

بسم الله الرحمن الرحيم



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

داک نهایی پروژه مهندسی نرم افزار

درس:

مهندسی نرم افزار

استاد:

دکتر فاطمی

اعضای گروه:

امین کیانی، یزدان افرا، آرمان خلیلی، متین عصب الظهور، محمد مهدی کتابچی، محمد جعفری،

علی حسینی ف

فهرست

لیست نهایی نیازمندی‌ها:	۳
نیازمندی‌های اصلی:	۳
نیازمندی‌های اختیاری:	۴
تغییر نیازمندی‌ها:	۴
نیازمندی‌های پیاده‌سازی شده:	۴
نمودار معماری سیستم	۵
سند راهنمای کاربر	۶
استاد	۶
دستیار درس	۸
گزارش نهایی	۱۰
دستورالعمل نصب و راه‌اندازی سیستم	۱۰
تکنولوژی‌ها و ابزارهای استفاده شده به صورت جزئی	۱۵
معیارهای کیفی سیستم و اقدامات انجام شده جهت ارزیابی آنها	۱۵
پیشنهادات ممکن برای بهبود و توسعه سیستم در آینده	۱۶
نقش هریک از اعضای گروه در پروژه به صورت جزئی	۱۹
محمدامین کیانی	۱۹
یزدان افرا	۱۹
آرمان خلیلی	۱۹
متین عصب‌الظهور	۲۰
علی حسینی‌فرد	۲۰
محمد مهدی کتابچی	۲۱
محمد جعفری	۲۱
توضیح فایل‌های اصلی پروژه	۲۲
• بک‌اند	۲۲
• فرانت‌اند	۲۲

لیست نهایی نیازمندی‌ها:

نیازمندی‌های اصلی:

- R1. سیستم باید امکان ثبت نام و عضویت را به کاربران بدهد.
- R2. سیستم باید امکان ورود به حساب کاربری و خروج از آن را به کاربران بدهد.
- R3. سیستم باید امکان ویرایش پروفایل را به کاربران بدهد.
- R4. سیستم باید امکان مشاهده لیست دانشجویان توسط استاد را بدهد.
- R5. سیستم باید امکان جست‌وجو میان آگهی‌ها را به اساتید بدهد.
- R6. سیستم باید امکان مشاهده درس‌هایی که استاد ثبت کرده است را به او بدهد.
- R7. سیستم باید امکان نوشتن توضیحات استاد برای آگهی‌ها را داشته باشد.
- R8. دستیاران باید توانایی آپلود رزومه را داشته باشند.
- R9. سیستم باید طوری پیاده سازی شود که هر دانشجو آگهی درس‌ها برای دستیار شدن را ببیند.
- R10. سیستم باید امکان درخواست دادن به استاد برای دستیار شدن را داشته باشد.
- R11. سیستم باید امکان حذف درس را به اساتید بدهد.
- R12. استاد باید بتواند برای دستیاران حداقل معدل تعیین کند.
- R13. کاربران استاد باید توانایی بارگزاری آگهی نیازمندی برای دستیار درسی را داشته باشند.
- R14. سیستم باید قابلیت جست‌وجوی آگهی برای دستیاران را داشته باشد.
- R15. استادان باید بتوانند درخواست دستیاران را رد یا تایید کنند.
- R16. دستیاران باید بتوانند نتیجه درخواست خود را مشاهده کنند.

نیازمندی‌های اختیاری:

- R1. سیستم باید طراحی واکنش‌گرا (Responsive) داشته باشد.
- R2. دستیار باید امکان حذف درس را داشته باشد.
- R3. سیستم باید امکان امتیازدهی دستیاران را به استادان بدهد.
- R4. سیستم باید زمان ارسال آگهی‌ها را مشخص کند.
- R5. استاد پس از پایان ترم بتواند برای استادیاران خود مدرک صادر کند و یک گواهی رسمی مبنی بر فعالیت دانشجو به عنوان استادیار صادر بشود که مهر و امضای استاد و دانشگاه بر آن اعمال شده باشد.
- R6. استاد باید امکان حذف دستیار را داشته باشد.
- R7. سیستم باید مانع دسترسی کاربران به صفحات دیگر کاربران شود.
- R8. سیستم باید روی تعداد درسی که هر نفر میتواند تی‌ای آن دروس شود، نظارت داشته باشد.

