

بسم الله الرحمن الرحيم

پروژه درس برنامه نویسی پیشرفته

ساخت ربات تلگرام

محمد رضی فیجانی

۲۶ تیر ۱۳۹۷

۱ تعریف پروژه

در این پروژه سعی شده تا رباتی برای توسعه کانال با استفاده از ربات تلگرام ساخته شود. مفهوم جدید branch نیز به جای کانال استفاده شده است. روند کار ساده و قابل توسعه است. به این صورت که در ابتدا یک نفر در بات یک branch را میسازد و همچنین می تواند عده ای را بعنوان کمکی خود یا به عبارتی Co-Admin در نظر بگیرد.

پس از ایجاد آن، دیگران میتوانند عضو آن شوند. این عمل با دانستن اسم branch که هم نام با اسم کانال است، به راحتی میسر خواهد بود. همچنین امکان سرچ آنلاین برای دیدن تمامی branch ها نیز در نظر گرفته شده است. افراد پس از ورود به branch مورد نظر، می توانند برای ادمین و کو-ادمین ها مطلبی را ارسال کنند که با تایید آن ها آن متن به کانال وابسته به branch ارسال میشود.

۲ بررسی فنی پروژه

۱.۲ اطلاعات سرور

باتوجه به فیلتر شدن Telegram و همچنین عدم داشتن سرور، به اجبار از سرور رایگان **server Jahan** استفاده کردیم که آن نیز ما را مجبور به استفاده از php بعنوان زبان برنامه نویسی کرد. تمامی فایل های این پروژه توسط این **FTP** قابل مشاهده است. فایل های اصلی در فولدر های **bot** (فایل های ایجاد بات تلگرام) و همچنین **channel_manager** (جست و جوی آنلاین branch ها) قرار گرفته اند.

اطلاعات بیشتر در **جدول ۱** موجود می باشد.

جدول ۱: اطلاعات سرور

telegramadminpro.ml	server
telegr28	Username
telegram123456	Password
21	Port
https://github.com/mohammadraziei1375/Ap-project.git	Git

۲.۲ توضیح اولیه

کد های این بات به طور خاص، در فایل `channel_manager.php` وجود دارد. این فایل شامل تمامی اطلاعات این بات است. دیتابیس در نظر گرفته شده همان طور که **شکل ۱** نشان میدهد؛ بسیار جامع است. و حتی فارغ از نوع مسنجر (تلگرام و ...) می باشد.

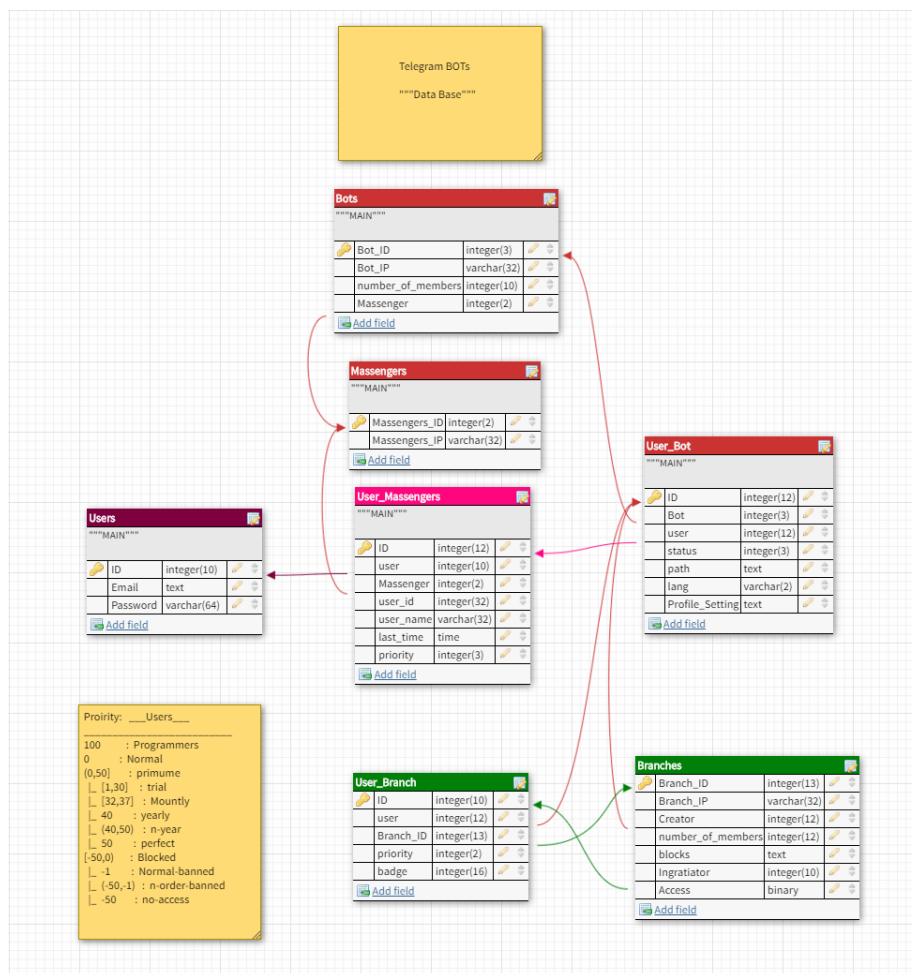
۳.۲ Telegram API

در این پروژه سعی شد تا بدون استفاده از کتابخانه های معرفی شده خود تلگرام و تنها با استفاده از API تلگرام و query کردن، اطلاعات مورد نیاز را در حالت فایل json به عنوان ورودی آنالیز کنیم و فرایند ساخت ربات را پیش ببریم.

