

بسم الله الرحمن الرحيم

پروژه شماره یک (لینک لیست)

شبیه سازی سایت Free Lancer

سید امین کاظم پور - فراز وهاب زاده

گروه شماره: 12

کلیات پروژه:

پیکر بندی کل برنامه تشکیل یافته از ۷ کلاس هست که یکی از کلاس ها (menu) کار مدیریت برنامه را انجام میدهد سه کلاس بعدی (freeLancer ,project ,employer) نقش ساختار لینک لیست ها را ایفا میکنند و سه کلاس آخر (freeLancers ,projects ,employers) سه لینک لیست برنامه هستن در ادامه به کار تک تک کلاس ها میپردازیم.

Menu

کلاس **menu** کار چاپ کردن و ورودی گرفتن از کاربر را انجام میدهد که در کل مدیریت برنامه با این کلاس هست این کلاس دارای ۱۲ تابع می باشد که به تک تک آنها میپردازیم.

تابع **start** دیتابیس را باز میکند و سه **table** آنرا تبدیل به سه لینک لیست میکند.

تابع **menu** منو اولیه برنامه رو چاپ میکنم به کاربر امکانت **login** و ثبت نام و خارج شدن از برنامه را میدهد.

تابع **login** با توجه به درخواست کاربر به او کمک می کند که بعنوان کار فرما یا فریلنسر وارد برنامه شود.

تابع **register_freelancer** کار ثبت نام بعنوان فریلنسر را انجام میدهد.

تابع **register_employer** کار ثبت نام بعنوان کارفرما را انجام میدهد.

تابع **freelancer_profile** بعد اینکه فریلنسر توانست به برنامه لاگین کند پروفایل اون را چاپ میکنم به به فریلنسر کمک میکند تا بتواند پروفایل خود را ادیت کند، پروژه ای را جستجو کند ، حساب خود را حذف کند ، از حساب خود خارج شود، یا برنامه را ببندد.

تابع **employer_profile** بعد اینکه کارفرما توانست به برنامه لاگین کند پروفایل آن را چاپ میکند و به کارفرما کمک میکند پروفایل خود را آپدیت کند ، پروژه ای به برنامه اضافه کند ، پروژه های قبلی خود را ببیند، حساب خود را حذف کند، از حساب خود خارج شود، یا برنامه را ببندد.

تابع **edit_employer** به کارفرما کمک میکند اطلاعات پروفایل خود را آپدیت کند هر قسمت از پروفایل که فریلنسر نمیخواهد آنرا تغییر دهد باید خالی بگذارد و اینتر بزند.

تابع **my_project** پروژه های کارفرما را داخل یک جدول به همراه تمام اطلاعات خلاصه آن چاپ میکند. کارفرما میتواند با وارد کردن آیدی هر کدام از پروژه های خود اطلاعات کامل آره را ببیند و یا آنرا ادیت کند.

تابع **edit_project** به اینصورت عمل میکند که ابتدا اطلاعات کامل پروژه مورد نظر کارفرما را نشان میدهد و کار فرما می تواند با کلید واژه **delete** پروژه خود را حذف کند یا با فشار دادن اینتر فیلد های پروژه خود از ادیت کند . هر فیلدی که نمی خواد تغییر دهد باید آنرا خالی بگذارد.

تابع **add_project** به کاربر این امکان را میدهد که پروژه ای به حساب خود اضافه کند.

تابع **search** به فریلنسر اجازه میدهد که بصورت اتوماتیک فریلنسر بر اساس توانایی های خود پروژه ای را پیدا کند.

freeLanser

کلاس **freeLanser** شبیه سازی یک **node** لیست پیوندی
freeLansers هست که از دو قسمت **data** و اشاره گر **next** تشکیل شده
است. قسمت **data** دارای اطلاعات **id, username, password, first**
name, last name, abilities, bio فریلنسر هست.

Project

کلاس **project** شبیه سازی یک **node** لیست پیوندی **projects**
هست که از دو قسمت **data** و اشاره گر **next** تشکیل شده است.
قسمت **data** دارای اطلاعات **id, owner id, title, description,**
abilities, price پروژه هست.

Employer

کلاس **employer** شبیه سازی یک **node** لیست پیوندی **employers** هست که از دو قسمت **data** و اشاره گر **next** تشکیل شده است. قسمت **data** دارای اطلاعات **id , username, password, first name, lastname** کارفرما هست.

freeLancers

کلاس **freeLancers** اولین لیست پیوندی هست که اطلاعات فریلنسر داخل آن بصورت لینک لیست ذخیره میشود این کلاس دارای **4** تابع می باشد که در ادامه به تک تک آن میپردازیم.

