

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



**دانشگاه بزرگمهر قائنات**  
**دانشکده فنی و مهندسی**

**پایان نامه دوره کارشناسی پیوسته مهندسی کامپیوتر – نرم افزار**

**سامانه اطلاع رسانی خبرنامه**

**استاد راهنما:**

**جناب آقای مهندس سید مجتبی بنائی**

**دانشجو:**

**رضا صفائیان**

**تابستان ۱۳۹۶**

## تعهدنامه رعایت اصالت محتوایی

بدین وسیله اعلام می دارم که اصالت کار انجام گرفته در این پایان نامه به غیر از موارد نقل قول شده و ارجاع داده شده، متعلق به اینجانب بوده و در نگارش و تدوین این پایان نامه کلیه موارد رعایت شئونات فرهنگی و اصول حق تالیف (کپی رایت) منظور شده است.

رضا صفائیان

۹۲۱۲۶۰۶۰۱۲

بهار ۹۶

## تشکر و قدردانی

از استاد گرامی جناب آقای مهندس سید مجتبی بنائی بسیار سپاس گزارم برای اینکه در دوران تحصیل به طور خالصانه دانش و دانسته های ارزنده خود را در اختیارمان قرار دادند و همچنین بدون راهنمایی های ایشان نگارش این پایان نامه بسیار مشکل می نمود.

## چکیده

امروزه نقش ارتباطات در دنیای مجازی بر هیچ کس پوشیده نیست. در دنیایی که روزنامه های کاغذی جای خود را به رسانه های دیجیتال و خبرگزاری های آنلاین داده اند. از این رو طراحی و توسعه نرم افزارهای ارتباط جمعی و شبکه های مجازی بیش از پیش گسترش یافته و هر کدام به نوعی سعی در جذب مخاطب و ارائه خدمات به کاربران خود را دارند.

سامانه خبرنامه نیز جهت سهولت در ارتباط بین سازمان ها، ادارات، محیط های آموزشی و ... با مخاطبین خود ایجاد شده است. به این صورت که خبرها در دسته بندی های مختلف قرار میگیرند و کاربران نیز به گروه های مختلف تقسیم بندی می شوند. سپس هر گروه کاربری به دسته بندی های خبری خاص خود دسترسی خواهد داشت.

مزیت استفاده از این سامانه، سرعت بالا، امنیت و سهولت در بروزرسانی اخبار و اطلاعات از سمت پنل مدیریت می باشد.

## فهرست مطالب

فصل ۱.....	۱
۱-۱- مقدمه.....	۲
۲-۱- شرح سیستم.....	۳
۳-۱- امکانات و ویژگی های سامانه.....	۴
۴-۱- بررسی سیستم های مشابه.....	۱۱
فصل ۲.....	۱۲
۱-۲- زبان و تکنولوژی های استفاده شده در پروژه.....	۱۳
۲-۲- توضیح پوشه ها و نحوه ساختار بندی پروژه.....	۱۴
۳-۲- توضیح نحوه عملکردهای کد های سامانه.....	۱۶
۱-۳-۲- ورود.....	۱۶
۲-۳-۲- خروج.....	۱۷
۳-۳-۲- دریافت لیست شهرها.....	۱۷
۴-۳-۲- مشاهده پلن های مالی.....	۱۸
۵-۳-۲- نمایش داشبورد مدیریت.....	۱۹
۶-۳-۲- مدیریت نقش های کاربری.....	۱۹
۷-۳-۲- مشاهده مجموعه ها.....	۲۰
۸-۳-۲- جزئیات مجموعه های تعریف شده.....	۲۱
۹-۳-۲- اختصاص مجموعه های مجاز هر نقش.....	۲۲
۱۰-۳-۲- انتساب مجموعه های پیش فرض هر نقش.....	۲۳
۱۱-۳-۲- ویرایش مجوزهای کاربران.....	۲۴
۱۲-۳-۲- مدیریت نقش های کاربران.....	۲۵
۱۳-۳-۲- افزودن کاربر به عنوان نویسنده.....	۲۶
۱۴-۳-۲- افزودن خبر جدید.....	۲۷
۱۵-۳-۲- نمایش اخبار سیستم.....	۲۹
۱۶-۳-۲- گرفتن مجموعه های یک سیستم.....	۲۹
۱۷-۳-۲- ویرایش خبر.....	۳۰

۳۰	۲-۳-۱۸- نمایش اخبار در انتظار تأیید
۳۲	۲-۳-۱۹- افزودن کاربر جدید
۳۳	۲-۳-۲۰- مشاهده و مدیریت کاربران سامانه
۳۵	۲-۳-۲۱- گرفتن نقش ها
۳۵	۲-۳-۲۲- افزودن کاربر مهمان
۳۶	۲-۳-۲۳- نمایش راهنمای سیستم
۳۷	۲-۳-۲۴- نمایش لیست انتقادات و پیشنهادات
۳۸	۲-۳-۲۵- ارسال پیام به کاربران
۳۹	۲-۳-۲۶- مشاهده پیام های دریافتی
۴۰	۲-۳-۲۷- لیست پیام های ارسال شده
۴۱	۲-۳-۲۸- مشاهده نظرات خبرهای ثبت شده
۴۲	۲-۳-۲۹- ایجاد نظرسنجی جدید
۴۳	۲-۳-۳۰- مشاهده نتایج نظرسنجی ها
۴۴	۲-۳-۳۱- ارسال تیکت جدید
۴۵	۲-۳-۳۲- نمایش تمام تیکت های سیستم
۴۵	۲-۳-۳۳- نمایش یک تیکت
۴۶	۲-۳-۳۴- تنظیمات اعلان ها
۴۷	فصل ۳
۴۸	۳-۱ مقدمه
۶۴	فصل ۴
۶۴	نتیجه گیری
۶۵	نتیجه گیری
۶۶	مراجع
۶۷	آدرس پروژه روی GIT
۶۸	پیوست ۱
۸۲	پیوست شماره ۲- ارتباط جداول

ضمیمه ۱..... ۸۳

ضمیمه ۲..... ۸۴



**فهرست تصاویر**

- شکل ۱-۱ : نمایی از داشبورد مدیر در وب سایت..... ۴
- شکل ۲-۱ : نمایی از مدیریت مجموعه ها..... ۵
- شکل ۳-۱ : نمایی از بخش اختصاص مجموعه های پیش فرض..... ۶
- شکل ۴-۱ : نمایی از صفحه افزودن کاربر جدید..... ۷
- شکل ۵-۱ : نمایی از بخش مجوز های دسترسی..... ۸
- شکل ۶-۱ : نمایی از بخش ایجاد نظر سنجی جدید..... ۹
- شکل ۷-۱ : نمایی از بخش انتقادات و پیشنهادات در پنل وب..... ۱۰
- نمای کلی ارتباط جداول..... ۸۲

# فصل ۱

**۱-۱- مقدمه**

با توجه به گسترش روز افزون اینترنت و همچنین تلفن های هوشمند، بستری مناسب فراهم شده است تا بتوان ارتباطی مناسب با مردم و افشار مختلف برقرار نمود. ضمن اینکه روش های سنتی اطلاع رسانی مانند پیامک با محدودیتهایی مواجه هستند که از جمله آنها می توان به عدم امکان ارسال پیام های چندرسانه ای و یا طولانی، هزینه بر بودن پیام برای سامانه های پیامکی و همراه نبودن همیشگی رایانه برای سیستم های مبتنی بر وب اشاره کرد.

سامانه های مبتنی بر وب و همینطور تلفن همراه با توجه به همراهی همیشگی با مخاطب، بستری مناسب را برای اطلاع رسانی نوین فراهم می کنند و امکان ارتباط سریع و دوطرفه بین مخاطب و سازمان را نیز فراهم می کنند. در همین راستا سیستم جامع اطلاع رسانی همراه طراحی شده تا بتوان امکان ارتباط و همچنین تعامل دوطرفه بین مخاطبان را فراهم کرد.

