



به نام خدا

گزارش کارآموزی

سارا سلطانی گردفرامرزی

## فهرست

۳	شرکت لیان گوهران
۴	مرحله مقدماتی یادگیری (کورس مقدماتی)
۴	مرحله متوسط یادگیری
۵	آشنایی مقدماتی با حوزه فرانت اند
۶	استارت پروژه اصلی و بررسی نیازمندی های آن
۶	فاز اول پروژه
۶	احراز هویت کاربران
۶	کار کردن با سشن ها
۷	ایجاد دیتابیس های مربوط به غذا و سفارشات بیرون بر به صورت جداگانه
۷	مشاهده تاریخچه سبد خرید
۷	اتصال به درگاه پرداخت
۷	فاز دوم پروژه
۸	اضافه کردن فرم رزرو و دیتابیس های مربوط به آن
۸	امکان رزرو غذا با میز
۸	امکان کنسل کردن رزروها
۹	فاز سوم پروژه
۹	ایجاد رول های مختلف
۹	ایجاد کارتابل شخصی مربوط به هر رول
۱۱	ایجاد تسک های اتوماتیک
۱۱	وصل کردن برنامه به postgre

## شرکت لیان گوهران

شرکت مهندسی لیان گوهران در دو بخش اتوماسیون اداری و صنعتی فعالیت می کند. در قسمت اتوماسیون صنعتی دارای سیستم های کنترل و مانیتورینگ صنعتی (SCADA)، سیستم اطلاعات تولید، سیستم توزین هوشمند خاک کارخانجات سرامیک و سیمان و همچنین قابلیت راه اندازی و برنامه ریزی انواع ماشین آلات خط تولید می باشد.

در زمینه اتوماسیون اداری تجربه زیادی در برنامه نویسی انواع وب اپلیکیشن ها برای اپلیکیشن های مختلف دارد. این شرکت با جدیدترین و قدرتمندترین فناوری ها از جمله React Js, Vue Js, Php Laravel, Django و ... برای مشتریان خود اپلیکیشن یا وبسایت در هر زمینه ای که دلخواه مشتری باشد تولید می کند و یکی از ویژگی های بارز این نرم افزارها استفاده حداکثری از هوش مصنوعی در نرم افزارها و الگوریتم ها می باشد. همچنین بهینه سازی های لازم در سمت سرور برای ارائه سرعت باورنکردنی در اجرای برنامه ها انجام شده است.

همچنین این شرکت در حوزه سخت افزاری نیز فعالیت می کند. دستگاه ها و روند انجام کارها را در مجموعه های تولیدی بررسی کرده؛ آن ها را بروزرسانی و برنامه نویسی آنها را انجام می دهد. شرکت لیان گوهران به کمک صاحبان صنایع مختلف می رود تا در دوران تحریم دستگاه خریداری شده آنها که فاقد خدمات پس از فروش بوده و یا قطعات آنها بعضاً کم و یا با قیمت گزاف در دسترس هستند را با بهترین قطعات موجود در بازار جایگزین کرده و در این امر از هر دو قدرت سخت افزاری و نرم افزاری (برنامه نویسی) برخوردار است.

کسب و کارهای بزرگ و کوچکی در صنایع مختلف به شرکت اعتماد کرده اند و همکاری بلند مدتی را با این شرکت آغاز نموده اند. شرکت هایی در زمینه تولید کاشی، مواد غذایی، فولاد و ...

## کارهای انجام شده در دوره کارآموزی

در این بخش به بررسی کامل تر فعالیت های مربوط به دوره کارآموزی پرداخته میشود. سه هفته ابتدایی کارآموزی به یادگیری و انجام مینی پروژه های آموزشی در حین گذراندن کورس های آموزشی پرداخته شد.

ابتدا به تحقیق و مقایسه درباره دو فریمورک Django , Flask پرداختم و سپس با توجه به نیاز پروژه فریمورک Django انتخاب شد. (با توجه به امکانات پنل ادمین و انعطاف پذیری بیشتر این فریمورک مناسب تر بنظر رسید.) در مرحله بعد برای راحتتر بودن دیباگ پروژه و مشارکت راحتتر با دیگر اعضای

گروه (اعضای مربوط به بخش فرانت اند پروژه ها) به یادگیری git , github پرداختم. Django فریمورکی به زبان برنامه نویسی پایتون برای توسعه بک اند سایت میباشد. در مرحله بعد پکیج های مورد نیاز جنگو راه اندازی و نصب شدند و در نهایت در محیط pycharm این فریمورک راه اندازی شد و یک virtual environment برای نگهداری پکیج ها ایجاد شد. در مرحله بعدی به یادگیری کلیات این فریمورک پرداختم.

