاد مشکلات هنگام تغییرات آتی و نگهداری می‌گردد. با استفاده از این روش ظاهر نمایشی سایت، بصورت جداگانه در فایل‌های خاص ذخیره می‌گردد. اکنون با اعمال تغییر در هر نوع محتوا، نیازی به ویرایش دیگری نیست و این دو موجودیت مستقل می‌باشند.

لایه‌بندی

در حالت معمول هر برنامه نوشته شده با جنگو دارای ۳ لایه مهم می‌باشد. بخش نمایشی^{۳۶}، محتوا و یا کدهای کنترلی^{۳۷} و بخش ذخیره دائمی اطلاعات^{۳۸}. البته با توجه به نوع برنامه و نیازهای برنامه‌نویس می‌توان این لایه‌ها را ادغام کرده یا نادیده گرفت. نحوه ارتباط این لایه‌ها در تصویر ۳-۳ نشان داده شده است.

³⁶Template

³⁷View

³⁸Model

امکان استفاده موثر از سطح بالایی از تجرید و انتزاع

جنگو در بسیاری از موارد با استفاده از مفهوم انتزاع، سهولت زیادی را فراهم کرده است. برای مثال برای کار با تکنولوژی‌هایی چون پروتکل انتقال پرونده^{۳۹} یا زبان نشانه‌گذاری ابرمتنی^{۴۰}، با یک مفهوم انتزاعی و سطح بالا روبرو خواهیم بود که با استفاده از روابط و توابع متعدد، برنامه‌نویسی را بسیار آسان و قدرتمند می‌کند.

علاوه بر موارد ذکر شده، جنگو ویژگی‌های دیگری نیز دارد. از جمله:

- امکان تعیین اینکه کدام کد یا توابع مسئول جواب دادن به آدرس درخواست داده شده هستند.
- تسهیل نمایش، اعتبارسنجی و نمایش فرم‌های ساخته شده از زبان نشانه‌گذاری ابرمتنی یکی از مهمترین روش‌ها برای دریافت اطلاعات از یک کاربر وب می‌باشد.
- حذف پسوند فایل از آدرس‌های وب (مثلا .aspx یا .php)

۳-۵-۲ تاثیر پایتون بر روی جنگو

همانطور که اشاره شد، جنگو بر اساس زبان برنامه‌نویسی پایتون نوشته شده که باعث شده است تا ویژگی‌های زبان پایتون برای جنگو نیز به ارث رسیده باشد. در زیر به برخی از این ویژگی‌ها اشاره شده است:

- پایتون زبانی تفسیری^{۴۱} بوده و برای اجرا نیازی به کامپایل ندارد. این ویژگی باعث می‌شود که بعد از تغییر کد، نتایج کار بلافاصله قابل مشاهده باشد.
- داده‌ها و متغیرها در پایتون پویا و بدون نیاز به تعریف نوع^{۴۲} هستند.
- نحو زبان پایتون کوتاه و در عین حال واضح و قابل فهم است. این ویژگی باعث می‌شود که برای انجام کارهای مشابه در زبان‌های برنامه‌نویسی دیگر، به نسبت کدنویسی کمتری انجام شود.

³⁹File Transfer Protocol;FTP

⁴⁰HyperText Markup Language

⁴¹interpretive

⁴²Type

۳-۵-۳ بخش‌های جنگو

در هسته جنگو، برای راحتی و سهولت کار بخش‌هایی طراحی و پیاده‌سازی شده است. برخی از این بخش‌ها عبارتند از:

- میزبان وب مستقل و کوچک برای آزمون^{۴۳} برنامه
 - سیستمی برای معتبرسازی^{۴۴} و سری‌سازی^{۴۵} فرم‌هایی با زبان نشانه‌گذاری ابرمتنی
 - چهارچوبی برای نهان‌سازی اطلاعات جهت استفاده مجدد با استفاده از حافظه نهان^{۴۶} که روش‌های مختلف ذخیره و بازیابی حافظه نهان را در اختیار قرار می‌دهد.
 - پشتیبانی از ابزارهای میانی^{۴۷} که امکان اجرای توابع و دستورات مورد نظر را در مراحل مختلف پردازش یک درخواست، فراهم می‌کند.
 - یک توزیع‌کننده^{۴۸} درونی که به بخش‌های مختلف برنامه امکان دریافت سیگنال‌ها و رویدادهای مختلف را می‌دهد.
 - سیستم بین‌الملل‌سازی^{۴۹} که حتی امکان ترجمه بخش‌های مختلف جنگو به زبان‌های مختلف را فراهم می‌کند. از این ویژگی جهت تنظیم ساعت محلی نیز استفاده می‌شود.
 - سیستمی برای تسلسل و سری‌سازی که امکان کار با انواع داده‌های مبتنی بر زبان نشانه‌گذاری امتدادپذیر^{۵۰} و نماد شیء جاوااسکریپت^{۵۱} را فراهم می‌کند.
 - سیستمی برای توسعه قابلیت‌های موتور قالب. مانند تعیین دسترسی برای هر بخش مختلف از سایت.
- همچنین بسته جنگو شامل ابزارها و برنامه‌های جانبی مختلفی می‌باشد که در داخل بسته contrib قرار دارند. برخی از این ابزارها عبارتند از:

⁴³Test

⁴⁴Validation

⁴⁵Serialization

⁴⁶Cache

⁴⁷Middleware

⁴⁸Dispatcher

⁴⁹Internationalization

⁵⁰eXtensible Markup Language (XML)

⁵¹JavaScript Object Notation (JSON)

- سیستم تصدیق و شناسایی کاربر^{۵۲}
- سیستم تعیین دسترسی به کاربر^{۵۳}
- رابط مدیریتی پویا
- ابزارهایی برای ایجاد RSS و Atom
- سیستم نظردهی انعطاف‌پذیر
- یک چارچوب برای ایجاد برنامه‌های کاربردی GIS
- ابزارهایی برای تولید نقشه سایت^{۵۴}
- ابزارهای امنیتی برای جلوگیری از حملات CSRF
- کتابخانه‌های قالب که امکان استفاده از زبان‌های نشانه‌گذاری کم حجم مانند Textile و Markdown را فراهم می‌سازد.

۳-۵-۴ انطباق با میزبان‌های مختلف

جنگو با استفاده از واحد^{۵۵} mod_python به‌خوبی بر روی میزبان‌های وب آپاچی^{۵۶} قابل اجرا است. همچنین بر روی تمامی میزبان‌هایی که از WSGI پشتیبانی می‌کنند قابل اجرا می‌باشد. به غیر از موارد ذکر شده، جنگو توانایی راه‌اندازی سرور FastCGI را دارا می‌باشد.

۳-۶ چهارچوب نرم‌افزاری REST جنگو

برای آشنایی با این چهارچوب، ابتدا دو موضوع رابط برنامه‌نویسی نرم‌افزار^{۵۷} و REST باید توضیح داده شود.

⁵²Authentication

⁵³Authorization

⁵⁴Sitemap

⁵⁵Module

⁵⁶Apache

⁵⁷Application Programming Interface (API)

۳-۶-۱ رابط برنامه‌نویسی نرم‌افزار چیست؟

در حوزه توسعه نرم‌افزار، واژه رابط برنامه‌نویسی نرم‌افزار بسیار تکرار می‌شود. در واقع، از آنجا که می‌توان واژه Interface را به «فصل مشترک» در فارسی ترجمه کرد، می‌توان گفت که رابط برنامه‌نویسی نرم‌افزار فصل مشترکی مابین دو نرم‌افزار یا اپلیکیشن است.

آشنایی با انواع رابط‌های برنامه‌نویسی نرم‌افزار

با در نظر گرفتن این نکته که رابط برنامه‌نویسی نرم‌افزار سازوکاری است که از طریق آن تعامل سیستم با سیستم به جای تعامل کاربر با سیستم صورت می‌گیرد، می‌توان دسته‌بندی‌های مختلفی در نظر گرفت که عبارتند از:

- رابط برنامه‌نویسی در سخت‌افزار
- رابط برنامه‌نویسی نرم‌افزاری در سطح سیستم‌عامل
- رابط برنامه‌نویسی نرم‌افزاری در زبان‌های برنامه‌نویسی
- کیت‌های توسعه نرم‌افزار^{۵۸}
- رابط برنامه‌نویسی نرم‌افزاری تحت وب (وب سرویس)

تقسیم‌بندی رابط‌های برنامه‌نویسی نرم‌افزار از بُعد سطح دسترسی

علاوه بر تقسیم‌بندی‌های فوق، رابط برنامه‌نویسی نرم‌افزارهای تحت وب را می‌توان از نقطه نظر سطح دسترسی به دسته‌های مختلفی تقسیم‌بندی کرد که عبارتند از:

- رابط برنامه‌نویسی نرم‌افزار باز^{۵۹}: این دست رابط‌های برنامه‌نویسی نرم‌افزار که اصطلاحاً Public APIs نیز نامیده می‌شوند، بدون هیچ‌گونه محدودیت در سطح دسترسی برای کاربردهای مجموعه‌های تجاری به مشتری^{۶۰}، در اختیار برنامه‌نویس‌ها قرار می‌گیرند.

^{۵۸}Software Development Kit (SDK)

^{۵۹}Open API

^{۶۰}Business to Customer (B2C)

- رابط برنامه‌نویسی نرم‌افزار شریک^{۶۱}: این گروه از رابط‌های برنامه‌نویسی نرم‌افزار صرفاً در اختیار کسب‌وکارهای به اصطلاح تجاری به تجاری^{۶۲} و تجاری به مشتری است و برخلاف مورد قبل هر برنامه‌نویسی به آن‌ها دسترسی ندارد؛ برای استفاده از آن‌ها معمولاً باید هزینه کرد.