تغییر نیازمندی‌ها:

- اضافه شدن R16: به دلیل بالا بردن کارایی سیستم و بهبود تجربه کاربری کاربران دستیار.
- اضافه شدن نیازمندی‌های اختیاری: نیازمندی R2 تا R8.
- انتقال R8 به نیازمندی‌های اختیاری به علت پایین‌تر بودن اولویت آن نسبت به نیازمندی‌های اساسی سیستم.

نیازمندی‌های پیاده‌سازی شده:

تکمیل و بهبود تمامی نیازمندی‌های اصلی سیستم.

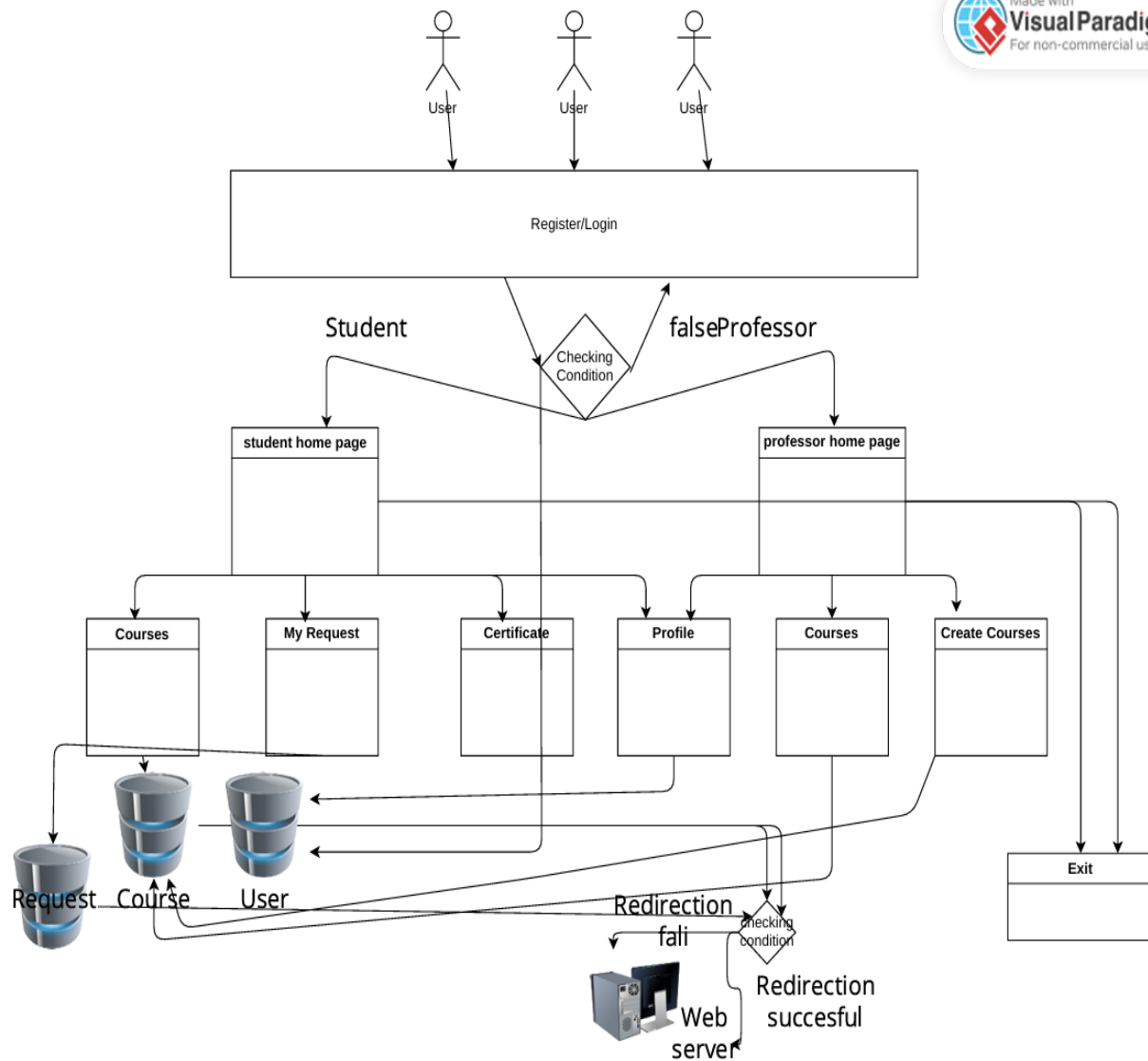
پیاده‌سازی R3 امتیازی

پیاده‌سازی R5 امتیازی

پیاده‌سازی R7 امتیازی

میزان نیازمندی‌های پیاده‌سازی شده: ۱۲۰٪

نمودار معماری سیستم



سند راهنمای کاربر

استاد:

۱. ثبت نام و عضویت

- برای عضویت در سیستم، به صفحه ثبت نام مراجعه کنید.
- اطلاعات خواسته شده شامل نام، ایمیل، رمز عبور و مشخصات تکمیلی را وارد کنید.
- پس از ثبت نام، یک ایمیل تأییدیه برای شما ارسال خواهد شد. با کلیک روی لینک موجود در ایمیل، عضویت شما تکمیل می‌شود.

۲. ورود و خروج از حساب کاربری

- برای ورود به حساب کاربری خود، به صفحه ورود بروید.
- ایمیل و رمز عبور خود را وارد کرده و روی دکمه "ورود" کلیک کنید.
- برای خروج از حساب کاربری، روی نام کاربری خود در بالای صفحه کلیک کرده و گزینه "خروج" را انتخاب کنید.

۳. ویرایش پروفایل

- برای ویرایش پروفایل، به بخش "پروفایل من" در منوی اصلی بروید.
- اطلاعات پروفایل خود را ویرایش کرده و تغییرات را ذخیره کنید.

۴. مشاهده لیست دانشجویان