### مرحله مقدماتی یادگیری (کورس مقدماتی):

در مرحله ابتدایی ، یادگیری را با یک کورس مقدماتی شروع کردم. در این کورس با تسک هایی همچون احراز هویت (لاگین و رجیستر) و ایجاد دیتابیس های مورد نیاز هر پروژه و آشنایی با روابط بین دیتابیس های هر پروژه بود. همچنین در این مرحله با دستورات دیتابیس در زبان sqlite (که پیش فرض کار کردن جنگو با دیتابیس است) و ادیت یا حذف اطلاعات از دیتابیس آشنا شدم.

بعد از یادگیری این موارد مقدماتی جهت آزمودن کیفیت مهارت های کسب شده یک مینی پروژه شبیه به سایت GoodReads پیاده سازی شد که امکاناتی همچون احراز هویت و کامنت گذاری کاربران و فیلتر پست های موجود سایت براساس کتگوری یا تگ های مربوطه و همچنین سرچ داخل پست های سایت و pagination صفحات با پست های زیاد را دارا بود.

### مرحله متوسط یادگیری (پیاده سازی یک مینی پروژه شبکه اجتماعی):

ادامه یادگیری این فریمورک با یک کورس متوسط همزمان با پیاده سازی یک مینی پروژه شبکه اجتماعی بود. در این دوره با ویوهای class based و پیاده سازی آن ها ، کاربرد widget ها در فرم ها و ایجاد فرم ها ، ایجاد URL namespace (برای سازماندهی و مدیریت بهتر url های سایت) ، اعتبار سنجی اطلاعات ارسالی توسط فرم ها با استفاده از متد clean ، آشنایی با lazy querysets و کاربر آن ها ، آشنایی با متد dispatch و کاربردهای آن ، محدود کردن سطح دسترسی های کاربران غیر لاگین شده در سایت به برخی url ها با استفاده از login required mixin ها ، آشنایی با authenticate backend های مختلف و کاربرد هر کدام تحت چه شرایطی ، آشنایی با متد get\_absolute\_url و کاربرد آن برای بهبود مشخص کردن url ها در template ها ، آشنایی با متد setup و کاربردهای آن ، نحوه هندل کردن و نمایش بهتر ارور های ۴۰۴ ، نحوه وصل شدن به سرویس دهنده جیمیل برای ارسال لینک بازایی رمز عبور به ایمیل کاربران ، آشنایی با توابع سورت کردن ، آشنایی با روابط backward در

دیتابیس و استفاده از فیلد `related names` در دیتابیس ، آشنایی با پارامتر `next` و کاربرد آن در `url` ها ، آشنایی با `method decorator` ها و کاربرد آن در ویوهای `function based` ، آشنایی با نحوه اکستند کردن مدل یوزر جنگو ، آشنایی با سیگنال ها و کاربرد آن در جنگو.

در انتهای این دوره با استفاده از تمامی مهارت های آموخته شده ذکر شده در قسمت بالا یک مینی پروژه مشابه شبکه اجتماعی پیاده سازی کردم که شامل امکاناتی از قبیل پروفایل شخصی کاربران و امکان ادیت و آپدیت آن توسط هر کاربر ، نمایش پست ها در قالب عکس با عنوان و تاریخ انتشار ، مشاهده جزئیات مربوط به هر پست با هدایت به صفحه مربوط به هر پست به صورت جداگانه ، امکان حذف و آپدیت یا ادیت هر پست توسط کاربر مورد نظرش ، امکان ایجاد و اشتراک گذاری پست توسط کاربران عضو و لاگین شده (در این جهت لازم بود برای مدیریت فایل های استاتیک از قبیل عکس های آپلود شده تنظیماتی برای ذخیره فایل های استاتیک انجام گیرد)، امکان بازیابی رمز عبور و سپس `confirm` کردن رمز جدید ایجاد شده ، امکان کامنت گذاری و نمایش کامنت های مرتبط با هر پست ، امکان لایک کردن هر پست توسط کاربران دیگر ، امکان ریپلای کردن کامنت ها و پست ها توسط کاربران ، امکان فالو کردن کاربران دیگر ، و در نهایت اکستند کردن مدل یوزر پیشفرض جنگو و ایجاد تغییراتی در مدل پیشفرض (لاگین کردن کاربران توسط ایمیل).