- رابط برنامه‌نویسی نرم‌افزار داخلی^{۶۳}: این گروه از رابط‌های برنامه‌نویسی نرم‌افزار که تحت عنوان Private APIs نیز شناخته می‌شوند، صرفاً برای استفاده‌های داخلی یک سیستم طراحی می‌شوند.

۳-۶-۲ REST چیست؟

با توجه به تعریف [؟] REST^{۶۴} یک مدل معماری برای طراحی برنامه‌های کاربردی شبکه است. ایده اصلی این معماری این است که به جای استفاده از مکانیزم‌های پیچیده‌ای مانند CORBA، RPC یا SOAP برای اتصال ماشین‌ها، از HTTP و به سادگی برای برقراری ارتباط بین ماشین‌ها استفاده شود.

این مدل دارای ویژگی‌های زیر است:

- بر پایه مشتری-میزبان^{۶۵} است.

- بدون حالت^{۶۶} است.

- از حافظه نهان^{۶۷} پشتیبانی می‌کند.

- سیستم لایه‌بندی شده است.

- دارای واسط واحد^{۶۸} است.

⁶¹Partner API

⁶²Business to Business (B2B)

⁶³Internal API

⁶⁴Representational State Transfer

⁶⁵client-server

⁶⁶Stateless

⁶⁷Cache

⁶⁸Uniform Interface

به سیستمی که این قیود را رعایت نماید، RESTful می‌گویند. از لحاظ رویکرد برنامه‌نویسی REST جایگزینی ساده برای سرویس‌های وب است. توسعه‌پذیری در تعاملات میان اجزا، عمومیت واسط‌ها و توسعه مستقل اجزا از کلیدی‌ترین اهداف این معماری است.

استفاده از معماری REST در برنامه‌نویسی، کارایی، سادگی، انعطاف‌پذیری، امکان مشاهده و نظارت، قابلیت حمل و قابلیت اطمینان را افزایش می‌دهد. امروزه برنامه‌های سنتی وب در حال حرکت به سمت سرویسی شدن هستند. بدین صورت که برنامه سمت مشتری تنها از طریق وب‌سرویس‌ها با سرور در تماس هستند. به بیانی دیگر ارتباط سمت مشتری با لایه داده برنامه، از طریق وب‌سرویس‌ها صورت می‌پذیرد. در چنین برنامه‌هایی منطق برنامه کاملاً در سمت مشتری پیاده‌سازی می‌شود و سمت میزبان دیگر هیچ نقشی جز فراهم کردن داده برای برنامه‌های سمت مشتری را برعهده ندارد.

یکی از نکات مثبت این معماری این است که دسترسی برنامه سمت مشتری به منابع، از طریق درخواست‌های HTTP انجام می‌گیرد. این درخواست‌ها با روش^{۶۹}های مختلفی می‌توانند ارسال شوند که هر یک معنا و مفهومی خاصی دارد. هریک از این روش‌ها در زیر تعریف شده‌اند:

- روش GET به منظور بازیابی و خواندن منبع استفاده می‌شود.
- روش POST زمانی استفاده می‌شود که بخواهیم منبع جدیدی را ایجاد کنیم.
- روش DELETE به منظور حذف یک منبع مورد استفاده قرار می‌گیرد.
- روش‌های PUT و PATCH برای دست‌کاری در یک منبع مورد استفاده قرار می‌گیرند.

۳-۶-۳ جنگو REST

چهارچوب نرم‌افزاری جنگو REST [۲]، یک ابزار قدرتمند و انعطاف‌پذیر برای ساختن رابط‌های برنامه‌نویسی نرم‌افزارهای شبکه وب است. برخی از ویژگی‌های این چهارچوب نرم‌افزاری به صورت زیر است:

- خط مشی‌های تأیید اعتبار شامل بسته‌هایی برای OAuth2 و OAuth1a
- پشتیبانی از منابع داده ORM^{۷۰} و غیر ORM

^{۶۹}Method

^{۷۰}Object-Relational Mapping

- مستندات گسترده و پشتیبانی
- مورد استفاده و مورد اعتماد شرکت‌های معتبر بین المللی از جمله موزیلا، Red Hat، Heroku و Eventbrite است.

۷-۳ چهارچوب نرم‌افزاری Vue.js

Vue.js [۶] یک چارچوب متن باز JavaScript برای ساخت رابط‌های کاربری^{۷۱} و برنامه‌های تک‌صفحه‌ای^{۷۲} است. این چهارچوب نخستین بار توسط ایوان یو^{۷۳} ایجاد شده است و اکنون توسط او و بقیه اعضای تیم اصلی که از شرکت‌های مختلفی چون Netlify و Netguru آمده‌اند، نگهداری می‌شود.

در ادامه برخی از ویژگی‌های این چهارچوب توضیح داده خواهد شد.

۱-۷-۳ اجزاء

هدف این اجزا^{۷۴}، افزایش استفاده مجدد از کد است. در مواقعی که در چندین بخش مختلف به یک المان خاص در طراحی صفحات نیاز است، از این اجزا استفاده می‌شود.

۲-۷-۳ قالب‌ها

Vue از یک نحو^{۷۵} که مشابه قوانین نوشتاری زبان نشانه‌گذاری ابرمتنی است، استفاده می‌کند. همه قالب‌های Vue دارای زبان نشانه‌گذاری ابرمتنی معتبری هستند که توسط مرورگرهای استاندارد قابل تجزیه^{۷۶} است. Vue این امکان را می‌دهد تا قالب‌ها قبل از بروزرسانی مرورگر، تغییرات لازم را اعمال کنند.

کاربران Vue می‌توانند از نحو استاندارد در قالب‌ها استفاده کنند و مستقیماً و با استفاده از JSX بنویسند.

⁷¹User Interface

⁷²Single Page Application

⁷³Evan You

⁷⁴Componenets

⁷⁵Syntax

⁷⁶Parse

۳-۷-۳ انتقال‌ها

Vue روش‌های مختلفی را برای اعمال جلوه‌های انتقال^{۷۷} هنگام وارد کردن، به روزرسانی یا حذف شدن از DOM^{۷۸} را فراهم کرده است. این موارد عبارتند از:

- اعمال خودکار کلاس‌ها برای انتقال‌ها و انیمیشن‌های CSS
- تعامل با کتابخانه‌های شخص ثالث CSS، مانند Animate.css
- تعامل با کتابخانه‌های شخص ثالث جاوا اسکریپت، مانند Velocity.js

۴-۷-۳ چهارچوب نرم‌افزاری Nuxt.js

Nuxt.js^[۳] یک چارچوب برنامه وب منبع آزاد و مبتنی بر Vue.js، Node.js، Webpack و Babel.js است. ابتدا به معرفی Vue.js پرداخته می‌شود.

از مزایای استفاده از این چهارچوب^{۷۹}، کاهش زمان پاسخگویی برنامه و همچنین بهبود بهینه سازی موتور جستجو^{۸۰} در مقایسه با سایر برنامه‌های تک صفحه‌ای است. دلیل کاهش زمان تعامل این است که محتوای کامل هر صفحه، قبل از اجرای جاوا اسکریپت در برنامه سمت مشتری، توسط وب سرور ارائه^{۸۱} می‌شود. به بیان دیگر، با استفاده از این چهارچوب، به طور همزمان از مزایای نمایش سنتی HTML و همچنین نمایش به وسیله برنامه‌های تک صفحه‌ای جدید بهره برده می‌شود. اما فایده اصلی این چهارچوب، ساده سازی و یکپارچه سازی پروژه سمت کاربر برای برنامه نویس است.

۸-۳ پایگاه داده رابطه‌ای Postgress

Postgress^[۴]، که همچنین با عنوان Postgres شناخته می‌شود، یک سیستم مدیریت پایگاه داده رابطه‌ای^{۸۲} آزاد و متن باز است. ابتدا به مفاهیم پایگاه داده رابطه‌ای پرداخته می‌شود و سپس

⁷⁷Transitions

⁷⁸Document Object Model

⁷⁹Framework

⁸⁰SEO

⁸¹Render

⁸²Relational

پایگاه داده Postgress معرفی می‌گردد.

۳-۸-۱ پایگاه داده رابطه‌ای

پایگاه داده رابطه‌ای به آن دسته از پایگاه‌های داده اطلاق می‌شود که بر اساس مدل رابطه‌ای طراحی و ایجاد شده باشند. پس از پایگاه‌های داده‌ای سلسله مراتبی و شبکه‌ای، که هر یک دارای ضعف‌هایی بودند، متخصصان در جستجوی مدلی بودند که دارای ساختار داده‌ای با انتزاع قوی باشد. مدل رابطه‌ای در سال ۱۹۷۰ توسط ادگار کاد^{۸۳} مطرح شد. این مدل دارای ساختار داده‌ای با انتزاع قوی بوده و اساساً ساختار داده‌ای در آن بر اساس یک مفهوم ریاضی به نام رابطه استوار است. در اینجا لازم است به این نکته توجه شود که مفهوم رابطه با مفهوم ریاضی آن تاحدودی متفاوت است.