- به بخش "دانشجویان" در منوی اصلی بروید.
- لیستی از تمامی دانشجویان مرتبط با درس‌های شما نمایش داده می‌شود.

۵. جست‌وجو میان آگهی‌ها

- به بخش "آگهی‌ها" در منوی اصلی بروید.
- از قسمت جست‌وجو برای پیدا کردن آگهی‌های مرتبط با نیازمندی‌های خود استفاده کنید.

۶. مشاهده درس‌های ثبت شده

- به بخش "درس‌های من" در منوی اصلی بروید.
- لیست تمامی درس‌هایی که شما ثبت کرده‌اید، نمایش داده می‌شود.

۷. نوشتن توضیحات برای آگهی‌ها

- در بخش "آگهی‌ها"، پس از انتخاب آگهی مورد نظر، گزینه "ویرایش" را انتخاب کنید.
- توضیحات خود را در کادر مربوطه وارد کرده و تغییرات را ذخیره کنید.

۸. حذف درس

- به بخش "درس‌های من" بروید.
- درس مورد نظر را پیدا کرده و روی گزینه "حذف" کلیک کنید.
- در پنجره تأیید، گزینه "بله" را انتخاب کنید.

۹. تعیین حداقل معدل برای دستیاران داوطلب

- هنگام ایجاد آگهی جدید یا ویرایش آگهی موجود، می‌توانید حداقل معدل مورد نظر را در قسمت مشخص شده وارد کنید.

۱۰. بارگزاری آگهی نیازمندی برای دستیار درسی

- به بخش "آگهی‌ها" در منوی اصلی بروید.
- روی دکمه "ایجاد آگهی جدید" کلیک کنید.
- فرم مربوطه را پر کرده و اطلاعات مورد نیاز را وارد کنید.
- پس از تکمیل فرم، روی دکمه "ثبت" کلیک کنید.

۱۱. رد یا تایید درخواست دستیاران

- به بخش "درخواست‌ها" در منوی اصلی بروید.
- لیست درخواست‌های دستیاران را مشاهده کنید.