## ۱-۲- شرح سیستم

سیستم جامع خبرنامه به طور کلی یک سامانه اطلاع رسانی مبتنی بر وب ، می باشد. هدف این سیستم ،ارتباط سریع و آسان میان مدیران و مسئولین با کاربران و افراد یک مجموعه و اطلاع رسانی به گروه های کاربری می باشد، که این کار از طریق دسته بندی مخاطبین براساس نقش های سیستمی هر کاربر صورت می پذیرد.

به زبان ساده تر، کاربران براساس نقش هایی که به آنها داده می شود ، دسته بندی می شوند (مثلا مدیران ، کارمندان ، کاربران عادی ، مهمانان و ...) و به هر یک از این نقشها ، مجموعه هایی نسبت داده می شود.(یا تنظیم می شود که هر یک از این نقش ها اخبار کدام مجموعه ها را دریافت کنند) سپس بعد از این کار، خبر موردنظر به مجموعه هایی ارسال می شود ، و کاربرانی که نقش شان اجازه دریافت اخبار آن مجموعه ها را داشته باشد، خبر مورد نظر را دریافت و مشاهده می کنند.

با توجه به پلنی که توسط مشتریان انتخاب میشود، امکانات متفاوتی در اختیار آنها قرار می گیریددر این پلن ها، در پنل وب، امکان ایجاد زیر سیستم به مدیر داده می شود که بعد از تعریف زیرسیستم ها هر یک بصورت مجزا با اخبار و کاربرانی مجزا شروع به کار میکنند.

### ۱-۳- امکانات و ویژگی های سامانه

سیستم جامع اطلاع رسانی  
صفحه اصلی راهنما درباره

مدیریت سیستم

داشبورد

نظرات خبر جدید  
مشاهده جزئیات

پیشنهادهای جدید  
مشاهده جزئیات

پیام جدید  
مشاهده جزئیات

آخرین بازدیدها:

ردیف	عنوان خبر	کاربر	تاریخ
۱	پاتریک مودیانو برنده نوبل ادبیات ۲۰۱۴	کاربر مهران	۱۱:۰۴:۴۳ ساعت ۱۳۹۵/۱۲/۱۱

شکل ۱-۱: نمایی از داشبورد مدیر در وب سایت

#### • مدیریت اخبار:

درج ، ویرایش ، حذف اخبار و مشاهده آمار هر خبر (تعداد بازدید ها و نظرات و لایک ها) پنل وب، فقط کاربرانی که نقش مدیر و ویرایشگر به آنها داده شده ، اجازه دسترسی دارند، و دیگر کاربران عادی فقط می توانند از اپ موبایل استفاده کنند، تفاوت مدیر و ویرایشگر در این است که ویرایشگر از نظر انتشار مستقیم خبر(خبر مستقیما منتشر شود یا به حالت انتظار برود و بعد از تایید مدیر منتشر شود) و یا ویرایش خبرها (فقط خبرهایی که نویسنده خبر ، همین ویرایشگر باشد یا کل اخبار) و دسترسی به مجموعه ها قابل محدود شدن است.

فقط کاربرانی که به خبر درج شده دسترسی دارند، به محض درج خبر یک نوتیفیکیشن دریافت میکنند، و میتوانند خبرها را لایک و به خبرهای مورد علاقه خود اضافه کنند، برای هر خبر نظر ارسال کنند که پس از تایید مدیر، نظر به دیگر کاربران نمایش داده میشود، همچنین میتواند پربازدیدکننده ترین و محبوب ترین خبرها را مشاهده کنند و خبری را جستجو کنند.

## • مدیریت مجموعه های خبری:

براساس مجموعه ها دسته بندی و به افراد دلخواه ارسال می شوند.

مدیر می تواند، مجموعه های جدید خبری تعریف و مدیریت کند ، برای هر نقش ، مجموعه هایی دلخواه نسبت دهد و کاربران هر مجموعه را مشاهده کند.

اخبار به مجموعه ارسال میشود و نقش ها فقط اخبار مجموعه هایی را که به آنها نسبت داده شده دریافت میکنند، و کاربران براساس نقش ها دسته بندی می شوند.

پس مدیر میتواند تعیین کند که هر کاربر، دقیقا به کدام مجموعه های خبری دسترسی داشته باشد، و کاربر نیز می تواند تنظیم کند که از بین این مجموعه هایی که به آنها دسترسی دارد ، اخبار کدامیک را دریافت کند.

**مدیریت مجموعه ها**

ابتهدا بر روی مجموعه دلخواه راست کلیک کرده و سپس زیرمجموعه ها را ایجاد نمایید.

سیستم جامع اطلاع رسانی صندوق بیمه کشاورزی

- اخبار عمومی صندوق بیمه
- آیین نامه ها
- باغبانی و منابع طبیعی
- کارگزار انعقاد
- کارگزار ارزیاب خسارت
- دام و طیور و آبزیان
- کارگزار ارزیاب خسارت
- کارگزار انعقاد
- مخصوص مدیرعامل
- مناسبتها

داشبورد

مدیریت خبرها

مجموعه ها

مدیریت مجموعه ها

مجموعه های پیش فرض هر نقش

نقش ها

مدیریت کاربران

مدیریت نقش های کاربران

مدیریت تصاویر اسلایدر

راهنمای سیستم

شکل ۱-۲: نمایی از مدیریت مجموعه ها

- امکان تعیین مجموعه های پیش فرض خبری برای هر گروه کاربری

برای هر گروه از کاربران، می توانیم تعیین کنیم که چه مجموعه های خبری را به صورت پیش فرض بتوانند مشاهده کنند . هر چند به ازای یک کاربر خاص، می توان این مجموعه ها را کم و زیاد کرد.

**مجموعه های پیش فرض هر نقش**

---

انتخاب

▼

کارگزار انعقاد دام سبک و سنگین

نقش ها :

سیستم جامع اطلاع رسانی صندوق بیمه کشاورزی

☒

اخبار عمومی صندوق بیمه

☒

آگین نامه ها

☒

باغبانی و منابع طبیعی

☐

کارگزار انعقاد

☐

کارگزار ارزیاب خسارت

☐

دام و طیور و آبزیان

☒

کارگزار ارزیاب خسارت

☐

کارگزار انعقاد

☒

مخصوص مدیرعامل

☐

مناسبتها

☒

ثبت

**شکل ۱-۳ : نمایی از بخش اختصاص مجموعه های پیش فرض**

- مدیریت کاربران:

مدیر می تواند کاربران جدید تعریف کند، کاربران قبلی را ویرایش و حذف کند، نقش های هر کاربر را تغییر دهد، کاربرانی که به پنل وب دسترسی دارند را ، از نظر سطح دسترسی به منو ها محدود کند و کاربرانی که نقش ویرایشگر را دارند، همانطور که قبلا گفته شد، از جهات مختلف محدود کند.

---

<input type="text"/>	نام :
<input type="text"/>	نام خانوادگی :
<input type="text"/>	نام کاربری :
<input type="text"/>	گذرواژه :
<input type="text"/>	تکرار گذرواژه :

**ذخیره**

---

برای ثبت کاربران به صورت گروهی ، ابتدا فایل نمونه زیر را دانلود نموده و پس از تکمیل اطلاعات آنرا بارگزاری نمایید.

**دانلود فایل نمونه**

No file selected. **...Browse**

گروه کاربری :

**ذخیره**

### شکل ۴-۱ : نمایی از صفحه افزودن کاربر جدید

البته برای کاربری که به صورت دستی تعریف می شود، باید بعد از تعریف کاربر، دسترسی ها و نقش هایی که خواهد داشت را در بخشی جداگانه (زیر منوی مدیریت نقش های کاربر) تنظیم کرد.

در بخش مدیریت کاربران، می توان کاربران را جستجو کرده یا براساس گروه کاربری آنها را جداگانه مشاهده کرد و با کلیک روی هر کاربر ، اطلاعات پایه او مانند نام و نام خانوادگی و رمز عبور را تغییر داد.