برای طراحی پایگاه داده‌ها در سطح انتزاعی پایین‌تر از سطح مدل‌سازی، به یک ساختار داده‌ای از یک مدل داده‌ای نیاز است و اساساً همین مدل داده‌ای تأمین‌کننده محیط انتزاعی است. در پایگاه داده رابطه‌ای بالاخص در محیط انتزاعی مورد استفاده کاربر، رابطه نمایشی جدولی دارد و اساساً پایگاه داده رابطه‌ای مجموعه‌ای است از تعدادی نوع جدول. مفاهیم ساختار جدولی عبارتند از: سطر، جدول و ستون هر جدول از نظر محتوای داده‌ای مجموعه‌ای است از نمونه‌های متمایز از انواع سطرها و هر سطر نیز مجموعه‌ای از مقادیر است که هر کدام از یک مجموعه برگرفته شده‌اند. به هر یک از عناصر سطر یک ستون گویند. لازم است ذکر شود که در ساختار جدولی، تنها عنصر ساختاری اساسی همین مفهوم نوع جدول است.

ویژگی‌های رابطه

رابطه به عنوان تنها عنصر ساختاری اصلی در مدل رابطه‌ای برای نمایش انواع موجودیت‌ها و انواع ارتباطات به کار می‌رود. در واقع در مدل رابطه‌ای هم نوع موجودیت و هم نوع ارتباط با مفهوم رابطه نمایش داده می‌شوند و در نتیجه هم نمونه موجودیت و هم نمونه ارتباط با مفهوم تاپل نشان داده می‌شوند. رابطه دارای چهار ویژگی زیر می‌باشد:

- رابطه تاپل تکراری ندارد، زیرا بدنه رابطه مجموعه‌است و مجموعه نمی‌تواند عنصر تکراری داشته باشد.

⁸³Edgar F.Codd

- تاپل‌ها نظم ندارند زیرا بدنه رابطه مجموعه است و مجموعه در حالت کلی فاقد نظم است.
- صفات رابطه نظم مکانی ندارند زیرا سرآیند، رابطه مجموعه است و مجموعه در حالت کلی فاقد نظم است.
- تمام صفات تک مقداری (تجزیه نشدنی) هستند زیرا در نمایش جدولی رابطه، در تقاطع هر سطر و ستون باید یک مقدار وجود داشته باشد.

انواع کلیدها در مدل رابطه‌ای

- ابر کلید: هر ترکیبی از صفات جدول را که یکتایی مقدار داشته باشد، ابر کلید گویند. به بیانی دیگر هر زیر مجموعه عنوان رابطه را دارد، که یکتایی مقدار در بدنه رابطه را داشته باشد. تعریف دیگر ابر کلید عبارت است از هر ترکیبی از اسامی صفات رابطه که در هیچ دو تاپل مقدار یکسان نداشته باشد.
- کلید کاندید: کلید کاندید امکانی است برای ارجاع به «تک تاپل» در رابطه. مجموعه صفات k از رابطه R یک کلید کاندید است، اگر دارای خاصیت غیر کاهشی و یکتایی باشد.
- کلید اصلی:
- یکی از کلیدهای کاندید رابطه که شرایط زیر را داشته باشد:
 ۱. شناسایی‌کننده نوع موجودیت در رابطه باشد. مانند شماره عضویت کتابخانه برای هر دانشجو.
 ۲. از نظر طول، کوتاه‌تر باشد. یعنی بین دو کلید کاندید، کلید کوتاه‌تر برای کلید اصلی بودن بهتر است.
- کلید جانشین: هر کلید کاندید به غیر از کلید اصلی را کلید جانشین گویند.
- کلید خارجی: دو رابطه R_1 و R_2 را در نظر بگیرید. هر زیر مجموعه از صفات رابطه R_2 که هر مقدار معلوم آن با یک مقدار از کلید کاندید R_1 برابر باشد، کلید خارجی در رابطه R_2 است. نقش کلید خارجی برای نمایش ارتباطات بین انواع موجودیت‌ها و در نتیجه بین نمونه‌های آنها به کار می‌رود.

۲-۸-۳ پایگاه داده PostgreSQL

Postgress یک سامانه مدیریت پایگاه داده‌های رابطه‌ای است که برای سیستم‌های مختلفی از جمله لینوکس، فری بی‌اس‌دی^{۸۴}، ویندوز و مک اواس macOS موجود است. این پایگاه داده توسط گروه توسعه سراسری PostgreSQL توسعه داده می‌شود، که شامل تعداد زیادی از افراد داوطلب است. این پایگاه داده بر توسعه‌پذیری و رعایت استانداردهای فنی تأکید دارد. همچنین توانایی پاسخگویی به بار کاری زیاد، در مقیاس یک کامپیوتر تا انبارهای داده^{۸۵}، برای استفاده همزمان کاربران را دارد.

تراکنش‌های PostgreSQL شامل ویژگی‌های زیر است:

- اتمی^{۸۶}
- پایداری^{۸۷}
- ایزوله^{۸۸}
- با دوام^{۸۹}
- بروزرسانی خودکار نمایه‌ها^{۹۰}
- پشتیبانی از تریگر^{۹۱}
- پشتیبانی از کلیدهای خارجی
- پشتیبانی از توابع تجمیعی
- کلیدهای خارجی
- پشتیبانی از رویه‌های ذخیره شده^{۹۲}

⁸⁴FreeBSD

⁸⁵Data Warehouse

⁸⁶Atomicity

⁸⁷Consistency

⁸⁸Isolation

⁸⁹Durability

⁹⁰Automatically Updatable Views

⁹¹Trigger

⁹²Stored Procedures

۹-۳ ماشین مجازی

برای پیاده‌سازی این پروژه در محیط عملیاتی از میزبان مجازی شخصی^{۹۳} استفاده شده است. برای این که بدانیم میزبان مجازی شخصی چیست، ابتدا باید با ماشین مجازی^{۹۴} آشنا شویم. یک ماشین مجازی از منابع فیزیکی یک کامپیوتر استفاده می‌کند. این منابع می‌توانند پردازنده، حافظه رم و یا دیسک‌های ذخیره‌سازی باشند. ماشین‌های مجازی با بهره‌گیری از این منابع، یک نوع از کامپیوتر را شبیه‌سازی می‌کنند. برای درک بهتر، کافی است امکان تصویر در تصویر یک تلویزیون را تصور کنید. یک تلویزیون امروزی می‌تواند همزمان دو تصویر از دو شبکه متفاوت را پخش کند. این امکان در حالی برای شما فراهم می‌شود که در حال استفاده از یک تلویزیون هستید؛ اما نتیجه‌ای که حاصل می‌شود، دو تصویر همزمان است. پس در این حالت از یک منبع واحد، حداقل دو خروجی دریافت کرده‌اید. پس یک یا چند ماشین مجازی می‌تواند درون یک سخت‌افزار یک کامپیوتر واحد اجرا شود. به‌عنوان مثال می‌توان یک ماشین مجازی با سیستم‌عامل ویندوز را درون یک کامپیوتر که خود نیز از سیستم‌عامل ویندوز بهره می‌برد اجرا کرد. این امکان وجود دارد که چندین سیستم‌عامل را از طریق ماشین‌های مجازی به واسطه یک سخت‌افزار واحد اجرا کرد. این دقیقاً همان کاری است که شرکت‌های ارائه میزبانی^{۹۵} انجام می‌دهند. اما این بار تفاوت آن‌جا است که آن‌ها به جای کامپیوترهای شخصی از سرورهای پر قدرت استفاده می‌کنند. سرورهایی که هر کدامشان می‌توانند بیش از ۴۸ هسته پردازشگر و بیش از ۱ ترابایت حافظه رم داشته باشند. تعداد زیادی از این سرورها در محل‌هایی به نام مراکز داده^{۹۶} نگهداری می‌شوند. هر مرکز داده نیز می‌تواند بیش از صدها و شاید هزاران سرور را درون خود جای دهد. هر کدام از این سرورها نیز توانایی اجرای چندین و شاید صدها ماشین مجازی را دارند. این ماشین‌های مجازی می‌توانند توسط سازمان‌ها، ادارات و یا حتی افراد استفاده شوند. اگر یک ماشین مجازی اجاره داده شود، اصطلاحاً به آن سرور اختصاصی مجازی یا میزبان مجازی شخصی می‌گویند. پس بنابراین یک میزبان مجازی شخصی یک ماشین مجازی به حساب می‌آید و فقط نام تجاری به خود گرفته‌است.

⁹³Virtual Private Server (VPS)

⁹⁴Virtual Machine

⁹⁵Hosting

⁹⁶Data center

۳-۹-۱ دلایل استفاده از میزبان مجازی شخصی

میزبان مجازی شخصی به عنوان یک سرویس به کاربر ارائه می‌شود. اما دیگر سرویس‌هایی که در مقابل میزبان مجازی شخصی قرار می‌گیرند، هاست‌های اختصاصی و هاست‌های اشتراکی هستند. هاست‌های اختصاصی به صورت کامل در اختیار کاربران قرار می‌گیرند. به این صورت که مشتری یک سرور کامل را اجاره کرده و آن را در تحت نظارت و استفاده می‌گیرد. به این ترتیب هیچ‌کس دیگری به آن سخت‌افزار دسترسی نداشته و کنترل کامل سخت‌افزار در اختیار اجاره کننده است. در مقابل، هاستینگ اشتراکی بدین معناست که در یک سرور واحد، چندین برنامه اجرا شده و چندین مشتری همزمان از سخت‌افزار استفاده می‌کنند. هاست‌های اختصاصی معمولاً از منابع سخت‌افزاری غنی‌تری نسبت به میزبان مجازی شخصی بهره برده و البته اجاره آن‌ها نیز بسیار گران است. در حالی که اجاره میزبان‌های اشتراکی نسبت به هاست‌های اختصاصی ارزان‌تر بوده ولی به هیچ عنوان انعطاف‌پذیری میزبان مجازی شخصی را ندارند. می‌توان گفت اجاره یک میزبان مجازی شخصی نیز فقط کمی گران‌تر از هاست‌های اشتراکی است. میزبان مجازی شخصی قابل انعطاف است. چرا که شما تقریباً هر کاری را می‌توانید با یک ماشین مجازی ویندوزی انجام دهید. می‌توان از یک میزبان مجازی شخصی به عنوان فضایی برای ذخیره‌سازی اطلاعات شخصی استفاده کرد.