فایل `bot_class.php` واقع در دایرکتوری `bot` همین کار را برای ما انجام میدهد. همچنین فایل `input_analyzer_class.php` برای آنالیز ورودی و تهیه متغیر `$input` مورد استفاده قرار میگیرد. منظور از آنالیز این است که تلگرام هنگام query کردن سرور ما، token را نیز میدهد. با گرفتن آن، و مقایسه با token اصلی، میتوان تشخیص داد که این درخواست؛ معتبر است یا خیر. در صورت عدم تطابق درخواست را رد یا به اصطلاح `die()` می کند. همچنین همانطور که توضیح داده شد؛ ورودی به صورت json می باشد. بنابراین، در صورت تطابق، برای دسترسی راحت تر آن را درون متغیر `$input` به صورت آرایه ذخیره میکنیم.

۴.۲ چند زبانی

از افتخارات این پروژه در نظر گرفتن خاصیت چند زبانی (Multi Language) بودن آن است. برای این کار از کلاس `parser` استفاده شده است که داخل فایل `parser_class.php` قرار دارد. این کلاس دارای توابعی است که در **جدول ۲** توضیح داده شده اند. تنها کافیسست که value های فایل json ترجمه شوند تا بتوان آن هارا چندین زبان بدون دست بردن به کد حتی یک ذره نمود. این روش از ابداعات و خلاقیت های این پروژه بوده است و روش مشابه با آن از قبل دیده نشده بود.



شکل ۱: دیتابیس مورد استفاده

۵.۲ دیتابیس

دیتابیس مورد استفاده همان **شکل ۱** می باشد. چند نکته بایستی در مورد آن متذکر شد.

- در **شکل ۱** هر جدولی که زیر اسم آن `""main""` نوشته شده باشد، به صورت عمومی است. یعنی برای هر تعداد بات و حتی مستجری قابل استفاده است.
- جدولی که با رنگ سبز مشخص شده اند به طور خاص برای این بات مورد استفاده قرار میگیرند.
- تمامی branch ها در داخل جدول branches یافت میشوند. هر یک از آن ها یک سازنده دارد و همچنین تعدادی را در قالب جیسون بعنوان بلاک میگیرد.

جدول ۲: کلاس parser

constructor	تابع سازنده کلاس که فایل ورودی و زبان را به صورت رشته دریافت میکند. فایل در حال بررسی را به صورت رشته آدرس آن را خروجی میدهد. این خروجی با ورودی تابع سازنده متفاوت است.
langFile	مثلا اگر فایل و زبان ورودی تابع سازنده به صورت "default.json" و "fa" باشد؛ آنگاه خروجی این تابع به صورت "default_fa.json" می باشد. در صورت عدم یافتن آن فایل "default_en.json" خوانده میشود. همچنین در صورت عدم یافتن آن، هر زبان دیگری را که وجود داشته باشد، را برمیگرداند.
lang	همان زبان را به صورت خروجی می دهد.
subFile	این تابع مثلا "help.pdf" را میگیرد و مثلا "help_fa.json" را خروجی میدهد.
parse	این مهمترین تابع است. دو ورودی به صورت رشته میگیرد: دامنه و کلید. کلید که تعریفش مشخص است و خروجی مقدار آن کلید است. دامنه به مجموعه کلید های مرتبط که در یک کلاس ریخته شده اند اطلاق میشود. همان گونه که گفته شد خروجی به صورت رشته است. این تابع عامل چند زبانی شدن برنامه و پراستفاده ترین تابع آن است.

- هرکس در داخل بات اولاً یک یوزر بات است و ثانياً در پس از ایجاد کردن یا پیوستن به هر شاخه، یک یوزر برنج نیز میشوند. **priority** هر یوزر برنج، در حقیقت سمت هرکس در آن شاخه است. بعنوان مثال؛ برای سازنده، ۱۰ و برای کو-ادمین ها، ۹ و برای اعضا، ۱ است. حالت صفر نیز مخصوص افراد بلاک شده است. حالت منفی در است مفروض نیست. اما در برنامه نویسی ربات، برای انتخاب کردن شاخه در حال کار، priority آن را قرینه کردیم. در واقع هرکس حداکثر یک priority منفی دارد.
- یوزر-بات، جدولی است که می گوید هر یوزری در داخل هر باتی، یک status و یک path دارد. این دو برای ایجاد دکمه هایی مانند back و forward لازم هستند. فایل [stack_class.php](#) به پردازش این ۲ اطلاعات می پردازد.

۳ نحوه کار کردن با بات

ابتدا از ما انتخاب زبان را میخواهد. سپس با توجه به زبان انتخاب شده متن را تنظیم میکند. سپس از ما میخواهد یا log in یا log up کنیم یا به طور کلی از این وضعیت خارج شویم. در نوار زیرین دکمه های back و forward و home و restart به صورت ایموجی وجود دارد.

برای ایجاد یا پیوستن به یک شاخه کافیت روی menu کلیک کنیم. حال مثلا وارد ایجاد کردن میشویم.

در این حالت ابتدا نام کانال مورد نظر را انتخاب میکنیم. سپس تا زمانی که done را نزده ایم، به انتخاب کو ادمین ها می پردازیم. برای این کار کافیت، یا یوزرنیم آن ها را بدهیم، یا با استفاده از یک پیام فورواردی از آن ها این کار را کنیم. برای پیوستن هم کافیت این بار پیوستن را انتخاب کنیم و پس از جست و جو کردن در **سایت** یا وارد کردن نام کانال، به آن پیوست.

از طریق قسمت پروفایل، پروفایل به طور مناسبی قابل مشاهده می باشد. با رفتن روی هر شاخه و فرستادن پست به آن میتوان از آن استفاده کرد.

پایان