## • تعریف ویرایشگر و تعیین مجوزهای دسترسی

می توان علاوه بر ادمین سیستم، کاربرانی دیگر را نیز برای ورود اخبار در نظر گرفت که از آنها به عنوان ویرایشگر یاد می کنیم . ویرایشگران علاوه بر نرم افزار اندروید، می توانند وارد پنل وب شده و برای مجموعه هایی که مجوز لازم به آنها داده شود، خبر گذاشته و یا سایر بخشهای مدیریتی سیستم را برعهده بگیرند.

سیستم به گونه ای طراحی شده است که تمام منوی های موجود در پنل وب را می توان برای یک کاربر ویرایشگر، فعال یا غیر فعال کرد (از زیر منوی مجوزهای دسترسی).

**مجوزهای دسترسی**

انتخاب

نام کاربری : ۰۹۱۵۳۶۱۰۲۱۳۴

مجوز های کاربر : سارا بلندی

سیستم مدیریت مجوز ها	<input checked="" type="checkbox"/>
داشبورد	<input checked="" type="checkbox"/>
مدیریت خبرها	<input checked="" type="checkbox"/>
افزودن خبر جدید	<input checked="" type="checkbox"/>
مدیریت خبرها	<input checked="" type="checkbox"/>
خبرهای در انتظار تایید	<input type="checkbox"/>
مجموعه ها	<input type="checkbox"/>
مدیریت مجموعه ها	<input type="checkbox"/>
مجموعه های پیش فرض هر نقش	<input type="checkbox"/>
نقش ها	<input type="checkbox"/>
مدیریت نقش های سیستم	<input type="checkbox"/>
مدیریت کاربران	<input type="checkbox"/>
افزودن کاربران جدید	<input type="checkbox"/>
لیست کاربران	<input type="checkbox"/>
مدیریت نقش های کاربران	<input type="checkbox"/>
مدیریت نقش های کاربر	<input type="checkbox"/>
تعریف ویرایشگر	<input type="checkbox"/>
مجوز های دسترسی	<input type="checkbox"/>
مدیریت تصاویر اسلایدر	<input checked="" type="checkbox"/>
راهنمای سیستم	<input type="checkbox"/>

**ثبت**

**شکل ۱-۵: نمایی از بخش مجوز های دسترسی**

## • ارسال و دریافت پیام:

مدیر میتواند به صورت گروهی و یا تکی به کاربران پیام ارسال کند و کاربران نیز میتوانند ، به پیام ها پاسخ بدهند و بین مدیر و کاربران به صورت جدا ارتباط برقرار شود. با ارسال پیام جدید از سمت وب ، کاربر یا کاربران نوتیفیکیشن پیام جدید دریافت می کنند.

## • نظر سنجی:

مدیر می تواند ، یک نظرسنجی جدید ایجاد کند، هر کاربر فقط میتواند یک بار در این نظرسنجی شرکت کند، مدیر نتیجه کامل نظرسنجی را مشاهده می کند. بعد از از درج نظرسنجی نیز، به کاربران نوتیفیکیشن نظرسنجی ارسال می شود.

مدیریت سیستم

سیستم جامع اطلاع رسانی

صفحه اصلی راهنما درباره

خبرنامه

?

ایجاد نظرسنجی

گزینه ها :

افزودن گزینه :

-

+

متن نظرسنجی :

متن نظرسنجی را وارد کنید.

زیر سیستم ها :

همه

بعدا فعال شود.

ثبت نظرسنجی

داشبورد

پیام ها

مدیریت خبرها

مجموعه ها

مدیریت کاربران

نظرسنجی

ایجاد نظرسنجی

نتیجه نظرسنجی

نظرات خبرها

انتقادات و پیشنهادات

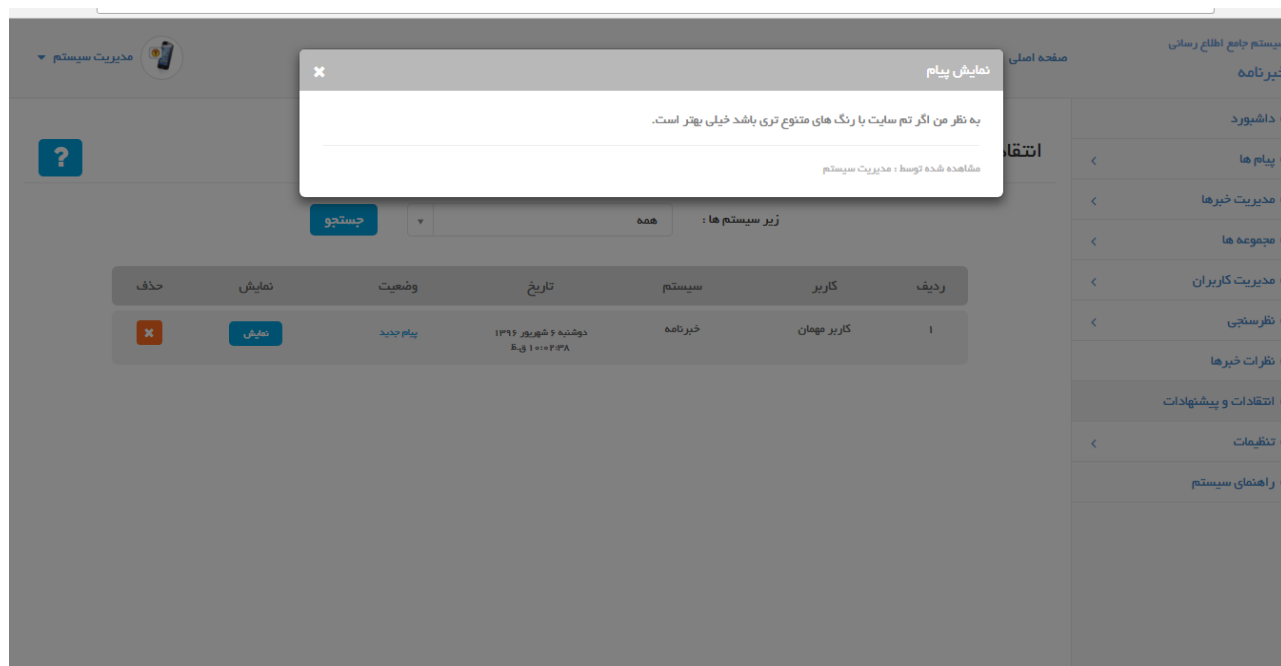
تنظیمات

راهنمای سیستم

شکل ۱-۶: نمایی از بخش ایجاد نظرسنجی جدید

## • انتقادات پیشنهادات:

کاربر میتواند انتقادات و پیشنهادات خود را برای مدیر ارسال کند، و مدیر آنها را در پنل وب مشاهده کند.



شکل ۱-۷: نمایی از بخش انتقادات و پیشنهادات در پنل وب

#### ۱-۴- بررسی سیستم های مشابه

از در دسترس ترین و نزدیک ترین سیستم ها به سیستم خبرنامه اما با امکاناتی ساده تر می توان به اپلیکیشن ها پیام رسان مانند تلگرام، واتس آپ و... اشاره کرد که با توجه به امکاناتی مانده ساخت گروه، کانال و... می تواند به عنوان خبررسان آن لاین استفاده شود. اما مزیت خبرنامه نسبت به این سیستم ها این است که سامانه جامع و منعطفی را در اختیار کاربران خود قرار می دهد که کاربران می توانند بر اساس نیاز های سازمان یا موسسه خود از این امکانات به صورت اختصاصی و در بستر امن بهره ببرند و همچنین در صورت نیاز به سفارشی سازی (افزودن امکانی خاص) این امکان برای ما فراهم است که آن را بر اساس نیاز هر مشتری خصوصی سازی کنیم.

همچنین ایجاد سیستم نقش دهی به اعضا این امکان را به مدیر سامانه می دهد که برای هر یک از گروه های کاربری به دلخواه خود دسترسی های لازم به سامانه را بدهد.