از تمامی امکانات ذکر شده در قسمت بالا برای بهبود کد و اصولی و استاندارد بودن متد ها در این مینی پروژه استفاده شد.

### آشنایی مقدماتی با حوزه فرانت اند:

در ادامه کار برای تست قسمت های توسعه داده شده از مینی پروژه ها و مشاهده خروجی کار ، آشنایی اندکی با حوزه توسعه فرانت اند لازم بود که در این راستا اندکی با زبان های `html` , `css` , `bootstrap` آشنا شده و یک طراحی جزئی برای `UI` مینی پروژه ها ایجاد شد. از طرف یگر برای وصل کردن بک اند سایت به فرانت اند (با فریمورک `react` ) تحقیق هایی انجام شده تا در نهایت برای وصل شدن پروژه اصلی به فرانت اند آن مورد استفاده قرار گیرد. ( در این جهت کمی آشنایی با `API` رست در جنگو لازم بود که کمی مورد بررسی و تحقیق قرار گرفت. )

## استارت پروژه اصلی و بررسی نیازمندی های آن:

در ادامه کار پروژه اصلی تعریف شده و نیازمندی های آن بررسی و جمع آوری شده و تسک های مربوط به آن و تحویل هر فاز به صورت مرحله به مرحله مشخص و زمان بندی آن تعیین شده است. پروژه اصلی توسعه یک وبسایت رستورانی و در عین حال فروشگاه (مشابه اسنپ فود) میباشد.

## فاز اول پروژه:

در این فاز بخش فروشگاه وبسایت پیاده سازی شد ( که سفارش غذا مشابه اسنپ فود انجام میگیرد). در این فاز کاربر باید میتواند به منوی غذا دسترسی داشته و به تعداد محدودی پرس از هر غذا به سبد خرید خود اضافه کند. تا در نهایت توسط پیک به دست شخص برسد.

## احراز هویت کاربران:

با توجه به اینکه در مینی پروژه های آموزشی نیز احراز هویت کاربران به شکل های مختلفی انجام شد در این قسمت چالش زیادی نداشتیم و خیلی نیاز به تغییرات اساسی نداشت. در این وبسایت رجیستر کردن کاربران توسط شماره تلفن همراه انجام شده سپس یک کد به شماره تلفن فرد پیامک شده و در صورت صحت کد وارد شده توسط کاربر و همچنین منحصر به فرد بودن شماره تلفن و همچنین ایمیل وارد شده توسط فرد رجیستر کردن فرد صورت میگیرد. در این مرحله اس ام اس شدن کد رندوم ساخته شده به شماره تلفن وارد شده پیاده سازی و بررسی شد.

لاگین کردن فرد نیز با استفاده از تلفن همراه و پسورد فرد صورت میگیرد. بازیابی رمز عبور نیز در مینی پروژه های آموزشی پیاده سازی شده بود که در این پروژه نیز مورد استفاده قرار گرفت. در نهایت کاربر میتواند با لاگ اوت کردن از اکانت خود خارج شود.

## کار کردن با سشن ها:

برای اینکه کاربر بتواند بدون لاگین کردن آیتم ها را به سبد خرید خود اضافه کند باید با **سشن** ها کار میکردم که اطلاعات سبد خرید فرد در سشن مرورگر ذخیره شود. هرگاه کاربر قصد داشت پرداخت خود را

نهایی کند و هزینه آن را بپردازد باید حتماً لاگین صورت می‌گرفت و بعد از لاگین شدن اطلاعات از قبل ذخیره شده در سشن مورد نظر، مورد استفاده قرار می‌گرفت.

