

# گزارش پیشرفت ۳۰ درصد پروژه

کیان بهزاد : ۹۵۲۳۰۱۷

پارسا اسدی : ۹۵۲۳۰۰۵

امیر بیات : ۹۵۲۳۰۱۸

۹ خرداد ۱۳۹۷

## ۱ مقدمه

در بخش ۳۰ درصد ابتدایی پروژه موارد زیر مد نظر بوده است:

- ساخت صفحه Login و Signup برای ثبت نام کاربر و وارد شدن کاربر به صفحه اصلی پیامرسان و پیاده سازی بخش سرور آن به صورت کامل و پایدار.
- در نظر گرفتن امکان فراموشی کلمه عبور توسط کاربر
- اضافه کردن امکان به به خاطر سپردن کلمه عبور کاربر توسط برنامه (مانند سایر پیامرسان ها)
- پیاده سازی بخش GUI صفحه تبادل پیام (به صورت آزمایشی و ابتدایی).
- امکان ارسال کد تاییدیه به ایمیل کاربر هنگام ثبت نام کاربر

## ۲ بخش کاربری

در این قسمت به بررسی بخشی از پروژه که طرف کاربر است می پردازیم.

## ۱.۲ کلاس Zprava

کلاس Zprava که از کلاس QMainWindow ارث بری می کند و صفحه اصلی پروژه را می سازد.

در این کلاس یک اشاره گر وجود دارد که به شی از کلاس ZpForm اشاره می کند. در اینجا مختصری از ZpForm توضیح داده می شود و در ادامه بیشتر تشریح می شود. ZpForm کلاسی است برای طراحی المان های قسمت Signup و Signin. در نتیجه اشاره گری به ZpForm داخل کلاس صفحه اصلی پروژه یعنی Zprava تعریف کردیم تا در نهایت به صفحه اصلی افزوده شود و به نمایش گذاشته شود.

داخل کانستراکتور کلاس Zprava ابتدا ابعاد صفحه داده شده اند. و سپس المان ها به صفحه اصلی افزوده شده اند. نکته ای که در اینجا وجود دارد این است که برای اضافه کردن المان ها به صفحه ابتدا QWidget را در یک QLayout قرار می دهیم و آنرا در نهایت به یک QWidget تبدیل میکنیم. این کار دسترسی بهتری به المان های صفحه را نتیجه می دهد.

در اینجا نیز به همین شکل این کار انجام شده است. یعنی ابتدا form از نوع ZpForm است که ZpForm از کلاس QWidget ارث بری می کند. سپس به QLayout اضافه شده است و در نهایت این QLayout به QWidget که نام آن w است اضافه گشته و با تابع setCentralWidget به صفحه اصلی اضافه شده است.

## ۲.۲ کلاس ZpForm

Parent این کلاس QWidget است. زیرا در کلاس Zprava در تابع setCentralWidget باید یک QWidget ورودی دهیم. در نتیجه این کلاس از QWidget ارث بری می کند. کلاس دارای چند دسته از method ها و feature ها است که وظایف متعددی بر عهده دارند. ساخت لمان های داخلی کلاس و طراحی آنها، ارسال اطلاعات در شبکه، ساخت صفحه verify و ساخت button ها و تعریف کارکرد آن ها از جمله این وظایف است. در ادامه توابع تعریف شده در کلاس را مورد بررسی قرار میدهیم.

- Constructor :

در این تابع ورودی بولی برای این است که اگر کاربر قبلا وارد شده باشد و گزینه remember me را فعال کرده باشد برای دفعات بعدی مستقیم وارد محیط پیامرسان شود. این کار توسط تابع is\_kept\_logged\_in انجام می پذیرد. .

- apply\_stylesheet :

این تابع فایل Qss را که برای Styling برنامه طراحی شده است را به کد اضافه می کند.

- create\_form\_widget :

در این تابع المان های صفحه به ترتیب گفته شده در بالا ساخته و پرداخته می شوند. یک نمونه را مورد بررسی قرار می دهیم، مثلا در بخش Signup ابتدا آیکون بالایی از نوع QLabel که نوعی Qwidget است ساخته می شود. سپس اسم و فایل آن به برنامه معرفی می شود. سپس layout ساخته می شود و چون المان های بخش ثبت نام به صورت افقی و زیر هم تعریف می شوند در نتیجه از QHBoxLayout استفاده کردیم. و تغییراتی در آن دادیم و سپس آنرا به QWidget تبدیل کردیم. برای بقیه نیز همین مراحل طی شده است. در نهایت کل بخش Signup به همین ترتیب پرداخته شده است و در نهایت تمام بخش ورود و ثبت نام هم به همین ترتیب اعمال ساخته شده اند.