# فصل ۲

شرح فنی

## ۲-۱- زبان و تکنولوژی های استفاده شده در پروژه

زبان برنامه نویسی استفاده شده در بخش Back End پروژه، زبان پایتون می باشد اما برای پیاده سازی قسمت های مختلف سامانه از زبان ها و تکنولوژی های زیر استفاده شده است:

- Tornado Framework
- HTML
- JavaScript
- JQuery
- ACSS
- MySQLDB

همانطور که در بالا هم اشاره شد زبان مورد استفاده برای پیاده سازی بخش وب و وب سرویس های مورد نیاز، فریم . Tornado زبان پایتون می باشد. در اینجا به توضیح مختصری در باره این فریم ورک و همچنین دلیل انتخاب برای طراحی این سامانه می پردازیم.

تورنادو یک فریمورک توسعه وب، وب سرور و کتابخانه شبکه async می باشد که برای زبان پایتون توسط شرکت FriendFeed نوشته شده است. با استفاده از شیوه عدم-مسدودسازی ورودی/خروجی، تورنادو می تواند هزاران اتصال را مدیریت کند و کار را برای تکنولوژی هایی همچون وب سوکت که نیاز به یک اتصال طولانی مدت دارند آسان کند. در حال حاضر تورنادو بعد از فروخته شدن شرکت سازنده اش به فیسبوک، یکی از پروژه های این شرکت می باشد و به صورت متن باز منتشر شده است و جزوی از PyPI می باشد.

### نکات مثبت:

- تکنولوژی non-blocking
- مناسب برای پروژه های با بار ترافیکی بالا

### نکات منفی:

- جامعه کاربری کوچک تر نسبت به دیگر فریمورک ها
- مستندات ضعیف

با توجه به هسته سبک تورنادو و امکانات خوب آن برای مدیریت اتصال ها، این فریم ورک برای توسعه سامانه انتخاب شده است.

یک نمونه کد نوشته شده با فریم ورک تورنادو در ضمیمه ۱ آورده شده است.

## ۲-۲- توضیح پوشه ها و نحوه ساختار بندی پروژه

پروژه به طور کلی از پوشه ها و فایل های زیر تشکیل شده است:

- پوشه **WebApplication**:

که شامل کلاس ها و کدهای پروژه می باشد.

- پوشه **Static** :

که شامل فایل های عمومی پروژه مانند فایل های CSS,JS,plugins تصاویر آپلود شده توسط کاربران و... می باشد.

- فایل **shared\_connection.py**:

کلیه تنظیمات پروژه مانند اطلاعات دیتابیس،پورت های هر بخش و ... در این فایل پایتون نگه داری می شوند.

- فایل **web\_boot.py** :

فایل اصلی ران کننده سامانه این فایل است برای استفاده از سامانه این فایل باید در سرور دائما در حالت اجرا قرار داشته باشد.

حال نگاهی می اندازیم به محتویات پوشه **WebApplication**

این پوشه شامل زیر پوشه های به شرح زیر است:

- پوشه **Clases**:

که شامل کلاس های استفاده شده در سامانه می باشد. داخل این مسیر، پوشه ای به نام **premissions** هم موجود است که شامل کلیه کلاس ها و توابعی است که در موضوع سطوح دسترسی مورد استفاده قرار می گیرند.

- پوشه **db\_models** :

شامل کلاس های مدل سازی دیتابیس می باشند. درون فایل `table_model.py` همه جداول پایگاه داده مدل شده و داخل فایل های موجود در پوشه `models` برای ارتباط با هر جدول کلاس با پیشوند `Sys` در نظر گرفته شده اند که در `handler` ها برای برقراری ارتباط با دیتابیس از این مدل ها استفاده می شود تا از نوشتن مستقیم دستورهای ارتباط با دیتابیس در هندلرها جلوگیری شود.

- پوشه **Handlers** : شامل کلیه هندلر های استفاده شده در پروژه.

- پوشه **Template** : کلیه `view` ها یا همان صفحات `HTML` در این مسیر قرار می گیرند.

- پوشه **ui\_modules** : این پوشه شامل مدال (modules) های به کاررفته در سیستم می باشد که خود شامل دو پوشه `template modules` , می باشد که شامل فایل های مورد نیاز برای هر مدال می باشد.

- پوشه **urls** : بر طبق ساختار تورنادو `url` های به کاررفته در پروژه به تفکیک نقش کاربران در این فایل ها نوشته شده اند.



## ۲-۳- توضیح نحوه عملکردهای کدهای سامانه

برای دسترسی به کدهای هر بخش از پروژه ابتدا یک آدرس url یکی از متدهای هندلر را فراخوانی می کند که بسته به نوع کاربرد آن می تواند متدها مختلفی از هندلر را فراخوانی کند (به عنوان مثال post ، put، get و....). حال در این بخش توضیحات مربوط به کدهایی که با فراخوانی هر یک از url ها انجام می شود توضیح داده شده است.

### ۲-۳-۱- ورود

• Login/

• Get()

خروجی: نمایش صفحه ورود به حساب کاربری

• Post()

پارامترهای ورودی:

▪ username\_: نام کاربری

▪ pass\_: رمز عبور

### شرح کار:

پس از بررسی صحت رمز و پسورد وارد شده و بررسی نقش کاربر در سامانه در صورت درست بودن اطلاعات پس از ذخیره اطلاعات مورد نیاز مانند اطلاعات شناسایی کاربر، دسترسی های مجاز وی و ... در Session به داشبورد منتقل می شود و در صورت عدم احراز هویت کاربر در سامانه، خطای مناسب برای وی نمایش داده می شود.

## ۲-۳-۲- خروج

- Logout/
- Get(),Post()

### شرح کار:

با فراخوانی تابع `get_post()` کلیه Session های ذخیره شده برای کاربر حذف شده و امکان دسترسی به امکانات کاربری گرفته می شود (کاربر از سامانه خارج شده)

## ۲-۳-۳- دریافت لیست شهرها

- GetCities/

برای دریافت فهرست شهرهای هر استان استفاده می شود.

- Post()

پارامتر ورودی:

▪ Pid: آی دی استان مورد نظر برای درصافت لیست شهرها

خروجی:

▪ لیست شهر های استان وارد شده

## ۲-۳-۴- مشاهده پلن های مالی

Plans/(id)/Orders/ •

برای مشاهده اطلاعات مربوط به پلن های مالی

Get() •

پارامتر ورودی:

- Id: آی دی پلن مربوطه
- Step: مرحله جاری در هنگام خرید پلن

خروجی:

صفحه مورد نظر برای نمایش اطلاعات

Post() •

با توجه به مرحله ای که کاربر در آن قرار دارد اطلاعات وارد شده را پردازش می کند.

پارامترهای ورودی:

- Plan: پلن انتخابی کاربر
- Username: نام کاربری
- Password: کلمه عبور
- Province: ای دی استان کاربر
- City: آی دی شهر کاربر
- visitor\_id: ای دی بازاریاب (کسی که از کاربر برای عضویت در سامانه دعوت کرده است)
- first\_name: نام کاربر
- last\_name: نام خانوادگی کاربر
- Phone: تلفن تماس کاربر
- school\_name: نام موسسه کاربر که برای آن پلن تهیه می کند
- Address: آدرس پستی موسسه کاربر
- Grade: مقطع تحصیلی موسسه (در صورت وجود)
- Email: پست الکترونیک کاربر

## شرح کار:

پس از اعتبار سنجی پارامترهایی که توسط کاربر وارد شده در صورت تایید اعتبار مشخصات کاربر در سامانه درج می شود و همچنین کد فعال سازی برای وی به صورت ایمیل ارسال می گردد. در صورتی هم که کاربر توسط بازاریابی دعوت شده باشد (کد بازاریاب در هنگام ثبت نام درج شده باشد)، اطلاعات مرتبط به ثبت نام وی برای بازاریاب مربوطه نیز در سامانه درج می شود.

### ۲-۳-۵- نمایش داشبورد مدیریت

- System/Dashboard/

برای مشاهده داشبورد کاربری مورد استفاده قرار می گیرد.

- Get()

اطلاعات لازم برای نمایش در صفحه داشبورد شامل پیام ها، پیشنهادها، نظرات و همینطور بازدیدها از پایگاه داده استخراج شده و در صفحه مورد نظر نمایش داده می شود.