### ایجاد دیتابیس های مربوط به غذا ها و سفارشات بیرون بر به صورت جداگانه:

هر کدام از این دیتابیس ها به صورت جداگانه ایجاد شده و از آنجایی که هر سطر از دیتابیس مربوط به سفارشات مربوط به یک نوع غذا میباشد پس دیتابیس غذا باید کلید خارجی این دیتابیس باشد و از طرف دیگر چون هر رکورد از سفارشات بیرون بر مربوط به یکی از یوزرهای سایت است از دیتابیس یوزر نیز یک کلید خارجی خواهیم داشت. آیدی هر رکورد از دیتابیس سفارشات به عنوان شماره سفارش و برای پیگیری محصولات استفاده میشود.

### مشاهده تاریخچه سبد خرید:

برای هر کاربر لاگین شده یک صفحه جهت نمایش محصولات موجود در سبد خرید که نهایی شدند و محصولاتی که هنوز نهایی و سفارش داده نشده اند وجود دارد.

### اتصال به درگاه پرداخت:

برای نهایی کردن سفارش و پرداخت هزینه توسط کاربر جهت هدایت کاربر به درگاه پرداخت زرین پال تنظیمات لازم صورت گرفته و این امکان راه اندازی شد. برای پرداخت آسانر دسترسی به کیف پول و شارژ کیف پول نیز برای هر کاربر فراهم شد.

### فاز دوم پروژه:

در این بخش امکانات رزرو میز (همراه با غذا) پیاده سازی شد. در این بخش کاربر میتواند با مشخص کردن اطلاعاتی همچون تاریخ شروع و زمان شروع و پایان و همچنین تعداد نفرات انتخابی میزهای موجود و در دسترس را مشاهده کرده و در نهایت یک یا چندین میز را برای رزرو انتخاب کند.

### اضافه کردن رزرو و دیتابیس های مربوط به آن:

در این بخش دیتابیس های مربوط به ذخیره رزرو ها و همچنین میزهای موجود ایجاد شده. برای هر میز متناسب با تعداد صندلی های آن یک بیعانه ی مشخصی ذخیره شده که توسط ادمین پیچ قابل تغییر است. فرم رزرو نیز شامل فیلدهای تاریخ شروع و زمان شروع و پایان استفاده از میز و تعداد نفرات رزرو است. اگر کاربر تاریخی وارد نکند به صورت دیفالت تاریخ همان روز و تاریخ شروع نیز به صورت دیفالت یک ساعت بعد از زمان فعلی و زمان پایان دو ساعت بعد از زمان فعلی در نظر گرفته شده و روی این اطلاعات فیلتر انجام میشود. فلیتر انجام شده و میزهای در دسترس به کاربر نشان داده شده. در این حالت کاربر میتواند چندین میز را انتخاب کند. البته کاربر فقط میتواند زمانهای حداقل یکساعت بعد از زمان فعلی را برای زمان شروع انتخاب کند.

### امکان رزرو غذا با میز:

کاربر میتواند هنگام انتخاب میز گزینه رزرو غذا را نیز انتخاب کند. که در این حالت وارد صفحه منوی رستوران شده و غذاهای انتخابی و تعداد پرس مورد نیاز هر کدام ثبت شده و در نهایت برای نهایی کردن پرداخت به درگاه پرداخت متصل شده و هزینه غذای سفارش داده شده و همچنین بیعانه میز انتخابی را پرداخت میکند.

### امکان کنسل کردن رزرو ها:

آپشن مشاهده تاریخچه رزرو نیز برای کاربر وجود دارد که کاربر میتواند میزها و غذاهای رزروی را از ابتدا تا کنون مشاهده کند. اما صرفا برای رزروهایی که دوساعت تا زمان شروعشان باقی مانده است امکان کنسل کردن وجود دارد و گزینه کنسل نمایش داده میشود. با قوانینی که در ادامه شرح داده شده هزینه محاسبه شده و در طی ۲۴ ساعت آینده به شماره حساب فرد که از کاربر گرفته شده واریز میشود. اگر رزرو بدون غذا باشد و روزهای قبل از تاریخ رزرو ، کنسل شود ۲۰٪ بیعانه برمیگردد و اگر همان روز ۲ ساعت قبلش کنسل کند اصلا بیعانه برنمیگردد. اما اگر رزرو با غذا باشد و روزهای قبل کنسل انجام شود ۵۰٪ هزینه غذا برگشت خورده و ۲۰٪ بیعانه میز برمیگردد اما اگر همان روز تا ۲ ساعت قبلش کنسل کند ۲۰٪ هزینه غذا صرفا برگشت داده میشود.