تابع **add_end** اطلاعات کامل یک فریلنسر را میگیرد و آنرا به لیست پیوندی اضافه میکند این تابع زمان آغاز اجرای برنامه اطلاعات را از دیتابیس گرفته و لینک لیست میکند و همچنین اگر فردی بخواهد ثبت نام کند اطلاعات او را میگیرد و به لینک لیست اضافه میکند. این تابع هر **node** را که اضافه میکند آنرا نیز برمیگرداند.

تابع **login** اطلاعات فریلنسر را میگیرد و آن را با پیمایش لیست با تمام **node** ها مطابقت میدهد تا اجازه لاگین دهد اگر اطلاعات نادرست بود **false** را برمیگرداند.

تابع **edit** یک **node** و اطلاعات آپدیت شده یک فریلنسر را میگیرد تا آن را هم در لینک لیست آپدیت کند هم در دیتابیس.

تابع **delete** یک **id** فریلنسر را میگیرد تا هم حساب آن آیدی را در دیتابیس حذف کند هم با پیمایش لیست **node** آنرا حذف کند.

projects

کلاس **projects** دومین لیست پیوندی هست که اطلاعات پروژه ها داخل آن بصورت لینک لیست ذخیره میشود. این کلاس دارای 6 تابع می باشد که در ادامه به تک تک آن میپردازیم.

تابع **add_end** اطلاعات کامل یک پروژه را میگیرد و آنرا به لیست پیوندی اضافه میکند این تابع زمان آغاز اجرای برنامه اطلاعات را از دیتابیس گرفته و لینک لیست میکند و همچنین اگر کارفرما پروژه ای اضافه کند اطلاعات آن را میگیرد و به لینک لیست اضافه میکند.

تابع **my_project** آیدی یک کارفرما را میگیرد و با پیمایش لیست پیوندی هر پروژه ای که با آیدی اون کارفرما مطابقت داشته باشد لیست میکند و برمیگرداند.

تابع **employer_search** آیدی پروژه و آیدی کارفرما میگیرد و با پیمایش لیست پیوندی پروژه ای که با این اطلاعات مطابقت کند برمیگرداند اگر پروژه مورد نظر وجود نداشت **false** برمیگرداند.

تابع **freelancer_search** توانایی های یک فریلنسر را میگیرد و با پیمایش لیست پیوندی هر پروژه ای که توانایی های مورد نیاز آن با فریلنسر مطابقت کند به یک لیست اضافه میکند و آنرا برمیگرداند.

تابع **edit** اطلاعات جدید یک پروژه را همراه آیدی آن میگیرد میدانیم که این پروژه حتما وجود دارد چرا که قبلاً وجود آن بررسی شده است به این ترتیب لیست پروژه هارا پیمایش میکند و اطلاعات پروژه مورد نظر را هم در دیتابیس هم در لیست پیوندی آپدیت میکند.

تابع **delete** دو کار را انجام میدهد اگر کارفرمایی بخواد حساب خود را حذف کند تمام پروژه های کارفرما را حذف میکند ولی اگر کاربری بخواهد یک پروژه را حذف کند فقط پروژه مورد نظر آن حذف میشود به اینصورت که پروژه هم از لیست پیوندی هم از دیتابیس حذف میشود.

Employers

کلاس **employers** آخرین کلاس و سومین لیست پیوندی هست که اطلاعات کارفرما ها در آن ذخیره میشود این کلاس دارای 4 تابع است که در ادامه به تک تک آن ها اشاره میکنیم.

تابع **add_end** اطلاعات کامل یک کارفرما را میگیرد و آنرا به لیست پیوندی اضافه میکند این تابع زمان آغاز اجرای برنامه اطلاعات را از دیتابیس گرفته و لینک لیست میکند و همچنین اگر فردی بخواهد ثبت نام کند اطلاعات او را میگیرد و به لینک لیست اضافه میکند . این تابع هر **node** را که اضافه میکند آنرا نیز برمیگرداند .

تابع **login** اطلاعات کارفرما را میگیرد و آن را با پیمایش لیست با تمام **node** ها مطابقت میدهد تا اجازه لاگین دهد اگر اطلاعات نادرست بود **false** را برمیگرداند.

تابع **edit** یک **node** و اطلاعات آپدیت شده یک کارفرما را میگیرد تا آن را هم در لینک لیست آپدیت کند هم در دیتابیس.

تابع **delete** یک **id** کارفرما را میگیرد تا هم حساب آن آیدی را در دیتابیس حذف کند هم با پیمایش لیست **node** آنرا حذف کند . البته این تابع ، تابع **delete** پروژه را نیز فراخوانی میکند تا همه پروژه های این کارفرما حذف شود.