### ۲-۳-۶- مدیریت نقش های کاربری

- System/Roles/

برای مشاهده و مدیریت نقش های کاربری سیستم

- Get()

کلیه نقش های موجود در سامانه را از پایگاه داده استخراج توسط متد `get_all()` کلاس `(SysRoles)` و در صفحه مربوطه نمایش می دهد.

- Post()

پارامترهای ورودی:

▪ :Method

فیلد کنترلی برای مشخص کردن کاری که باید انجام شود که شامل دو مقدار `save` برای ثبت نقش جدید و `update` برای به روز رسانی اطلاعات یک نقش همچنین `delete` برای حذف یکی از نقش های موجود به کار می رود.

- Role\_name : نام نقش در صورت درخواست افزودن نقش جدید
- cid: آی دی نقش در صورت درخواست حذف یا آپدیت
- val\_: نام جدید نقش در صورت درخواست ویرایش یکی از نقش های موجود.

## شرح کار:

با توجه به متد انتخابی نقش ، پس از اعتبار سنجی فیلد های ورودی در صورت نیاز وبا بررسی اینکه کاربر جاری دسترسی به انجام فرایند انتخابی را داشته باشد، عملیات انجام می شود.

## ۲-۳-۷- مشاهده مجموعه ها

- System/Collections/

مشاهده مجموعه های تعریف شده در سیستم

- Get()

مجموعه های تعریف شده توسط متد () get\_js\_tree1 کلاس ScopeTree از پایگاه داده استخراج می شود.

- Post()

برای نمایش مجموعه های سیستم های دیگر به کار می رود.

پارامترهای ورودی:

- Pid : آی دی سیستم مربوطه

خروجی:

نمایش کلیه مجموعه های سیستم وارد شده

## ۲-۳-۸- جزئیات مجموعه های تعریف شده

• System/Collections/Action/

دسترسی به جزئیات هر کدام از مجموعه های تعریف شده در سیستم

• Get()

پارامترهای ورودی:

- Action: فیلد کنترلی برای مشخص شدن نوع صفحه مورد جستجو شامل مقادیر add، برای افزودن مجموعه جدید به سیستم، edit، برای یکی ویرایش مجموعه های موجود (در صورت دسترسی)، showUser برای نمایش کاربران یک مجموعه.

▪ \_id: آی دی مجموعه موردنظر در صورت نیاز.

### شرح کار:

با توجه به action وارد شده صفحه مورد نظر برای کاربر نمایش داده می شود.

• Post()

پارامترهای ورودی

- Method: شامل مقادیر add برای افزودن مجموعه جدید به سیستم، برای ویرایش مجموعه های موجود، delete هم برای حذف مجموعه موجود.
- coll\_name: نام مجموعه
- Parent: نام مجموعه والد در صورت وجود.
- collection\_pic: تصویر مجموعه مورد نظر

### شرح کار:

با توجه به متد ارسالی و پس از اعتبارسنجی پارامترها، تغییرات مورد نظر با فراخوانی توابع کلاس SysCollections اجرا می شود.

## ۲-۳-۹- اختصاص مجموعه های مجاز هر نقش

• System/Roles/AccessDefaultCollection/

اختصاص مجموعه های مجاز برای هر نقش

• Get()

خروجی: لیستی از کلیه نقش های موجود در سیستم

### شرح کار:

کلیه نقش های سیستم توسط متد `get_all()` کلاس `SysRoles()` از پایگاه داده استخراج می شوند.

• Put():

ورودی:

`Role_id`: ای دی نقش مربوطه

خروجی:

لیستی از مجموعه های اختصاص داده شده به نقش کاربری انتخاب شده

### شرح کار:

لیستی شامل کل مجموعه ای مجاز و همچنین مجموعه های فعلی نقش کاربری مورد نظر توسط متد `get_js_tree()` از کلاس `ScopeTree()` از پایگاه داده استخراج می شود.

• Post()

انتساب مجموعه به نقش های سامانه

پارامتر های ورودی:

▪ `role_id`: ای دی نقش مورد نظر

## ▪ Collections :

آرایه ای از مجموعه های انتخابی برای  
انتساب به نقش

### شرح کار:

نقش های ارسالی توسط متد `add_role_collections()` کلاس `SysCollectionRoles()` به نقش مورد نظر انتساب می یابند.

## ۲-۳-۱۰- انتساب مجموعه های پیش فرض هر نقش

### • System/Roles/DefaultCollection

برای انتساب مجموعه های پیش فرض برای هر کدام از نقش های کاربری

### • Get()

با استفاده از متد `get_all()` کلاس `SysRoles()` کلیه نقش های کاربری از پایگاه داده استخراج و به صفحه ارسال می شود.

### • put() :

ورودی:

### ▪ Role\_id : ای دی نقش مربوطه

خروجی:

لیستی از مجموعه های پیش فرض اختصاص داده شده به نقش کاربری انتخاب شده

### شرح کار:

لیستی شامل کل مجموعه های مجاز و همچنین مجموعه های پیش فرض فعلی نقش کاربری مورد نظر توسط متد `get_js_tree()` از کلاس `ScopeTree()` از پایگاه داده استخراج می شود.



- **Post() :**

پارامترهای ورودی

- **Role\_id :** آی دی نقش مورد نظر

- **Collections :**

آرایه ای از مجموعه های پیش فرض انتخابی برای انتساب به نقش نقش های ارسالی توسط متد `change_default_collections()` کلاس `SysCollectionRoles()` به نقش مورد نظر انتصاب می یابند.

## شرح کار:

لیست مجموعه های ارسالی به نقش مورد نظر اختصاص داده می شود.

## ۲-۳-۱۱- ویرایش مجوزهای کاربران

- **System/Permissions/**

ویرایش مجوزهای کاربری کاربران

- **Get()**

نمایش صفحه مورد نظر

- **Put()**

پارامترها:

- **Username :** نام کاربری کاربر مورد نظر

خروجی:

لیستی از مجوز های مجاز تعریف شده در سامانه به همراه مجوز هایی که به این کاربر اختصاص یافته شده اند.

## شرح کار:

پس از اعتبارسنجی های مورد نیاز، خروجی مورد نیاز با استفاده از متد `get_all_user_permissions()` کلاس `SystemPermissions` استخراج و به خروجی برده می شود.

• `Post()`

پارامترهای ورودی:

- `Username` : نام کاربری کاربر مورد نظر
- `Permissions` : لیستی از مجوز های انتخاب شده برای کاربر مورد نظر

## شرح کار:

پس از اعتبارسنجی پارامترهای ورودی مجوز های مورد نظر به استفاده از متد `add_user_permission()` کلاس `SysPermissions` به کاربر اختصاص داده می شود.

## ۲-۳-۱۲- مدیریت نقش های کاربران

• `System/UserRolesManagement/Roles/`

مدیریت نقش های کاربران

• `Get()`

ابتدا توسط متد `get_all_sub_system()` کلاس `SysSystem` کلیه زیر سیستم ها استخراج شده و پس از آن به صفحه مربوطه ارسال می شود.

• `put ()`

پارامترهای ورودی:

- `Username` : نام کاربری
- `Sys_id` : آی دی سیستم مربوطه

## شرح کار:

پس از اعتبار سنجی ورودی ها توسط متد `get_js_tree()` کلاس `SysRoles` کلیه نقش های سیستم و همچنین نقش کاربری جاری در سیستم در پایگاه داده استخراج می شود.

• `Post()`

پارامترهای ورودی:

- `Username`: نام کاربری کاربر مربوطه
- `Roles`: لیستی از نقش های انتخابی برای کاربر
- `sys_id`: آی دی زیرسیستم انتخابی

## شرح کار:

پس از اعتبار سنجی پارامترهای ورودی توسط متد `add_user_roles_2()` کلاس `SysUserRoles` نقش (ها) انتخابی به کاربر مربوطه اختصاص داده می شود. همچنین در صورت لزوم (انتخاب نقش ویرایشگر برای کاربر) تنظیمات مربوط به ویرایشگر ها هم توسط متد `add_setting()` کلاس `SysEditorsSetting` به روز رسانی می شود.