## فاز سوم پروژه:

در این فاز رول های مختلف رستوران اضافه شده (گارسن و مشتری و آشپز و منشی و ادمین) و سطح دسترسی ها و کارتابل مشخص و جداگانه هر کدام پیاده سازی شده. در ادامه نیز پنل ادمین و امکانات مربوط به آن کاستومایز شده و امکانات و آپشن های بیشتری به آن اضافه شد. سپس تسک هایی که باید به صورت اتوماتیک و در بازه های زمانی مشخص اجرا شوند هندل شده و راه اندازی شدند و اینتروال های زمانی ایجاد شدند. و در نهایت برنامه به PostgreSQL وصل شده و از پیشفرض SQLite خارج شد.

## ایجاد رول های مختلف:

در این مرحله از امکانات PERMISSION ها در جنگو استفاده کرده و رول های مختلف ایجاد شدند. ۴ نوع PERMISSION ایجاد شده که هنگام ایجاد کاربران در دیتابیس این سطح دسترسی ها به کاربران انتساب داده شد.

برای لاگین کردن و وارد شدن به کارتابل شخصی هر فرد نیز از دکوریتهور های مربوط به PERMISSION استفاده کرده تا فرد بتواند به صفحه شخصی خود وارد شود نه صفحه شخصی رول های دیگر. در دیتابیس مربوط به ۴ Groups گروه مختلف افراد را ایجاد کرده و permission های ایجاد شده مربوط به هر کدام را به آن نسبت دادیم.

## ایجاد کارتابل شخصی مربوط به هر رول:

- **گارسن:** ابتدا کارتابل شخصی مربوط به گارسن ها را ایجاد کردم. گارسن ها امکان دسترسی به منو و مشاهده آن و سفارش غذا داشتند. از طرف دیگر میتوانند میزهای رزرو شده تاریخ همان روز را مشاهده کنند. جلوی هر رکورد رزرو گزینه "Order" جهت سفارش غذا برای میزی که مشتریانش وارد رستوران شدند (ستون Is\_arrived برای آن رکورد فعال شده است از قبل توسط گارسن یا منشی) و همچنین گزینه "arrived" برای رزروهایی که تایم شرویشان فرا رسیده است فعال میباشد. (گارسن میتواند با کلیک روی این گزینه مشخص کند که مشتریان سر میز خود مستقر شده و وارد رستوران شدند). همچنین از طرف دیگر امکان سرچ کردن داخل جدول نشان داده شده نیز وجود دارد که با شماره تلفن همراه مشتری سرچ انجام میشود تا در صورت شلوغی رستوران و تعداد رزروهای زیاد گارسن سریعتر میز رزرو شده را انتخاب کند.

از طرف دیگر گارسن میتواند به لیست سفارشات مربوط به هر میز نیز دسترسی داشته باشد و در یک صفحه جداگانه سفارشات که برای میزهای رزرو شده گرفته شده نمایش داده شده و از طرف دیگر سفارشات غذایی که از قبل با میز رزرو شده بود نمایش داده میشود. جلوی هر رکورد از این جداول گزینه "delivered" نیز وجود دارد که گارسن با کلیک روی این گزینه مشخص میکند که غذاهای مربوط به میز مشخص و رزرو شده تحویل میز و مشتری داده شده است و گزینه Is\_delivered این رکورد نیز فعال میشود.

امکان دیگری که برای گارسن پیاده سازی شده است اضافه کردن یک اردر جدید است که برای مشتریانی است که رزروی نداشته و وارد رستوران شده اند. در این حالت گارسن شماره تلفن و شماره میز مورد نظر مشتری را وارد کرده و وارد صفحه منوی رستوران شده و سفارشات مشتری را وارد کرده و با نهایی شده این سفارشات در دیتابیس مورد نظر ذخیره میشود.