- با کلیک بر روی هر درخواست، گزینه‌های "رد" یا "تایید" در دسترس خواهند بود.

۱۲. مشاهده نتیجه درخواست دستیاران

- پس از بررسی درخواست‌ها، دستیاران می‌توانند نتیجه درخواست خود را از طریق بخش "درخواست‌های من" مشاهده کنند.

دستیار درس

۱. ثبت نام و عضویت

- برای عضویت در سیستم، به صفحه ثبت نام مراجعه کنید.
- اطلاعات خواسته شده شامل نام، ایمیل، رمز عبور و مشخصات تکمیلی را وارد کنید.
- پس از ثبت نام، یک ایمیل تأییدیه برای شما ارسال خواهد شد. با کلیک روی لینک موجود در ایمیل، عضویت شما تکمیل می‌شود.

۲. ورود و خروج از حساب کاربری

- برای ورود به حساب کاربری خود، به صفحه ورود بروید.
- ایمیل و رمز عبور خود را وارد کرده و روی دکمه "ورود" کلیک کنید.
- برای خروج از حساب کاربری، روی نام کاربری خود در بالای صفحه کلیک کرده و گزینه "خروج" را انتخاب کنید.

۳. ویرایش پروفایل

- برای ویرایش پروفایل، به بخش "پروفایل من" در منوی اصلی بروید.
- اطلاعات پروفایل خود را ویرایش کرده و تغییرات را ذخیره کنید.

۴. آپلود رزومه

- به بخش "پروفایل من" بروید.
- گزینه "آپلود رزومه" را انتخاب کنید.
- فایل رزومه خود را انتخاب کرده و بارگذاری کنید.

۵. مشاهده آگهی‌های دستیار شدن

- به بخش "آگهی‌ها" در منوی اصلی بروید.
- لیست تمامی آگهی‌های موجود برای دستیار شدن را مشاهده کنید.

۶. درخواست دادن برای دستیار شدن

- در بخش "آگهی‌ها"، آگهی مورد نظر خود را انتخاب کنید.
- روی گزینه "درخواست برای دستیار شدن" کلیک کنید.
- فرم مربوطه را پر کرده و درخواست خود را ارسال کنید.

۷. مشاهده نتیجه درخواست

- به بخش "درخواست‌های من" در منوی اصلی بروید.
- لیست تمامی درخواست‌های ارسال شده توسط شما نمایش داده می‌شود.
- وضعیت هر درخواست (تایید یا رد شده) در این بخش قابل مشاهده است.

۸. جست‌وجوی آگهی‌ها

- به بخش "آگهی‌ها" در منوی اصلی بروید.
- از قسمت جست‌وجو برای پیدا کردن آگهی‌های مرتبط با نیازمندی‌های خود استفاده کنید.

گزارش نهایی

دستورالعمل نصب و راه اندازی سیستم

پیش نیازها

Docker: اطمینان حاصل کنید که Docker و Docker Compose بر روی سرور شما نصب شده باشد. اگر نصب نشده است، از دستورات زیر استفاده کنید:

نصب: Docker

```
sudo apt update
sudo apt install docker.io
sudo systemctl start docker
sudo systemctl enable docker
```

نصب: Docker Compose

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose sudo
chmod +x /usr/local/bin/docker-compose docker-compose --version
```

مراحل نصب و راه اندازی

۱. کلون کردن پروژه

دریافت کد منبع پروژه از مخزن Git