## ۲-۳-۱۳- افزودن کاربر به عنوان نویسنده

• `System/UserRolesManagement/AddEditors/`

افزودن کاربر به عنوان نویسنده به سیستم

• `Get()`

نمایش صفحه مورد نظر در خروجی

• `Put()`

پارامترهای ورودی:

- `Username`: نام کاربری کاربر مربوطه

## شرح کار:

پس از اعتبار سنجی وودی ها و بررسی اینکه نقش کاربری کاربر انتخابی ویراشگر باشد (متد `is_editor()` کلاس `SysUserRoles`)، توسط متد `get_js_tree()` کلاس `ScopeTree` مجموعه های موجود در سیستم و مجموعه هایی که کاربر به آنها دسترسی دارد (در صورت وجود) به خروجی برده می شود.

• `Post()`

پارامترهای ورودی:

- `Username`: نام کاربری کاربر مربوطه
- `Collections`: لیستی از مجموعه های انتخابی برای نویسنده
- `e1`: مشخص کردن امکان ویرایش اخبار
- `e2`: مشخص کردن امکان انتشار مستقیم اخبار

## شرح کار:

پس از اعتبارسنجی پارامترهای ورودی، توسط متد `add_editors_collection()` کلاس `SysEditorsCollections` مجموعه های مورد نظر به نویسنده اختصاص داده می شود و پس از آن تنظیمات مربوط به نویسندگان به روزرسانی می شود. (متد `update_setting()` کلاس `SysEditorsSetting`)

## ۲-۳-۱۴- افزودن خبر جدید

• `System/News/Add/`

افزودن خبر جدید

• `Get()`

پس از استخراج کلیه متون و همچنین مجموعه های مربوطه از پایگاه داده، صفحه مربوط به افزودن پست (خبر) جدید به کاربر نمایش داده می شود.

• `Post()`

پارامترهای ورودی:

- news\_title : تیتر خبر
- full\_text : متن خبر وارد شده
- type\_ : نوع خبر
- Public : نشان دهنده عمومی بودن خبر
- direct\_publish : امکان انتشار مسقیم خبر
- Collections : مجموعه های مربوط به خبر
- Pic : تصویر مربوط به خبر
- ink\_name : نام لینک مربوط به خبر
- link\_address : آدرس لینک مربوط به خبر
- random\_id : شناسه تصادفی خبر
- file\_name : نام فایل مربوط به خبر
- attach\_file : فایل مربوط به خبر

### شرح کار:

پس از اعتبار سنجی پارامترهای ورودی ابتدا تصویر خبر توسط متد `upload_photo()` در سامانه آپلود می شود پس از آن با همین متد فایل های مربوط به خبر آپلود می شود. توسط تابع `add_news()` کلاس `SysNews` خبر در سامانه درج می شود. همچنین تگ های انتخابی توسط متد `add_many()` کلاس `SysTags` در پایگاه داده ذخیره می شود. تصاویر مربوط به خبر توسط تابع `add_one()` کلاس `SysNewsPic` آپلود میشود. متد `add_one()` از کلاس `SysNewsFile` فایل ها را آپلود و متد `add_one()` کلاس `SysNewsLink` لینک ها را در پایگاه داده ذخیره می کند. مجموعه های مربوط به خبر توسط متد `add_many()` کلاس `SysNewsCollections` به خبر اضافه میشود و پس از آن در صورت نیاز از طریق فراخوانی تابع `get_all_users_for_news()` کلاس `SysGcmUsers` به کاربران ارسال میشود.

## ۲-۳-۱۵- نمایش اخبار سیستم

• System/News/Management/

نمایش اخبار سیستم

• Get()

پارامترهای ورودی:

پس از استخراج اخبار و هم چنین لیست سیستم ها و مجموعه ها و اطلاعات مورد نیاز، صفحه مربوط به نمایش اخبار به کاربر نمایش داده می شود.

• Put()

متد مشخص کننده نوع عملیات که شامل مقدار delete برای حذف مقادیر.

▪ cid: آی دی پست مورد نظر

### شرح کار:

پس از اعتبارسنجی پارامترهای ورودی، خبر هم چنین اخبار مربوط به آن (در صورت وجود) از سامانه حذف می شوند. توسط متد delete\_subnews() کلاس SysNews پس از آن خود خبر با استفاده از متد delete\_news() کلاس SysNews از سامانه حذف می شود.

## ۲-۳-۱۶- گرفتن مجموعه های یک سیستم

• /GetCollections

برای گرفتن مجموعه های یک سیستم

• Post()

پارامترهای ورودی:

▪ Pid: آی دی سیستم مربوطه (در صورت لزوم)

## شرح کار:

کلیه مجموعه های سیستم مورد نظر با استفاده از متد `get_all()` کلاس `SysCollections` از پایگاه داده استخراج و به خروجی ارسال می شود.

### ۲-۳-۱۷- ویرایش خبر

• `/System/News/Edit/`

ویرایش یک خبر

• `Get()`

پارامترها:

`Id` : ای دی خبر مورد نظر برای ویرایش

## شرح کار:

بعد از اعتبارسنجی پارامتر ورودی و همچنین استخراج موارد مورد نیاز برای نمایش در صفحه، صفحه ویرایش خبر برای کاربر ارسال میشود.

• `Post()`

شبه کاری که در بخش افزودن خبر استفاده می شود در اینجا هم اتفاق می افتد با این تفاوت که اطلاعات جدید با استفاده از متد `update_news()` کلاس `SysNews` در سامانه به روز رسانی می شود.

### ۲-۳-۱۸- نمایش اخبار در انتظار تأیید

• `System/News/PendingNews/`

نمایش خبر های در انتظار تایید

• `Get()`

پارامترهای ورودی:

▪ Page

برای نمایش خبرهای به صورت صفحه بندی شده

## شرح کار:

بر اساس سطح دسترسی کاربر جاری، اخباری که در منظر تایید هستن توسط متد `get_news_p4` کلاس `SysNews` به صفحه مورد نظر ارسال می شود.

• `Post()`

پارامترهای ورودی:

▪ `Method`: پارامتر کنترلی مشخص کننده نوع عملیاتی که باید انجام شود و شامل مقادیر `serach` برای جستجو بین خبرهای تایید نشده و `accept` برای تایید خبرها می باشد.

▪ `Search`: کلید واژه برای جستجو در میان خبرها (در صورت جستجو)

▪ `Subsystem`: آی دی زیر سیستم مربوطه در صورت نیاز

▪ `cid`: آی دی خبر مورد نظر (در صورت تایید خبر)

## شرح کار:

در صورتی که `method = search`

پس از اعتبارسنجی ورودی ها و بررسی سطوح دسترسی کاربر فعلی، نتایج جستجو با استفاده از متد `get_news_by_search_p4()` کلاس `sys_n` از پایگاه داده استخراج و به خروجی برده می شود.

در صورتی که `method=accept`

در اینجا نیز پس از اعتبارسنجی ورودی ها و بررسی سطوح دسترسی، خبر انتخابی با استفاده از متد `update_status_news()` کلاس `SysNews` تایید می شود.



## ۲-۳-۱۹- افزودن کاربر جدید

• System/UsersManagement/AddUsers/

افزودن کاربر جدید به سامانه

• Get()

پس از استخراج نقش های سیستم و گروه های کاربری از پایگاه داده توسط متدهای get\_all() و get\_js\_tree() از کلاس SysRoles صفحه مربوط به افزودن کاربر نمایش داده می شود.

• Put()

پارامترهای ورودی:

▪ U: نام کاربری وارد شده (شماره تلفن همراه)

### شرح کار:

با استفاده از متد is\_exist() کلاس SysUsers بررسی می شود که کاربری از قبل با نام کاربری وارد شده در سامانه وجود دارد یا خیر.

• Post

پارامترهای ورودی:

- Method : پارامتر کنترلی دارای دو مقدار save برای افزودن یک کاربر و group\_save برای افزودن گروهی از کاربران به صورت یکجا به سامانه از طریق آپلود فایل اکسل
- Fname: نام کاربر
- Lname: نام خانوادگی کاربر
- Username: نام کاربری کاربر
- Password: کلمه عبور کاربر
- conf\_password: تایید کلمه عبور کاربر
- Role: لیستی از نقش های انتخابی برای کاربر

- Inp: فایل اکسل آپلود شده توسط کاربر برای درج گروهی کاربران (در صورتی که method انتخابی group\_save باشد).