- **آشپز:** در این مرحله کارتابل شخصی آشپز نیز پیاده سازی شد. آشپز نیز همانند گارسن به منوی رستوران دسترسی داشته و میتواند در صورت تمام شدن هر غذا آن غذا را از دسترس خارج کند تا گارسن یا مشتریان نتوانند آن را سفارش دهند. برای هر غذا در منو این گزینه وجود دارد.  
از طرف دیگر آشپز امکان مشاهده سفارشات داده شده در همان روز را نیز دارد. که این سفارشات با اولویت و رنگ مشخص به آشپز نشان داده میشود. ابتدا سفارشات غذایی که با رزرو میز بوده نیم ساعت قبل از زمان شروع رزرو به آشپز نشان داده میشود. بعد از آن سفارشات که از میز رزرو شده گرفته شده بود نمایش داده شده و بعد از آن سفارشات داخل رستوران که رزرو نشده بودند نمایش داده میشود. بعد از این موارد سفارشات بیرون بر به آشپز با پایین ترین اولویت نشان داده میشود. جلوی هر رکورد از این جداول گزینه "Is\_finished" نیز وجود دارد که آشپز با کلیک کردن روی این گزینه به گارسن اعلام کرده که پخت غذا تمام شده و آماده تحویل است و سپس این سفارش در کارتابل گارسن در قسمت اردرها نمایش داده میشود و در کارتابل خود آشپز حذف میشود.

- **منشی:** در این مرحله کارتابل شخصی منشی نیز پیاده سازی شده است. منشی امکاناتی همچون دسترسی به منو برای ثبت سفارشات بیرون بر داشته. از طرف دیگر امکان اضافه کردن سفارشات بیرون بر داخل رستوران داشته که از مشتریان دریافت میکند. همچنین همچون گارسن امکان دسترسی به میزهای رزرو شده همان روز و گزینه "arrive" نیز

دارد. از طرف دیگر امکان مشاهده لیست تمامی سفارشات بیرون بر نیز دارد. از طرف دیگر سفارشات که توسط گارسن به میز مورد نظر تحویل داده شده اند به کارتابل شخصی منشی در قسمت سفارشات تحویل داده شده فرستاده شده تا موقع ترک مشتری پرداخت هزینه یا برگرداندن بیعانه توسط منشی انجام شود. و گزینه **paid** نیز روبه روی هر رکورد از این جدول وجود دارد تا در صورت انتخاب این گزینه توسط منشی مشخص شود که سفارشات میز مورد نظر پرداخت هزینه شده است.

**ایجاد تسک های اتوماتیک:** در این مرحله تسک هایی که باید به صورت اتوماتیک در پس زمینه برنامه در حال اجرا در بازه های زمانی مشخص اجرا شوند ایجاد و راه اندازی شدند. Celery beat ها نصب و راه اندازی شده و با استفاده از آنها **interval** های زمانی ۲ دقیقه (برای حذف کدهای ۴ رقمی ایجاد شده در دیتابیس برای ارسال به شماره همراه موقع رجیستر کردن) و همچنین **interval** های زمانی ۳۰ دقیقه (برای کنسل کردن اتوماتیک رزرو هایی که نیم ساعت از زمان شروعشان گذشته و مشتری هنوز وارد رستوران نشده است). ایجاد شد. در ادامه توابع مربوط به تسک ها نوشته شده و تسک های ایجاد شده در قسمت **period task** ها به **interval** زمانی مشخص شده نسبت داده و تعریف شد. در نهایت با نصب ابزار **rabbit mq** (نوعی **message broker**) و ورکهای برای اجرای **celery beat** ها در ترمینال ایجاد و اجرا شده تا تسک های تعریف شده را در بازه های زمانی مشخص شده اجرا کنند. در ادامه برای اینکه لازم نباشد از ترمینال برای اجرای ورکها استفاده شود لازم بود برنامه به پس زمینه منتقل شود که به صورت خودکار اجرا شود و لازم به وارد کردن کامندهای مربوطه در ترمینال نباشد که این کار با **deaminization** انجام گرفت.

**وصل کردن برنامه به postgre:** با توجه به اینکه **sqlite** فقط سایتهای کوچک با کاربران محدود را میتواند ساپورت کند مجبور شدیم از پیشفرض جنگو خارج شده و برنامه را به **postgre** وصل کنیم.