```
git clone https://github.com/your-repository/uisama.git
cd uisama
```

۲. ایجاد فایل های Docker

Dockerfile برای بک اند (Django) یک فایل با نام Dockerfile در دایرکتوری backend ایجاد کرده و محتوای زیر را در آن قرار دهید:

FROM python:3.9

ENV PYTHONUNBUFFERED 1

WORKDIR /app

COPY requirements.txt /app/

RUN pip install --no-cache-dir -r requirements.txt

COPY . /app/

CMD ["gunicorn", "--bind", "0.0.0.0:8000", "backend.wsgi:application"]

Dockerfile برای فرانت اند (**React**) یک فایل با نام Dockerfile در دایرکتوری frontend ایجاد کرده و محتوای زیر را در آن قرار دهید:

FROM node:14

WORKDIR /app

COPY package.json /app/

RUN npm install

COPY . /app/

RUN npm run build

CMD ["npx", "serve", "-s", "build"]

فایل **Docker Compose**: یک فایل با نام docker-compose.yml در ریشه پروژه ایجاد کرده و محتوای زیر را در آن قرار دهید:

version: '3'

services:

db:

image: postgres:13

volumes:

- postgres_data:/var/lib/postgresql/data

environment:

POSTGRES_DB: uisama_db

POSTGRES_USER: uisama_user

POSTGRES_PASSWORD: your_password

backend:

build:

context: ./backend

volumes:

- ./backend:/app

environment:

- DB_NAME=uisama_db

```
- DB_USER=uisama_user
- DB_PASSWORD=your_password
- DB_HOST=db
- DB_PORT=5432
depends_on:
  - db
ports:
  - "8000:8000"
```

```
frontend:
  build:
context: ./frontend
volumes:
  - ./frontend:/app
ports:
  - "3000:5000"
```

```
volumes:
  :postgres_data
```

۳. تنظیمات پایگاه داده در Django

تنظیمات فایل کانفیگ Django: در فایل `backend/settings.py` مقادیر زیر را ویرایش کنید:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
```

```
'NAME': os.environ.get('DB_NAME'),  
'USER': os.environ.get('DB_USER'),  
'PASSWORD': os.environ.get('DB_PASSWORD'),  
'HOST': os.environ.get('DB_HOST'),  
'PORT': os.environ.get('DB_PORT'),  
}  
}
```

۴. اجرای Docker Compose

ساخت و اجرای کانتینرها:

docker-compose up –build

۵. مهاجرت و ایجاد جداول پایگاه داده

اجرای مهاجرتها:

docker-compose exec backend python manage.py migrate

۶. دسترسی به سیستم

- دسترسی به فرانت‌اند: مرورگر خود را باز کرده و به آدرس <http://localhost:3000> بروید.
- دسترسی به بک‌اند (API): مرورگر خود را باز کرده و به آدرس <http://localhost:8000> بروید.

تکنولوژی‌ها و ابزارهای استفاده شده به صورت جزئی

• فرانت‌اند

- طراحی *UI/UX* با *Adobe XD*
- پیاده‌سازی و طراحی *API* صفحات با کتابخانه *ReactJS*
- استایل صفحات با استفاده از *Tailwind CSS*

• بک‌اند

- ساخت مدل و کنترل کردن دیتابیس با فریم‌ورک *Django*
- ساخت *API* با *Rest API*

معیارهای کیفی سیستم و اقدامات انجام شده جهت ارزیابی آنها

امنیت:

معیار: تامین امنیت کاربران هنگام ورود و خروج به حساب کاربری

اقدامات انجام شده:

- استفاده از توکن برای ارسال و دریافت داده‌ها
- استفاده از پروتکل‌های امنیتی مانند *HTTPS* برای ارتباطات

معیار: تامین امنیت صفحات سیستم

اقدامات انجام شده:

- استفاده از گارد برای صفحات

کارایی:

معیار: بالا بودن سرعت گردش کارها برای کاربران

اقدامات انجام شده:

- استفاده از الگوریتم‌ها و تکنولوژی‌های متناسب با نیاز کاربران
- استفاده از سرورهای پر قدرت متناسب با نیاز
- انجام *Unit Test* برای آزمودن صفحات

قابلیت استفاده

سادگی و کاربرپسند بودن رابط کاربری (User Interface)

دسترسی آسان به قابلیت‌های مختلف سیستم

اقدامات انجام شده:

- انجام تست‌های کاربرپذیری (Usability Testing) با کاربران مختلف و دریافت بازخورد
- طراحی رابط کاربری ساده و قابل فهم برای تمامی کاربران
- ارائه راهنمای کاربری جامع و کاربردی

نگهداری

سادگی در انجام تغییرات و به‌روزرسانی‌ها

مستندسازی جامع و قابل فهم

اقدامات انجام شده:

- نوشتن کدهای تمیز و مستندسازی شده
- استفاده از سیستم‌های مدیریت نسخه (Version Control Systems) مانند Git
- ارائه مستندات کامل و جامع برای توسعه‌دهندگان و کاربران

پیشنهادهای ممکن برای بهبود و توسعه سیستم در آینده

۱. افزایش تعاملات کاربران

- پیاده‌سازی سیستم پیام‌رسانی داخلی: ایجاد امکان ارسال و دریافت پیام بین کاربران (اساتید و تی‌ای‌ها) برای هماهنگی بهتر و افزایش تعاملات.
- سیستم اطلاع‌رسانی پیشرفته: اضافه کردن قابلیت ارسال اعلان‌ها (notifications) برای آگاه‌سازی کاربران از تغییرات و اتفاقات مهم، مانند تایید یا رد درخواست‌ها.

۲. بهبود رابط کاربری و تجربه کاربری (UI/UX)

- طراحی مجدد رابط کاربری: بازبینی و بهبود طراحی رابط کاربری بر اساس بازخوردهای کاربران برای افزایش سادگی و کاربرپسندی
- آموزش تعاملی و راهنماهای ویدیویی: ایجاد راهنماهای ویدیویی و آموزش‌های تعاملی برای کاربران جدید جهت آشنایی سریع‌تر و بهتر با سیستم

۳. افزایش قابلیت‌های جست‌وجو و فیلتر

- جست‌وجوی پیشرفته: بهبود قابلیت جست‌وجو و اضافه کردن فیلترهای متنوع برای پیدا کردن آگهی‌ها و درخواست‌ها به صورت دقیق‌تر
- برچسب‌گذاری (Tagging): اضافه کردن سیستم برچسب‌گذاری برای آگهی‌ها و پروفایل‌ها به منظور دسته‌بندی و جست‌وجوی سریع‌تر

۴. افزایش امنیت سیستم

- احراز هویت دو مرحله‌ای (2FA): اضافه کردن احراز هویت دو مرحله‌ای برای افزایش امنیت حساب‌های کاربری
- رمزنگاری داده‌ها: پیاده‌سازی رمزنگاری برای داده‌های حساس به منظور حفاظت بیشتر از اطلاعات کاربران

۵. توسعه ویژگی‌های تحلیل و گزارش‌دهی

- گزارش‌های پیشرفته: ارائه گزارش‌های پیشرفته و داشبوردهای تحلیلی برای اساتید و مدیران برای بررسی عملکرد تی‌ای‌ها و دانشجویان
- تحلیل داده‌ها: استفاده از ابزارهای تحلیل داده برای شناسایی الگوها و ارائه پیشنهادات به کاربران بر اساس رفتارهای گذشته

۶. افزایش قابلیت‌های همکاری و گروه‌بندی

- پروژه‌های گروهی: ایجاد امکان تشکیل گروه‌های کاری و پروژه‌های گروهی بین تی‌ای‌ها و دانشجویان

- تقویم و زمان‌بندی مشترک: پیاده‌سازی تقویم مشترک برای هماهنگی بهتر در زمان‌بندی کلاس‌ها، جلسات و پروژه‌ها

۷. بهبود فرآیندهای خودکار

- اتوماسیون فرآیندها: بهبود و افزایش اتوماسیون فرآیندهای درخواست و تایید دستیاران برای کاهش زمان و خطاهای انسانی
- یادآوری‌های خودکار: ایجاد سیستم یادآوری خودکار برای رویدادهای مهم و مهلت‌های تعیین شده

۸. افزایش قابلیت‌های سفارشی‌سازی

- پروفایل‌های سفارشی: ایجاد امکان سفارشی‌سازی پروفایل‌ها بر اساس نیازهای خاص کاربران
- تنظیمات پیشرفته: ارائه تنظیمات پیشرفته برای کاربران جهت شخصی‌سازی تجربه کاربری خود در سیستم

۹. گسترش پشتیبانی و آموزش

- پشتیبانی چندزبانه: اضافه کردن پشتیبانی چندزبانه برای کاربران با زبان‌های مختلف
- مرکز آموزش: ایجاد مرکز آموزش آنلاین با مقالات، ویدیوها و دوره‌های آموزشی برای کاربران