نکته ای راجع به یک میزبان مجازی شخصی باید در نظر گرفته شود، این است که در صورت استفاده نادرست از آن، کاربر نمی‌تواند هیچ‌گونه آسیبی به سرور اصلی بزند. به عنوان مثال اگر یک میزبان مجازی شخصی آلوده به ویروس شود، دیگر ماشین‌های مجازی موجود در سرور در امان خواهند بود و کوچک‌ترین اختلالی در نحوه سرویس‌دهی آن‌ها پیش نخواهد آمد. یک میزبان مجازی شخصی می‌تواند به راحتی راه‌اندازی مجدد شده و یا حتی در عرض چند دقیقه دوباره با یک سیستم‌عامل جدید در اختیار کاربر قرار گیرد. تنها نکته‌ای که باید مدنظر داشت، تهیه نسخه پشتیبان از اطلاعات حیاتی است، چرا که آن‌ها هنگام نصب دوباره سیستم‌عامل از بین خواهند رفت. ماشین‌های مجازی ایمن هستند. به این معنی که هر ماشین مجازی به صورت جداگانه از منابع استفاده کرده و دسترسی کاربران به ماشین مجازی فرد دیگری، غیرممکن است. اما این مهم در صورت استفاده از میزبان‌های اشتراکی کمی متفاوت است. به دلیل قرارگیری اپلیکیشن‌ها و وب‌سایت‌ها در کنار همدیگر و مدیریت آن‌ها توسط نرم‌افزاری واحد مانند وب‌سرورها؛ امکان این که دیگر اپلیکیشن‌ها مورد سوءاستفاده توسط دیگر کاربران قرار گیرد، وجود دارد.

۳-۹-۲ موارد استفاده از میزبان مجازی شخصی

در ذیل به مواردی خواهیم پرداخت که با استفاده از یک میزبان مجازی شخصی می‌توان به آن‌ها دست پیدا کرد.

اجرای یک وبسایت

این روزها بهترین راه اجرای یک وبسایت از طریق میزبان مجازی شخصی است. معمولاً برای یک میزبان مجازی شخصی منابع اختصاصی در نظر گرفته می‌شود. به این صورت که بخشی از منابع که اختیار کرده‌اید، همواره در اختیار شما خواهد بود و برنامه دیگری امکان اشغال آن را ندارد.

ایجاد یک سرور یا سرویس

شاید یک سازمان، شرکت و یا حتی اشخاص بخواهند برای خود سرویس ایمیل اختصاصی داشته باشند. با استفاده از میزبان مجازی شخصی و نرم‌افزار سرویس ایمیل می‌توان سرور اختصاصی پست الکترونیک خود را داشته و شخصاً آن را مدیریت کرد. این راه کار بسیار مورد استقبال شرکت‌ها قرار گرفته است.

ایجاد یک محیط آزمایشی

با استفاده از یک میزبان مجازی شخصی می‌توان به آزمایش و اصلاح انواع نرم‌افزارهای گوناگون پرداخت. اشاره شد که اگر اتفاق ناگواری برای یک میزبان مجازی شخصی بیافتد دیگر ماشین‌های مجازی از این رخداد صدمه‌ای نخواهند دید. در صورتی که سیستم‌عامل یک میزبان مجازی شخصی دچار اشکال شود، به راحتی هرچه تمام‌تر می‌توان سیستم‌عامل جدیدی را جایگزین کرد. این در صورتی است که اگر این اتفاق برای یک هاست اختصاصی رخ دهد، نصب و راه‌اندازی یک سیستم‌عامل جدید کاری پردردسر و زمان‌بر خواهد بود.

۳-۱۰ نتیجه‌گیری

همانطور که در قسمت‌های مختلف این بخش به تفصیل توضیح داده شد، این پروژه از قطعات سخت‌افزاری، تکنولوژی‌های نرم‌افزاری و همچنین محیط‌هایی برای عملیاتی کردن استفاده شده

است. برای پیاده‌سازی سخت‌افزاری پروژه از برد رزبری پای مدل 3A استفاده شد. همچنین جهت تعامل این برد با کاربر، از نمایشگر لمسی 3.5 اینچی Waveshare استفاده شد. برنامه‌ای که بر روی نمایشگر نمایش داده می‌شود، با استفاده از چهارچوب نرم‌افزاری سمت کاربر Nuxt.js که خود بر پایه Vue.js نوشته شده است، برنامه‌نویسی شد. این برنامه با استفاده از وب‌سرویس REST با چهارچوب نرم‌افزاری Django در تعامل است.

برای عملیاتی کردن پروژه بر روی محیطی که همه کاربران به برنامه‌ها دسترسی داشته باشند از ماشین مجازی استفاده شد.

فصل چهارم

طراحی و پیاده‌سازی

در این بخش، طراحی و پیاده‌سازی بخش‌های مختلف سخت‌افزاری و نرم‌افزاری شرح داده خواهد شد.

۴-۱ مقدمه

به صورت کلی، طراحی پروژه در تصویر ۴-۱ است. همانطور که در تصویر مشخص شده است، میزبان اطلاعات را از پایگاه داده دریافت کرده و به هر دو برنامه سمت مشتری می‌فرستد. برنامه‌ای که جمع‌آوردگان سفارش از آن استفاده می‌کنند به صورت وب و در تلفن همراه هوشمند نمایش داده می‌شود. همچنین برنامه‌ای که برای تعیین کردن جایگاه محصولات استفاده می‌شود، بر روی نمایشگر لمسی که به یک برد رزبری پای مدل 3A وصل شده است نمایش داده می‌شود. در این پروژه، رزبری پای صرفاً وظیفه نمایش اطلاعات به کاربر و همچنین دریافت اطلاعات از کاربر و ارسال آن به میزبان اطلاعات است. کلیه برنامه‌های سمت کاربر، برنامه‌های سمت میزبان و همچنین پایگاه داده بر روی یک ماشین مجازی^۱ اجرا شده‌اند و تمامی فرآیندها بر روی منابع ماشین مجازی انجام می‌شوند. به همین دلیل رزبری پای کمترین فشار محاسباتی را متحمل می‌شود. برای راه‌اندازی اولیه پروژه، ماشین مجازی با واحد محاسباتی تک هسته‌ای^۲، ۲ گیگابایت ظرفیت حافظه^۳ و همچنین ۵۰ گیگابایت ظرفیت حافظه سخت^۴ در نظر گرفته شده است. در قسمت‌های بعد به تفصیل، طراحی و پیاده‌سازی هر کدام از این عناصر توضیح داده خواهد شد.

۴-۲ نمایشگر لمسی

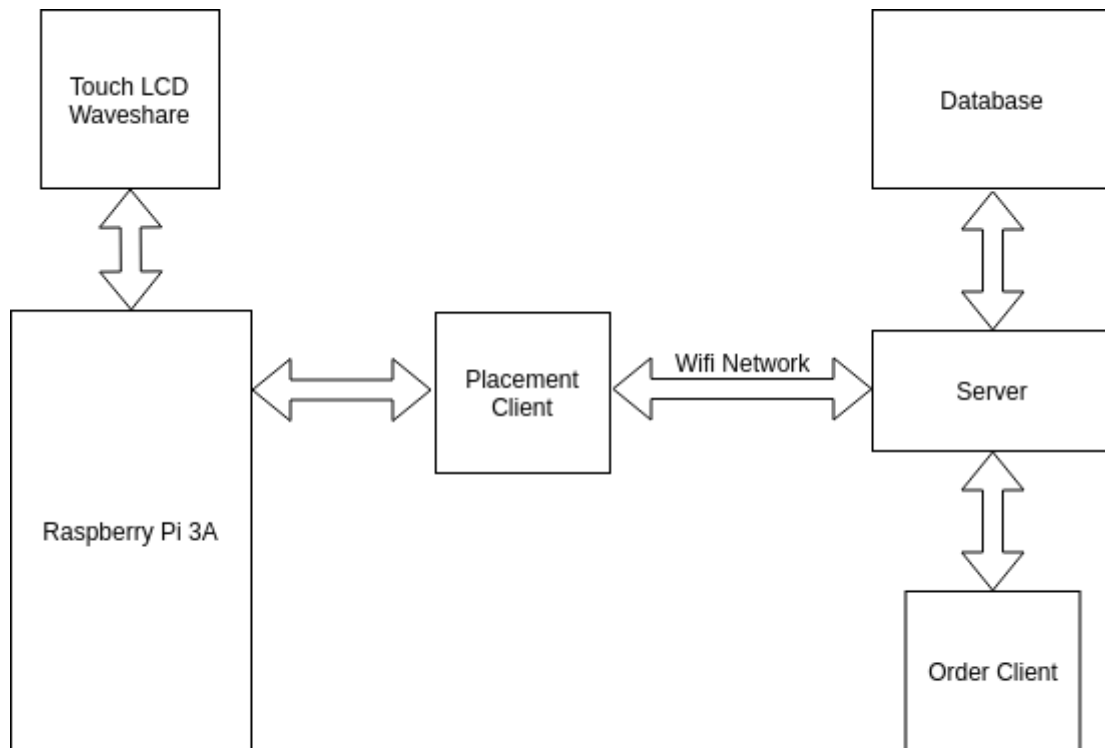
در این قسمت به راه‌اندازی نمایشگر لمسی بر روی رزبری پای پرداخته خواهد شد.