- create\_verify\_widget :

این تابع صفحه Verify را می سازد و که آن المان های صفحه طراحی شده اند و سیگنالی برای عمل کلیک کردن بر روی Verify Button ساخته شده است که تابع slotVerify\_Button\_Clicked را فراخوانی می کند. این صفحه وقتی ظاهر می شود که بر دکمه ی Signup کلیک کنیم. در نتیجه ی این کار ایمیلی حاوی کد فعالسازی به کاربر ارسال می شود.

- `:create_forget_widget`

این تابع مانند تابع قبلی که توضیح داده شد کار میکند و تفاوت اساسی با آن ندارد. `forget`.

- `: slotFading_#_widget`

این تابع به تعداد برای موارد مختلف استفاده شده است (برای همین از `#` استفاده شده است) وقتی در برنامه قرار است صفحه جدیدی باز شود از این تابع استفاده میکنیم و با استفاده از `fade out` کمی انیمیشن به پنجره ها اضافه می کنیم تا زیبا تر شود.

- `:initiate_networking`

در این تابع برای تبادل اطلاعات با سرور از کلاس `QNetworkAccessManager` استفاده شده است، و اطلاعات از طریق پروتوکل `http` و متد `get` مبادله می شود.

- `: handle_reply`

این تابع با توجه با `State` کاربر و پاسخ سرور، `QAction` ه ا را مقداری تغییر می دهد. مثلا وقتی `lrUsername` از قبل در سرور موجود باشد علامت خطا در صفحه ظاهر می شود.

- `: slotReadyRead`

سرور برای ما اطلاعات را ارسال میکند هر وقت این اطلاعات قابل خواندن شدند این تابع این اطلاعات را دریافت میکند.

### ۳ بخش سرور

ابتدا تصمیم تیم بر آن بود تا از node.js به عنوان زبان برنامه نویسی سرور استفاده کند. در نتیجه در ابتدا قسمت Login و Signup بدین صورت پی ریزی شد و Database آن توسط فایل های Json نوشته شد.

اما بعد از نوشته شدن Interface در سمت کاربر و تحقیقات بیشتر در مورد سرور تیم به این نتیجه رسید که Django نیاز های ما را بهتر برطرف می کند. پس با استفاده از این Framework و SQLite3 به عنوان Database سرور Zprava بازنویسی شد. حال در زیر به بررسی نحوه عملکرد سرور میپردازیم.

#### ۱. Signup Application

این برنامه از سه بخش تشکیل شده است:

##### • Registration

در این قسمت Username و Password و Email توسط Query های URL از کاربر گرفته می شود. و در صورت وجود نداشتن Username و Email کاربر به Database اضافه شده و Verification Code برای کاربر ارسال می شود.

##### • Verification

در این قسمت Username و Verification Code توسط Query های URL از کاربر گرفته می شود. و در صورت وجود Username و همخوانی کد فعال سازی اکانت کاربر فعال می شود. برای کاربر ارسال می شود.

##### • Handshaking

در این قسمت سرور در صورت دریافت کلمه قرار دادی Hello پاسخ Hello back را برای کاربر ارسال می کند تا دو طرف متوجه برقراری ارتباط با یکدیگر شوند.

#### ۲. Login Application

این برنامه از دو قسمت تشکیل شده است:

#### • Login

در این قسمت Username و Password توسط Query های URL از کاربر گرفته می شود. و در صورت موجود بودن آن ها کاربر اجازه دسترسی به حساب کاربری خود را پیدا می کند.

#### • Forget

در این قسمت Email توسط Query های URL از کاربر گرفته می شود. و در صورت موجود بودن آن Username و Password برای کاربر ارسال می شود.

#### ۳. Database

سرور Zprava از ابزار SQLite3 برای ذخیره اطلاعات استفاده میکند. به عنوان مثال هر کاربر اضافه شده به پیامرسان دارای پنج ویژگی در Database خواهد بود. این اطلاعات شامل Username و Password و Email و Verification Code و متغیر بولی is verified برای چک کردن کد تاییدیه است.