نقش هریک از اعضای گروه در پروژه به صورت جزئی

محمدامین کیانی

فعالیت عملی:

- پیاده‌سازی و استایل کلی صفحات با React JS

- اتصال صفحات به API و بکند

- برنامه‌ریزی و نظارت به عنوان سرپرست تیم

- Pair Programming

فعالیت آموزشی:

- یادگیری React JS

- یادگیری JavaScript

یزدان افرا

فعالیت عملی:

- پیاده‌سازی و استایل صفحات

- اتصال صفحات به API و بکند

- برنامه‌ریزی و نظارت به عنوان سرپرست فرانت‌اند

- Pair Programming

فعالیت آموزشی:

- یادگیری JavaScript

- یادگیری React JS

آرمان خلیلی

فعالیت عملی:

- بازطراحی UI/UX تمامی صفحات

- آماده‌سازی داک Prototype

فعالیت آموزشی:

- یادگیری HTML/CSS

- یادگیری Adobe XD

متین عصب‌الظهور

فعالیت عملی:

- پیاده‌سازی و استایل صفحات ورود و ثبت‌نام با HTML و CSS

- پیاده‌سازی و استایل صفحات

- اتصال صفحات به API و بکند

- Pair Programming

فعالیت آموزشی:

- یادگیری JavaScript

- یادگیری React JS

علی حسینی‌فرد

فعالیت عملی:

- کار روی مدل‌های یوزر و پروفایل‌های لازم برای پروژه

- تنظیم Url ها ، Serializer ها و class-based view ها برای API ها

- نظارت و برنامه‌ریزی به عنوان سرپرست بک‌اند

فعالیت آموزشی:

- تقویت یادگیری Django REST API

- Pair Programming

محمد مهدی کتابچی

فعالیت عملی:

- کار روی مدل‌های یوزر و پروفایل‌های لازم برای پروژه
- تنظیم Urlها ، Serializer ها و class-based view ها برای APIها

فعالیت آموزشی:

- یادگیری Django REST API

- Pair Programming

محمد جعفری

فعالیت عملی:

- اضافه کردن مدل درس و ساخت دیتابیس و ادمین و چند داده فرضی
- ساخت api و serializer و test برای برگرداندن درس های استاد

فعالیت آموزشی:

- یادگیری Django REST API

- Pair Programming

توضیح فایل‌های اصلی پروژه

• بک‌اند

urls.py: فایل `urls.py` در جنگو به منظور تعریف آدرس‌های URL برای یک اپلیکیشن استفاده می‌شود. این فایل مسیریابی درخواست‌های ورودی را به ویوهای مناسب انجام می‌دهد.

views.py: فایل `views.py` در جنگو (Django) نقش مهمی در مدیریت درخواست‌های ورودی به اپلیکیشن ایفا می‌کند و پاسخ‌های مناسب را بر اساس منطق تجاری و داده‌های موجود در مدل‌ها برمی‌گرداند.

models.py: فایل `models.py` که در یک پروژه Django استفاده می‌شود، شامل تعریف مدل‌های دیتابیزی است که توسط Django ORM مدیریت می‌شوند. این فایل برای تعریف انواع داده‌ها و روابط بین آن‌ها استفاده می‌شود.

serializers.py: فایل `serializers.py` در یک پروژه Django، ویژگی‌ها و رفتارهای مختلفی را برای تبدیل داده‌های مدل‌های دیتابیزی به فرمت‌های قابل استفاده برای API ارائه می‌دهد. این فایل شامل تعریف کلاس‌های سریالایزر می‌باشد که به تبدیل اطلاعات ورودی و خروجی از و به شیوه‌های مختلف می‌پردازد.