## شرح کار:

اگر method = save

پس از اعتبار سنجی پارامترهای ورودی کاربر، ابتدا توسط متد `is_exist()` کلاس `SysUsers` بررسی می شود که کاربری از قبل با این نام کاربری در سامانه وجود نداشته باشد. پس از آن اطلاعات کاربر با استفاده از متد `add_user()` کلاس `SysUsers` و اطلاعات مربوط به نقش(ها) انتخابی برای کاربر در سامانه با متد `add_user_roles()` کلاس `SysUserRoles` و همچنین اطلاعات مربوط به مجموعه های قابل دسترس برای کاربر از طریق متد `set_user_collections()` کلاس `SysUserCollections` در سامانه درج می شود.

اگر method = group\_save

درج تک تک کاربران همانند آنجا در بالا گفته شد انجام می شود با این تفاوت که در اینجا اطلاعات شناسایی تک تک کاربران از فایل اکسل آپلود شده توسط کاربر، خوانده می شود.

## ۲-۳-۲۰- مشاهده و مدیریت کاربران سامانه

• `System/UsersManagement/Users/`

مدیریت کاربران سامانه

• `Get()`

پس از دریافت تمامی زیر سیستم ها و همچنین نقش های کاربری از سیستم، کلیه کاربران سیستم جاری با استفاده از متد `get_all()` کلاس `SysUsers` از پایگاه داده استخراج و به صفحه برده می شود.

• `Put()`

پارامترهای ورودی:

- Method: پارامتر کنترلی که مشخص کننده نوع داده خروجی می باشد و دارای دو مقدار `Get` برای جستجوی کاربران با نام کاربری و `delete` برای حذف کاربر انتخاب شده از سامانه می باشد.

▪ sys\_id

آی دی زیر سیستم مورد نظر در صورت نیاز.

### شرح کار:

اگر method = get

کاربر مورد نظر با استفاده از متد is\_exist() کلاس SysUsers در سامانه جستجو شده و در صورت وجود داشتن اطلاعات وی به خروجی منتقل می شود.

اگر method = delete

پس از اعتبارسنجی های مورد نیاز، کاربر مورد نظر توسط متد delete\_user() کلاس SysUsers از سامانه حذف می شود.

• Post()

پارامترهای ورودی:

- Page: صفحه مورد نظر برای نمایش گزارش (در صورت لزوم)
- Fn: نام کاربر مورد نظر
- Ln: نام خانوادگی کاربر مورد نظر
- Username: نام کاربری کاربر مورد نظر
- Role: لیستی از نقش (های) انتخابی برای کاربر
- Sort: نوع مرتب سازی داده های گزارش (صعودی/ نزولی)
- Subsystem: آی دی زیر سیستم مورد نظر (در صورت لزوم)

### شرح کار:

بسته به نوع فیلتر گزارش در خواستی کاربر اطلاعات مورد نظر با استفاده از متد های get\_all() و یا get\_all\_by\_role() از کلاس SysUsers از سامانه دریافت و به خروجی برده می شوند.

## ۲-۳-۲۱- گرفتن نقش ها

- /GetRoles

گرفتن نقش ها

پارامتر ورودی:

▪ Pid: آی دی سیستم مورد نظر در صورت لزوم

خروجی:

کلیه کاربران و نقش های کاربری سیستم جاری

## شرح کار:

کلیه نقش های موجود در سیستم مورد نظر با استفاده از متد `get_all()` کلاس `SysRoles` و همچنین کلیه کاربران سیستم مورد نظر با استفاده از متد `get_all()` کلاس `SysUsers` از پایگاه داده دریافت و به خروجی برده می شوند.

## ۲-۳-۲۲- افزودن کاربر مهمان

- System/ManageUserGuest/

افزودن کاربر مهمان به سامانه

- Get()

با استفاده از متد `get_guest_users()` کلاس `SysUsers` کلیه کاربران مهمان موجود در سامانه به همراه لیست همه زیر سیستم ها موجود که با استفاده از متد `get_all_sub_system()` کلاس `SysSystem` گرفته می شود، به خروجی ارسال می شود.

- Post()

پارامترهای ورودی:

▪ guest\_name: نام کاربر

▪ Subsystems: لیستی از سیستم(ها) انتخابی برای کاربران

## شرح کار:

پس از اعتبارسنجی پارامترهای ارسالی، اطلاعات توسط متد `add_user()` کلاس `SysUsers` در جدول اطلاعات کاربران و همچنین اطلاعات مربوط به نقش جدید توسط متد `add_user_roles2()` کلاس `SysUserRoles` به سامانه اضافه می شود.

### • Put()

حذف کاربر مهمان از لیست کاربران

پارامترهای ورودی:

▪ `cid`: آی دی کاربر مورد نظر برای حذف از سیستم

## شرح کار:

کاربر انتخابی با استفاده از متد `delete_user()` کلاس `SysUsers` حذف می شود.

## ۲-۳-۲۳- نمایش راهنمای سیستم

### • System/Help/

نمایش راهنمای سیستم

### • Get()

پارامترهای ورودی:

▪ `_id`: آی دی بخش مورد نظر برای مشاهده راهنمای آن بخش

## شرح کار:

با توجه به آی دی ارسالی (در صورت ارسال) صفحه راهنمای مورد نظر برای کاربر نمایش داده می شود.

## ۲-۳-۲۴- نمایش لیست انتقادات و پیشنهادات

• System/Suggestions/

نمایش انتقادات و پیشنهادات ثبت شده

• Get()

بر اساس دسترسی کاربر مورد نظر اطلاعات مربوط با استفاده از متد `get_all_p4()` یا `get_all()` کلاس `SysSuggestions` به خروجی ارسال می شود.

• Put()

پارامترها:

- `id_`: آی دی پیام مورد نظر
- `Method`: پارامتر کنترلی که نوع عملیات را مشخص می کند و شامل دو مقدار `get` برای مشاهده برای مشاهده کامل پیام و `delete` برای حذف پیام از سامانه می باشد.

### شرح کار:

پس از اعتبار سنجی های مورد نیاز اگر `get = method` باشد پیام انتخاب شده با استفاده از متد `update_p4()` کلاس `SysSuggestions` به حالت خوانده شده در می آید و اگر `method=delete` پیام انتخاب شده از سامانه با استفاده از متد `delete_p4()` همان کلاس حذف می گردد.

• Post()

پارامترهای ورودی:

▪ Search

کلیدواژه مورد نظر برای جستجو

▪ Subsystem

ای دی سیستم مورد نظر در صورت لزوم

## شرح کار:

با توجه به سیستم انتخاب شده برای نمایش گزارش، گزارش مربوطه با استفاده از متد `get_all()` از کلاس `SysSuggestions` از پایگاه داده استخراج و به خروجی برده می شود.

## ۲-۳-۲۵- ارسال پیام به کاربران

- `System/Messages/SendNew/`

ارسال پیام به کاربران سیستم

- `Get()`

پس از دریافت لیست کاربران سامانه با استفاده از متد `get_all()` کلاس `SysUsers` و همچنین در صورت نیاز لیست همه کاربران زیر سیستم توسط متد `get_all_users_subsystem` کلاس `SysUsers` و همچنین لیست همه مجموعه ها با متد `get_all()` کلاس `SysCollections` صفحه مورد نظر برای کاربر به نمایش در می آید.

- `Post()`

پارامترهای ورودی:

- `Text`: متن پیام
- `Receivers`: لیستی از آی دی کاربران دریافت کننده پیام

## شرح کار:

پیام(ها) با استفاده از متد `send_many()` کلاس `SysMessages` به کاربر(ان) مربوطه ارسال می گردد.