^۱Virtual Machine

^۲1 Core CPU

^۳Memory

^۴Hard Disk



شکل ۴-۱: معماری کلی پروژه

۴-۲-۱ نحوه قرارگیری بر روی پین‌های GPIO

همانطور که در تصویر ۴-۲ مشخص است، این نمایشگر باید بر روی اولین پین‌های GPIO رزبری پای قرار گیرد تا از طریق رابط سریال پیرامونی^۵ ارتباط برقرار شود.

۴-۲-۲ راه‌اندازی نمایشگر بر روی رزبری پای

پس از قرار دادن نمایشگر در جای مناسب، برد را روشن کرده و با استفاده از آدرس قرارداد اینترنتی^۶ و نام کاربری، از طریق پوستر امن^۷ ارتباط برقرار شود. یک راه برای پیدا کردن آدرس برد، وصل کردن برد به یک نمایشگر به وسیله کابل رابط چندرسانه ای وضوح بالا و باز کردن پایانه^۸ و اجرای دستور زیر است:

^۵Serial Peripheral Interface

^۶IP Address

^۷SSH

^۸Terminal



شکل ۴-۲: قرارگیری نمایشگر لمسی بر روی رزبری پای

```
hostname -I
```

حال بر روی کامپیوتر مبدا، دستور پوسته امن به صورت زیر اجرا شود:

```
ssh user@ip_address
```

توجه شود که user نام کاربری موجود بر روی سیستم عامل رزبرین است. به طور پیش فرض مقدار آن pi است. سپس رمز کاربر وارد شود.

حال باید تنظیماتی بر روی رزبری پای اعمال شود. برای این کار دستور زیر در پایانه اجرا شود:

```
sudo raspi-config
```

در این قسمت Expand Filesystem فعال شود. همچنین Boot Option بر روی Desktop Autologin قرار داده شود.

پس از انجام تنظیمات اولیه، نمایشگر آماده نصب است. برای راه‌اندازی، ابتدا راه‌انداز نمایشگر با دستور زیر بارگیری شود:

```
git clone https://github.com/waveshare/LCD-show.git  
cd LCD-show/
```

حال باید فایل صفرویکی LCD35-show را اجرا کنیم. برای این کار در ابتدا باید اجازه^۹ اجرای فایل به وسیله دستور زیر داده شود:

```
chmod +x LCD35-show
```

حال فایل زیر اجرا شود:

```
./LCD35-show
```

پس از انجام این مرحله، نمایشگر نصب شده است و با راه‌اندازی مجدد بورد، اطلاعات راه‌انداز بورد بر روی نمایشگر نشان داده خواهد شد. ولی قسمت لمسی نمایشگر باید تنظیم گردد تا محل تماس را بتواند به درستی تشخیص دهد. در مرحله بعد این تنظیم انجام می‌شود.

۳-۲-۴ تنظیم صفحه لمسی

این نمایشگر با استفاده از برنامه‌ای به نام xinput_calibrator که به طور پیش‌فرض بر روی سیستم‌عامل رزبین نصب شده است، قابل تنظیم است. برای تنظیم نمایشگر، دستور زیر در پایانه وارد شود:

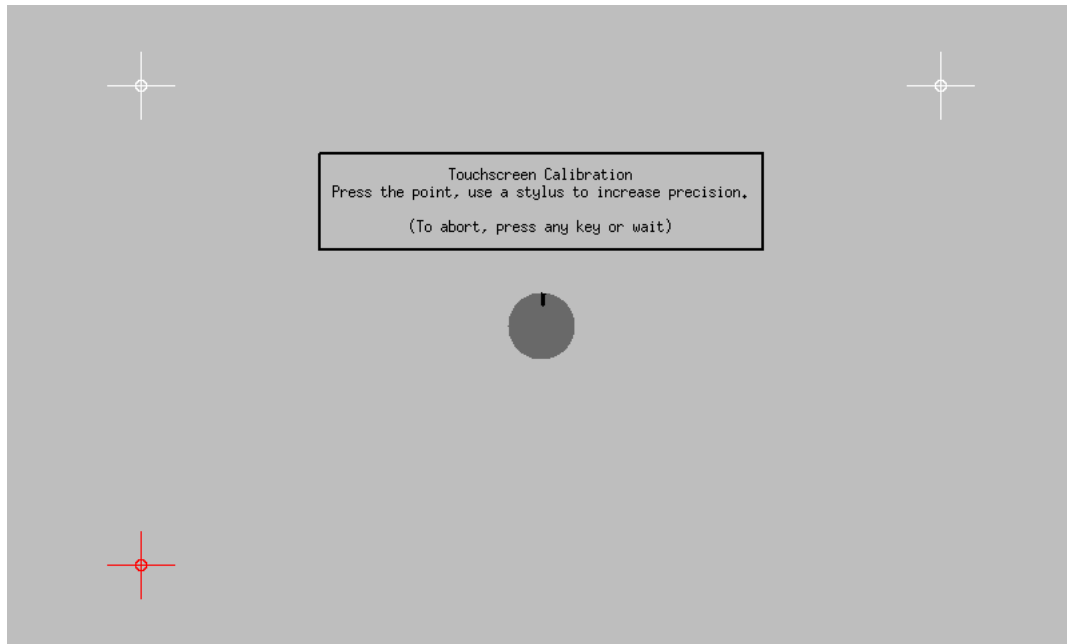
```
sudo DISPLAY=:0.0 xinput_calibrator
```

همانطور که در تصویر ۳-۴ مشخص است، نقاطی بر روی نمایشگر نشان داده می‌شود که باید به ترتیب لمس شوند تا جایگاه دقیق بخش لمسی نمایشگر تشخیص داده شود. پس از اتمام این مرحله خروجی مشابه زیر چاپ می‌شود:

```
Doing dynamic recalibration:
```

```
Setting new calibration data: 3919, 208, 236, 3913
```

^۹permission



شکل ۴-۳: تنظیم صفحه نمایش لمسی

به وسیله یک ویرایشگر، فایل 99-calibration.conf را ویرایش کرده و با اطلاعات بالا جایگزین شود. برای نمونه با استفاده از ویرایشگر VIM ای فایل در آدرس زیر قابل ویرایش است:

```
sudo vim /etc/X11/xorg.conf.d/99-calibration.conf
```

حال به وسیله دستور زیر، رزبری پای راه‌اندازی مجدد شود.

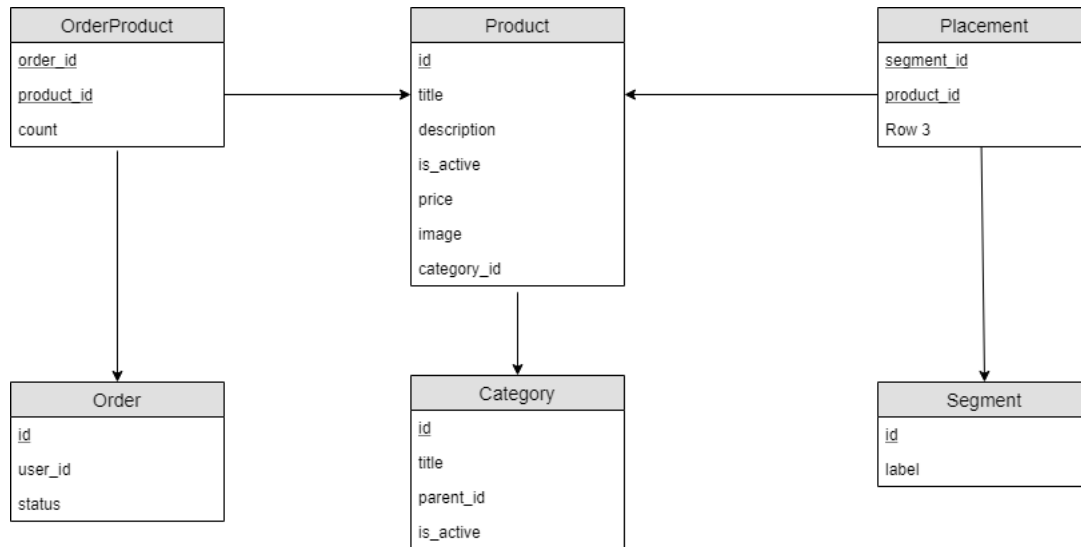
```
sudo reboot
```

۳-۴ طراحی و پیاده‌سازی پایگاه‌داده

با توجه به نیازمندی‌های مطرح شده، پایگاه‌داده به صورت نمودار ۴-۴ طراحی شد.

۱-۳-۴ محصولات

محصولات در جدول Product با مشخصه‌های زیر طراحی شد:



شکل ۴-۴: طراحی پایگاه داده

• **id**: شناسه محصول - کلید اصلی

• **title**: عنوان محصول

• **description**: توضیحات محصول

• **is_active**: فعال یا غیرفعال بودن محصول

• **price**: قیمت محصول

• **image**: تصویر محصول

• **category**: کلید خارجی به جدول Category

۲-۳-۴ سفارش‌ها

سفارش‌ها در جدول Order با مشخصه‌های زیر طراحی شد:

• **id**: شناسه سفارش - کلید اصلی

• **user_id**: شناسه مشتری

• **status**: وضعیت فعلی سفارش. وضعیت فعلی سفارش یا registered است و یا picked. اگر سفارشی ثبت شده باشد و محصولاتش جمع‌آوری نشده باشد، وضعیت registered است و اگر محصولات جمع‌آوری شد، به picked تغییر پیدا می‌کند.