• فرانت‌اند

header in component: برای ساخت یک کامپوننت Header در React، معمولاً از یک فایل `Header.jsx` یا `Header.js` استفاده می‌شود. این فایل شامل کدهای JSX و JavaScript برای رندر کردن بخش بالایی (Header) یک صفحه وب است.

login in page: برای ساخت یک صفحه Login در React، معمولاً یک فایل `Login.jsx` یا `Login.js` ایجاد می‌شود که شامل کدهای JSX و JavaScript برای رندر کردن فرم ورود (Login) است. این فرم شامل فیلدهای ورود برای نام کاربری و رمز عبور، و دکمه‌ای برای ارسال اطلاعات است.

files in public: پوشه `public` در یک پروژه React شامل فایل‌ها و منابعی است که به صورت مستقیم توسط مرورگر قابل دسترسی هستند. این فایل‌ها بدون پردازش توسط Webpack یا دیگر ابزارهای باندلینگ مستقیماً به کاربر ارائه می‌شوند. معمولاً فایل‌های استاتیکی مانند تصاویر، فایل‌های HTML، و برخی فایل‌های پیکربندی در این پوشه قرار می‌گیرند.

globals.css: فایل `globals.css` در پروژه‌های React معمولاً برای تعریف استایل‌های عمومی و مشترک در سراسر اپلیکیشن استفاده می‌شود. این فایل به طور کلی شامل تنظیمات پایه‌ای CSS است که برای همه کامپوننت‌ها و صفحات اپلیکیشن اعمال می‌شود. هدف اصلی استفاده از این فایل ایجاد یک نقطه مرکزی برای مدیریت استایل‌های کلی و هماهنگ‌سازی ظاهر و احساس (look and feel) اپلیکیشن است.

files in api folder: پوشه `api` در یک پروژه React معمولاً شامل فایل‌ها و کدهایی است که برای ارتباط با API های سمت سرور استفاده می‌شوند. این فایل‌ها به سازماندهی درخواست‌های HTTP و مدیریت داده‌های دریافتی از سرور کمک می‌کنند. هدف اصلی این پوشه جداسازی منطق ارتباط با سرور از سایر بخش‌های پروژه و ایجاد یک نقطه مرکزی برای مدیریت درخواست‌های API است. **eslintrc.json**: فایل `eslintrc.json` در یک پروژه React (یا هر پروژه جاوااسکریپت دیگر) برای پیکربندی ESLint استفاده می‌شود. ESLint یک ابزار برای تحلیل و پیدا کردن مشکلات در کدهای جاوااسکریپت است. این فایل تنظیمات و قوانین مختلفی را که ESLint باید در طول بررسی کد اعمال کند، مشخص می‌کند.

jsconfig.json: فایل `jsconfig.json` در یک پروژه React (یا هر پروژه جاوااسکریپت دیگر) برای پیکربندی پروژه جاوااسکریپت استفاده می‌شود. این فایل به ابزارهایی مانند Visual Studio Code و TypeScript کمک می‌کند تا پروژه را بهتر بفهمند و تجربه توسعه‌دهنده را بهبود ببخشند. این پیکربندی مخصوصاً برای پروژه‌هایی که از ویژگی‌های مدرن جاوااسکریپت مانند ماژول‌ها، کامپایلرها و غیره استفاده می‌کنند، مفید است.

package – lock.json: فایل `package-lock.json` در هر پروژه‌ای که از npm (مدیر بسته‌های Node.js) استفاده می‌کند، وجود دارد. این فایل به طور خودکار توسط npm ایجاد می‌شود و به طور خاص برای مدیریت وابستگی‌ها (dependencies) و تضمین سازگاری نسخه‌ها طراحی شده است. در پروژه‌های React نیز این فایل نقش مهمی در مدیریت نسخه‌های دقیق بسته‌ها ایفا می‌کند. **tailwind.config.js**: فایل `tailwind.config.js` در پروژه‌های React یا هر پروژه‌ای که از Tailwind CSS استفاده می‌کند نقش مهمی در سفارشی‌سازی و پیکربندی تنظیمات پیش‌فرض Tailwind CSS دارد. این فایل به توسعه‌دهندگان اجازه می‌دهد که تنظیمات پیش‌فرض را تغییر دهند، کلاس‌های سفارشی ایجاد کنند و Tailwind CSS را بر اساس نیازهای پروژه خود تنظیم کنند.