۳-۳-۴ بخش‌ها

بخش‌ها در جدول Segment با مشخصه‌های زیر طراحی شد:

• **id**: شناسه بخش - کلید اصلی

• **label**: برچسب بخش

۴-۳-۴ دسته‌بندی‌ها

دسته‌بندی‌ها در جدول Category با مشخصه‌های زیر طراحی شد:

• **id**: شناسه دسته‌بندی - کلید اصلی

• **title**: عنوان دسته‌بندی

• **parent_id**: پدر دسته فعلی

• **is_active**: فعال یا غیرفعال بودن دسته

۵-۳-۴ جداول میانی

برای ارتباط بین جداول معرفی شده، چندین جدول میانی طراحی گردید که ارتباط چند به چند بین این جداول را فراهم می‌کند.

جدول Placement ارتباط بین محصولات و بخش‌های مختلف را نشان می‌دهد که دارای مشخصه‌های زیر است:

• **product_id**: شناسه محصول

• **segment_id**: شناسه بخش

• **count**: تعداد

در این جدول هیچ سطری وجود ندارد که دارای شناسه محصول و شناسه بخش یکسان باشند.

جدول OrderProduct ارتباط بین محصولات و سفارش‌های مختلف را نشان می‌دهد که دارای مشخصه‌های زیر است:

• **product_id**: شناسه محصول

• **order_id**: شناسه سفارش

• **count**: تعداد

در این جدول هیچ سطری وجود ندارد که دارای شناسه محصول و شناسه سفارش یکسان باشند.

۴-۴ رابط‌های برنامه‌نویسی نرم‌افزار

همانطور که در قسمت معرفی تکنولوژی‌های مسئله توضیح داده شد، برای پیاده‌سازی رابط‌های برنامه‌نویسی نرم‌افزار پروژه از چهارچوب نرم‌افزاری Django Rest API استفاده شد. با توجه به نیازمندی‌های مسئله و برای برقراری ارتباط با برنامه سمت کاربر، رابط‌هایی که در ادامه معرفی می‌شوند طراحی و پیاده‌سازی شدند.

۴-۴-۱ دریافت محصولات فعال

آدرس دریافت محصولات به این صورت تعریف شده است. روش^{۱۰} فراخوانی این آدرس از نوع GET است.

```
/api/products
#output
[
  {
    "id": 4,
```

¹⁰Method

```
"title": "ThinkPad T480 - E",
"description": "",
"is_active": true,
"price": 18800000,
"image": "",
"category": "لپتاپ",
"segments": [
    "B-1"
]
},
]
```

۲-۴-۴ لیست سفارش‌ها

آدرس دریافت لیست سفارش‌ها به این صورت تعریف شده است. روش فراخوانی این آدرس از نوع GET است.

```
/api/orders
#output
[
  {
    "id": 4,
    "customer": "admin",
    "status": "picked",
    "products": [
      {
        "id": 4,
        "title": "ThinkPad T480 - E",
        "description": "",
```

```
        "is_active": true,
        "price": 18800000,
        "image": "",
        "category": "لپتاپ",
        "segments": [
            "B-1"
        ]
    },
],
"created_at": "2019-08-26T09:56:29Z"
}
```

۳-۴-۴ دریافت محصولات یک سفارش

آدرس دریافت محصولات یک سفارش خاص به این صورت تعریف شده است. روش فراخوانی این آدرس از نوع GET است.

```
/api/order-products/1
#output
[
    {
        "id": 1,
        "customer": "admin",
        "status": "picked",
        "products": [
            {
                "id": 5,
                "title": "MacBook Air MQD32 2017",
```

```
        "description": "",
        "is_active": true,
        "price": 11175000,
        "image": "",
        "category": "لیتاپ",
        "segments": []
    },
],
    "created_at": "2019-08-26T09:56:29Z"
}
]
```

۴-۴-۴ لیست بخش‌ها

آدرس دریافت بخش‌ها به این صورت تعریف شده است. روش فراخوانی این آدرس از نوع GET است.

```
/api/segments
#output
[
  {
    "id": 1,
    "label": "A-20"
  },
  {
    "id": 3,
    "label": "A-1"
  },
]
```

۵-۴-۴ ثبت جایگاه یک محصول

آدرس ثبت جایگاه یک محصول در بخش مورد نظر به این صورت تعریف شده است. روش فراخوانی این آدرس از نوع POST است. بدنه درخواست^{۱۱} این رابط، لیستی از شناسه‌های محصولات است.

`/api/submit-segment/{segment_id}`

۶-۴-۴ اطلاع اتمام جمع‌آوری محصولات سفارش

آدرس اطلاع اتمام جمع‌آوری محصولات سفارش به این صورت تعریف شده است. روش فراخوانی این آدرس از نوع POST است. بدنه درخواست این رابط، لیستی از شناسه‌های محصولات است.

`/api/pick-order/{order_id}`

۵-۴ برنامه انتخاب محصولات بخش‌ها

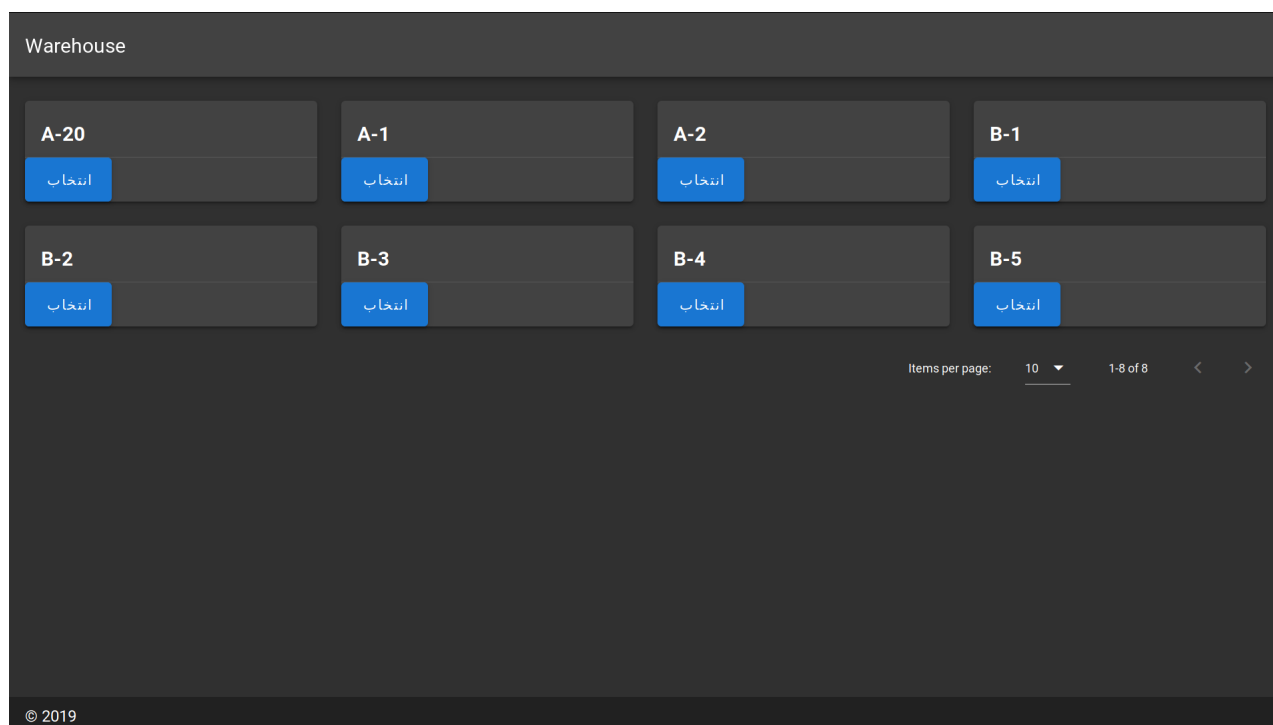
بر روی رزبری پای‌های موجود در هر بخش، مرورگری باز است که همواره یک صفحه برای انتخاب محصولات جدید آن بخش را نشان می‌دهد.

اما اولین بار که رزبری پای روشن می‌شود، آدرس زیر فراخوانی می‌شود:

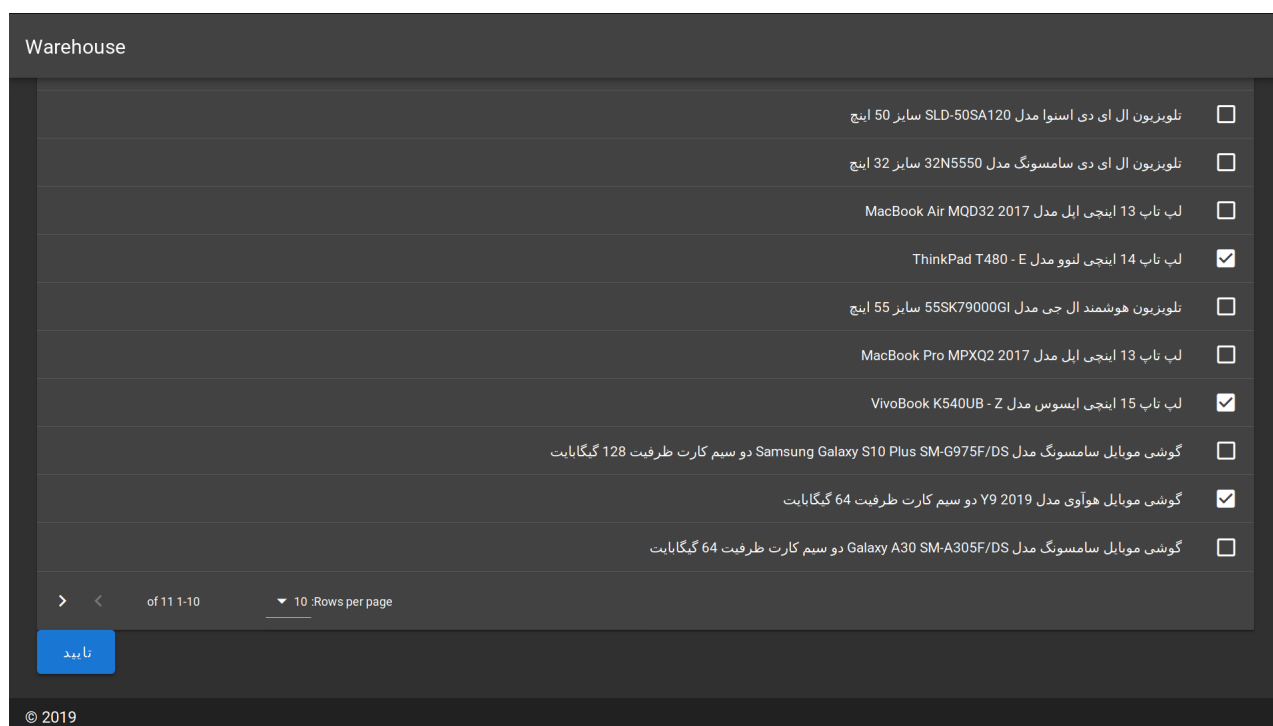
`http://ip.com/segments`

این صفحه که در تصویر ۵-۴ مشخص است، لیستی از بخش‌های موجود در انبار و یا فروشگاه را نشان می‌دهد. کاربر ابتدا باید بخش مورد نظر را انتخاب کرده تا به صفحه‌ای که به در تصویر ۶-۴ نمایش داده شده است، منتقل شود. در این صفحه لیست تمامی محصولات موجود و فعال که مدیر برنامه اضافه کرده است نمایش داده می‌شود و کاربر پس از انتخاب محصولات موردنظر، دکمه تایید را فشار می‌دهد تا اطلاعات محصولات جدید بخش، به میزبان فرستاده شود و در پایگاه داده ثبت گردد. همانطور که در تصاویر ۷-۴ و ۸-۴ مشخص است، مسئولین چیدمان

^{۱۱}Request Body



شکل ۴-۵: نمایش بخش‌ها

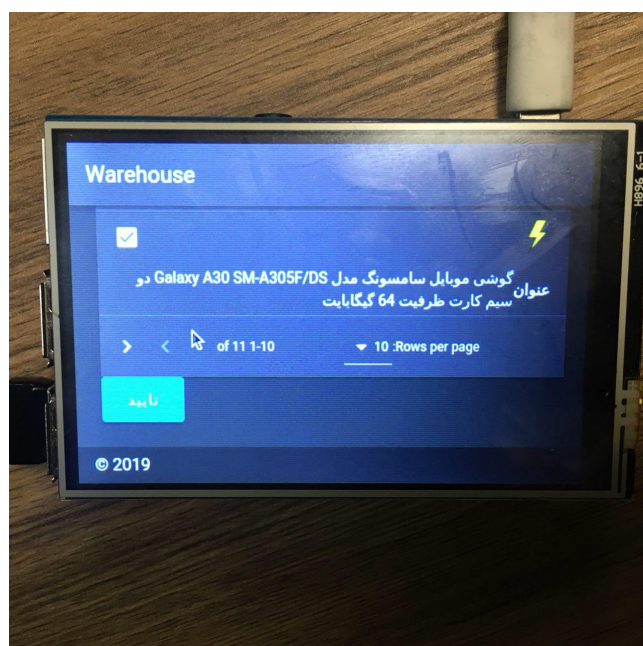


شکل ۴-۶: نمایش محصولات جهت ثبت در بخش مورد نظر

محصول در جایگاه‌ها می‌توانند با انتخاب محصولات از طریق نمایشگر و پس از لمس کردن گزینه تایید، جایگاه جدید محصولات را بروزرسانی کنند.



شکل ۴-۷: نمایش لیست محصولات برنامه انتخاب محصولات بخش‌ها بر روی رزبری پای



شکل ۴-۸: نمایش گزینه تایید برنامه انتخاب محصولات بخش‌ها بر روی رزبری پای

۴-۶ برنامه جمع‌آوری محصول

هنگامی که سفارشی ثبت می‌شود، وضعیت آن سفارش registered است. مسئولان جمع‌آوری^{۱۲} سفارش، نیاز دارند لیستی از سفارش‌های ثبت شده که هنوز جمع‌آوری نشده‌اند را ببینند. لیست

¹²picker

Warehouse					
ID	Customer	Status	Products Count	Date	Details
1	admin	picked	4 products	2019-08-26T09:56:29Z	DETAILS
<div> Rows per page: 10 1-1 of 1 </div>					

شکل ۴-۹: لیست سفارش‌ها

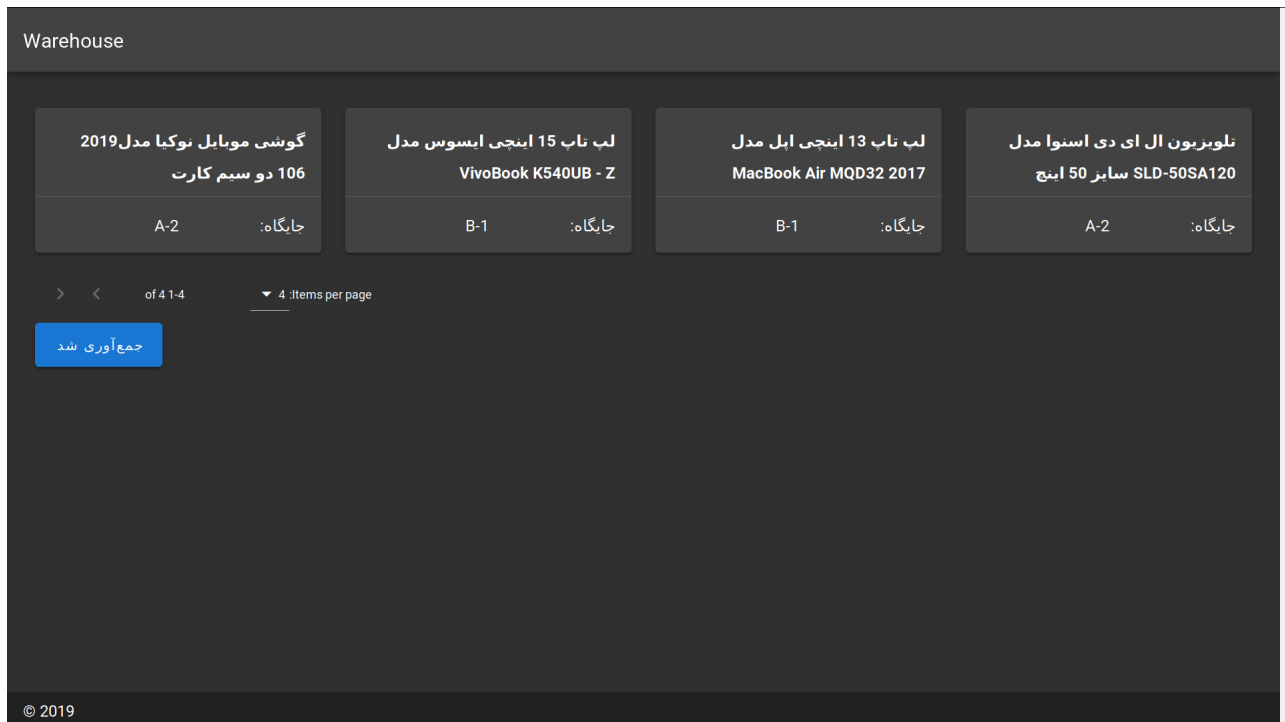
این سفارش‌ها که در تصویر ۴-۹ مشخص است، در آدرس زیر موجود است:

<http://ip.com/orders>

پس از انتخاب سفارش، مسئول جمع‌آوری لیستی از محصولات ثبت شده در سفارش را همراه با جایگاه آن‌ها مشاهده می‌کند. این صفحه در تصویر ۴-۱۰ نمایش داده شده است. هنگامی که مسئول جمع‌آوری محصولات سفارش را جمع‌آوری کرد، دکمه "جمع‌آوری شد" را فشار می‌دهد. این عمل تابعی را فراخوانی می‌کند تا وضعیت سفارش را به picked تغییر دهد.

۴-۷ رابط کاربری مدیریت

برای مدیریت عناصر موجود و همچنین با توجه به نیازمندی‌های مطرح شده، رابط کاربری برای مدیریت مجموعه طراحی و پیاده‌سازی شد. در این قسمت امکاناتی فراهم شد که هرکدام به تفصیل توضیح داده خواهد شد. این قسمت توسط پیشوند admin در دسترس است.



شکل ۴-۱۰: لیست محصولات سفارش

۴-۷-۱ احراز هویت و دسترسی

برای پیاده‌سازی احراز هویت و همچنین دادن دسترسی به کاربران، از مدل کاربر-گروه^{۱۳} جنگو استفاده شده است. در این سیستم دو ماهیت گروه و کاربر وجود دارد.

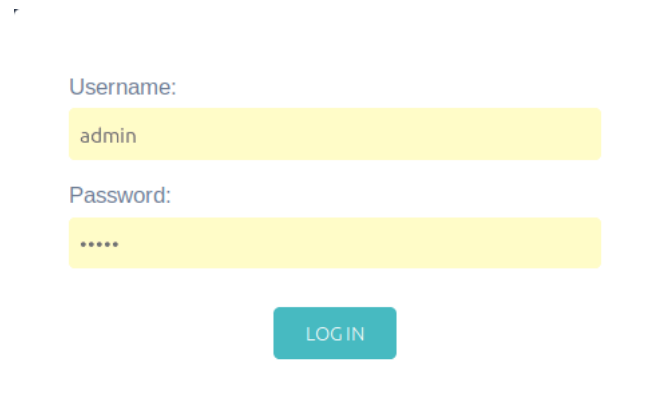
- گروه: مجموعه‌ای از افراد که برایشان دسترسی‌های یکسانی تعریف شده است.
- کاربر: هر کاربر که توسط مدیر^{۱۴} سایت ثبت‌نام می‌شود، می‌تواند به گروه‌های مختلف ملحق شود.

کاربران از طریق صفحه ۴-۱۱ که از طریق آدرس زیر در دسترس است احراز هویت می‌شوند:

<http://ip.com/admin/login>

^{۱۳}User-Group

^{۱۴}Admin



Username:

admin

Password:

.....

LOGIN

شکل ۴-۱۱: صفحه احراز هویت

۴-۷-۲ ایجاد و ویرایش عناصر پایگاه داده

برای راحتی مدیریت و کنترل محتوای پایگاه داده، صفحاتی ایجاد شد تا مدیر سایت توانایی ایجاد و ویرایش عناصر پایگاه داده، مانند محصولات، دسته‌بندی‌ها، بخش‌ها و سفارش‌ها را داشته باشد. به طور کلی این بخش شامل موارد زیر است: همچنین صفحات زیر برای مدیریت هرچه ساده‌تر پایگاه داده فراهم شد:

- اضافه کردن (تصویر ۴-۱۲)، ویرایش (تصویر ۴-۱۳) و یا حذف یک محصول
- اضافه کردن، ویرایش و یا حذف یک دسته‌بندی
- اضافه کردن، ویرایش و یا حذف یک بخش
- مشاهده و ویرایش سفارش

۴-۸ نتیجه‌گیری

در این پروژه برنامه‌های جمع‌آوری محصولات سفارش، تعیین جایگاه محصولات و همچنین رابط مدیریت پیاده‌سازی شد. برنامه جمع‌آوری محصول با استفاده از Nuxt.js و همچنین Django، داده‌های پایگاه داده را بروزرسانی می‌کند. همچنین برنامه تعیین جایگاه محصولات در بخش‌های مختلف، با استفاده از مرورگر نصب شده بر روی رزبری پای به همراه نمایشگر لمسی نصب شده روی آن، مورد استفاده قرار می‌گیرد. با استفاده از Django، رابط کاربری مدیریت نیز پیاده‌سازی شد.

HOME > WAREHOUSE > PRODUCTS > ADD PRODUCT

امین

GENERAL PLACEMENTS

Title:*

Description:

☐ Is active

Price:*

Image:

Category:*

SAVE Save and add another Save and continue editing

شکل ۴-۱۲: ایجاد محصول

HOME > WAREHOUSE > PRODUCTS > MacBook Pro MI لپ تاب 13 اینچی اپل مدل

امین

History

GENERAL PLACEMENTS

Title:*

Description:

☐ Is active

Price:*

Image:

Category:*

SAVE Save and add another Save and continue editing DELETE

شکل ۴-۱۳: ویرایش محصول

فصل پنجم

جمع‌بندی، نتیجه‌گیری و پیشنهادات

۱-۵ جمع‌بندی و نتیجه‌گیری

امروزه با گسترش فعالیت‌های محققان در زمینه‌های سخت‌افزاری و نرم‌افزاری، پیشرفت‌هایی حاصل شده است که همکاری هرچه بیشتر این دو موضوع مطالعاتی را هموار می‌کند. در صنعت‌های مختلف از قطعات سخت‌افزاری برای خودکارسازی فرآیندها و روندها استفاده می‌شود. یکی از محیط‌های با پتانسیل بالا، انبارها و فروشگاه‌های بزرگ و با تعداد محصولات زیاد هستند. در حال حاضر در کشور ما بیشتر فرآیندهای جمع‌آوری کالاها و البته بسته‌بندی به صورت انسانی و بدون کمک چندان از کامپیوتر انجام می‌گیرند. این کار باعث شده است تا مدیریت انبارهای بزرگ به گلوگاه تبدیل شود. از مهمترین مشکلات مدیریتی در انبارهای بزرگ، تعیین جایگاه محصولات است؛ به طوری که کمترین هزینه را برای مجموعه داشته باشد. اگر محصولات چیدمان بهینه‌ای نداشته باشند، به نیروهای بیشتر و در نتیجه هزینه بیشتری برای جمع‌آوری محصولات نیاز است.

هدف این پروژه کاهش این هزینه‌ها با فراهم کردن زیرساخت‌هایی جهت مدیریت راحت‌تر عناصر موجود در انبارها است.

برای پیاده‌سازی پروژه، در انبارهای بزرگ و در بخش‌ها و قفسه‌های مختلف، بوردهای سخت‌افزاری قرار داده می‌شود که بر روی هر برد، نمایشگری در جهت تعامل کاربر فراهم شده است. مسئولین چیدمان محصولات در انبار پس از تغییر جایگاه محصولات، به کمک برنامه‌ای که بر روی این بوردها نصب شده است، جایگاه فعلی محصولات را نیز بروزرسانی می‌کنند. همچنین پس از جمع‌آوری کالاهای سفارش‌های ثبت شده، کارکنان تعداد کالاهایی که از انبار کم شده است را بروزرسانی می‌کنند. این کار توسط برنامه‌ای که برای جمع‌آورندگان کالاهای سفارش‌های داخل انبار نوشته شده است انجام می‌گیرد. در نهایت برنامه‌ای برای مدیران انبار فراهم شده است تا به راحتی بتوانند وضعیت موجود در انبار را مشاهده کنند و در صورت نیاز، اقدام به تغییر عناصر آن بکنند.

۲-۵ کارهای آینده

مشکل اصلی این پروژه، هزینه نسبتاً بالای برد رزبری‌پی و همچنین نمایشگر لمسی در بازار کنونی است. با توجه به پیاده‌سازی تحت وب برنامه‌های سمت کاربر، این پروژه با هر گونه

سخت‌افزاری که توانایی جستجو در صفحات وب را داشته باشد انطباق داده می‌شود. همچنین با توجه به این که تمامی عملیات و تغییرات در میزبانی، جدا از سخت‌افزار در تعامل با کاربر انجام می‌شود، سخت‌افزاری با مشخصات نه چندان زیاد هم پاسخگوی نیازمندی‌های پروژه است. یکی از راه‌های ارزان‌تر برای عملی ساختن پروژه، استفاده از رایانه‌های لوحی^۱ ارزان قیمت و البته با سخت‌افزار نسبتاً ضعیف است.

از دیگر موارد مفید، می‌توان به استفاده از سنسورها و عملگرهای مختلف اشاره کرد. سنسور دما برای خواندن دمای قفسه در انبارهای مواد غذایی و یا محصولات حساس به دمای محیط، سنسور تشخیص دود و یا آتش‌سوزی، سنسور تشخیص میزان رطوبت هوا برای محصولات الکتریکی و دیگر محصولاتی که به رطوبت اطراف حساس هستند، می‌توانند در بهبود کارایی این سخت‌افزارها مفید واقع شوند. همچنین به صورت نرم‌افزاری می‌توان اخطارهایی در سیستم دیده‌بان^۲ تعریف کرد تا در صورت بروز مشکل، از طریق سرویس‌هایی مانند پیامک به مسئول مربوطه اطلاع داده شود.

استفاده از رباتیک برای انبارهای انبوه و بزرگ، چشم‌انداز غایی این پروژه است. ربات‌ها می‌توانند در جمع‌آوری محصولات داخل انبار و فروشگاه‌ها و همچنین چیدن محصولات در بخش‌های مختلف جایگزین انسان شوند. از بارزترین نمونه‌های استفاده از رباتیک در فرایندهای انبارداری، شرکت آمازون است. این شرکت هر سال مسابقاتی را جهت هوشمندسازی انبارداری در زمینه‌های چیدمان و جمع‌آوری محصولات برگزار می‌کند. در سال‌های اخیر در شرکت آمازون ربات‌ها وظیفه جمع‌آوری و همچنین چیدن محصولات را بر عهده دارند[۷].

حوزه هوش مصنوعی و علم داده را نیز می‌توان در این پروژه دخیل کرد. به مرور زمان داده‌های ارزشمندی از جایگاه محصولات و میزان مصرف محصولات در بخش‌های مختلف به دست می‌آید که به مدیر و مسئول چیدمان انبار کمک شایانی خواهد کرد. درنهایت از این داده‌ها می‌توان در پیدا کردن مسیرهای بهینه جمع‌آوری محصولات استفاده کرد.

^۱Tablet^۲Monitoring

منابع و مراجع

- [1] Django official website. <https://www.djangoproject.com/>.
- [2] Django rest framework official website. <https://www.django-rest-framework.org/>.
- [3] Nuxt.js official website. <https://nuxtjs.org/>.
- [4] Postgress official website. <https://www.postgresql.org/>.
- [5] Raspberry pi official website. <https://www.raspberrypi.org/>.
- [6] Vue.js official website. <https://vuejs.org/>.
- [7] Carlos Hernandez, Mukunda Bharatheesha, Wilson Ko Hans Gaiser Jethro Tan Kanter van Deurzen Maarten de Vries Bas Van Mil Jeff van Egmond Ruben Burger Mihai Morariu Jihong Ju Xander Gerrmann Ronald Ensing Jan Van Frankenhuyzen Martijn Wisse. Team delft's robot winner of the amazon picking challenge 2016. *Springer*, 21, 2017.