- `Put()`

دریافت کلید کاربران یک مجموعه خاص

پارامترهای ورودی

- `col_id`: آی دی مجموعه مورد نظر

## شرح کار:

لیست کاربران مجموعه مورد نظر با استفاده از متد `get_all_by_collection()` کلاس `SysUsers` از پایگاه داده استخراج و به خروجی برده می شود.

## ۲-۳-۲۶- مشاهده پیام های دریافتی

- `System/Messages/Received/`

مشاهده پیام های دریافت شده

- `Get()`

پیام های جدید و هم چنین پیام های خوانده دشه با استفاده از متد `get_all_for_system_p4` کلاس `SysMessages` از سیستم استخراج شده و به کاربر نمایش داده می شود.

- `Put()`

پارامترهای ورودی

- `_id`: آی دی پیام مورد نظر

- `Method`: پارامتر کنترلی که نوع عملیات رو مشخص می کند و داری مقادیر `get` برای مشاهده پیام، `delete` برای حذف پیام و `reply` برای ثبت پاسخی برای پیام می باشد.

- `Text`: متن پیام برای ارسال (در صورت لزوم)

## شرح کار:

بسته به سطح دسترسی کاربر جاری در سامنه که از روی پارامتر `current_plan_id` مشخص می شود به شرح زیر پیام های مناسب برای کاربر ارسال می شود.:

اگر `method=get`

پیام انتخاب شده با استفاده از متد `get_one()` از کلاس `SysMessages` از سیستم استخراج شده و پس از آن پیام به حالت خواننده شده در می آید و به همراه کلیه پیام های قبلی کاربر به خروجی برده می شود.



method = delete اگر

در این صورت پیام با آی دی مشخص شده توسط متد delete\_msg\_p4 کلاس SysMessages از پایگاه داده سیستم حذف می شود.

method=reply اگر

پیام جدید با استفاده از متد send() کلاس SysMessages در پاسخ به پیام انتخاب شده در سامانه درج می شود.

## ۲-۳-۲۷- لیست پیام های ارسال شده

• System/Messages/Sent/

فهرست پیام های ارسال شده

• Get()

کلیه پیام های ارسالی کاربر جاری با استفاده از متد get\_all\_sent\_for\_system\_p4() کلاس SysMessages از سامانه استخراج شده و به همراه لیست کلیه زیرسیستم های موجود که با استفاده از متد get\_all\_sub\_system() کلاس SysSystem به دست آمده، به خروجی برده می شود.

• Post()

پارامترهای ورودی:

- Search : آی دی زیر سیستم مورد نظر برای مشاهده لیست پیام های ارسالی
- Subsystem : آی دی زیر سیستم مورد نظر در صورت لزوم

## شرح کار:

پیام های ارسالی مورد نظر با استفاده از متد get\_all\_sent\_for\_system() کلاس SysMessages استخراج و همراه لیست زیر سیستم های موجود به خروجی برده می شود.

## ۲-۳-۲- مشاهده نظرات خبرهای ثبت شده

• System/Comments/

مشاهده نظرات ثبت شده توسط کاربران

• Get()

با توجه به سزح دسترسی کاربر، ابتدا کلیه پیام های جدید با استفاده از متد `get_all()` کلاس `SysNewsComments` از سیستم گرفته می شود و پس از آن نظرات قبلی کاربران توسط همان متد گرفته می شود و به همراه لیست کلیه زیر سیستم های موجود در سامانه به خروجی برده می شود.

• Post()

پارامترهای ورودی:

- Search: پارامتر کنترلی که اگر دارای مقدار Search باشد مشخص کننده این است که باید کامنت ها بر اساس زیرسیستم فیلتر شوند.
- Subsystem: آی دی زیرسیستم مورد نظر برای فیلتر کامنت ها (در صورت لزوم)
- Method: پارامتر کنترلی که در صورت اینکه عملیات مورد نظر جستجو نباشد مشخص کننده نوع عملیات است که دارای مقادیر `delete` برای حذف پیام، `accept` برای تایید یک کامنت می باشد.
- \_id: آی دی پیام مورد نظر در صورت لزوم

### شرح کار:

اگر `search = search`

کامنت های جدید و همچنین کامنت های قبلی زیر سیستم مورد نظر با استفاده از متد `get_all` کلاس `SysNewsComments` از پایگاه داده گرفته و به خروجی برده می شود.

در غیر این صورت براساس مقدار پارامتر `method` و همچنین سطح دسترسی کاربر، سایر عملیات انجام می شود به این شکل که اگر `method = delete` باشد، کامنت مورد نظر با استفاده از متد `delete_by_id_p4()` کلاس `SysNewsComments` کامنت مورد نظر از سیستم حذف می شود.

اگر هم `method = accept`

کامنت مورد نظر با استفاده از متد `update_p4()` کلاس `SysNewsComments` به حالت تایید شده در می آید.

## ۲-۳-۲۹- ایجاد نظر سنجی جدید

• `System/Poll/Insert/`

ایجاد نظر سنجی جدید

• `Get()`

کلیه زیر سیستم های موجود سیستم فعلی از پایگاه داده استخراج شده و در صفحه افزودن نظر سنجی به کاربر نشان داده می شود.

• `Post()`

پارامترهای ورودی:

- `Item` : لیست گزینه های نظر سنجی
- `Text`: متن نظر سنجی
- `status_poll`: پارامتر کنترلی که فعال یا فعال نبودن نظر سنجی را مشخص می کند
- `Subsystems`: لیست زیر سیستم های مربوط به نظر سنجی را مشخص می کند.

## شرح کار:

ابتدا اطلاعات مربوط به نظر سنجی با استفاده از متد `insert()` کلاس `SysPollQuestion` در سامانه درج می شود. و پس از اگر حالت نظر سنجی غیر فعال باشد با استفاده از متد `update_status()` کلاس `SysPollQuestion` مشخص می شود و پس از آن گزینه های نظر سنجی با فرخوانی تابع `insert_list_item()` کلاس `SysPoolItem` در دیتابیس سامانه درج می شود.

## ۲-۳-۳۰- مشاهده نتایج نظرسنجی ها

• System/Poll/Result/

مشاهده نتایج نظر سنجی ها

• Get()

بسته به نوع دسترسی کاربر جاری اطلاعات مربوط به نظرسنجی ها با استفاده از متد `get_all_p4()` و یا `get_all()` از کلاس `SysPollQuestion` از پایگاه داده استخراج و به خروجی برده می شود.

• Post()

پارامترهای ورودی:

- Search: پارامتر کنترلی که اگر دارای مقدار Search باشد بیانگر این است که باید لیست نظرسنجی ها فیلتر شود.
- Subsystem: آی دی زیر سیستم مورد نظر در صورت لزوم
- cid: آی دی نظر سنجی مورد نظر در صورت لزوم
- Method: پارامتر کنترلی مشخص کننده نوع عملیات و دارای مقادیر show برای نمایش اطلاعات یک نظرسنجی و delete برای حذف یک نظرسنجی از سامانه می باشد.

## شرح کار:

در صورتی که پارامتر `search=search`

با استفاده از متد `get_all_p4()` از کلاس `SysPollQuestion` کلیه نظر سنجی های مربوط به زیر سیستم مورد نظر برای کاربر به نمایش در می آید و به همراه لیست کلیه زیر سیستم های موجود به خروجی برده میشود.

در غیر این صورت و بسته به مقدار پارامتر `method` و همچنین سطح دسترسی کاربر جاری اگر `method = show` باشد اطلاعات مربوط به نظر سنجی با استفاده از متد `get_all()` کلاس `SysPoolItem` از پایگاه داده استخراج می شود و اطلاعات مربوط به پاسخ های سایر کاربران به این

نظر سنجی هم با استفاده از متد `get_all_p4()` کلاس `SysPollAnswer` گرفته می شود و به خروجی برده می شود.

اگر `method = delete`

با استفاده از متد `delete()` کلاس `SysPollQuestion` نظر سنجی مورد نظر از سیستم حذف می شود.

## ۲-۳-۳۱- ارسال تیکت جدید

• `System/Support/SendNewTicket/`

ارسال تیکت جدید

• `Get()`

صفحه مربوط به ارسال تیکت نمایش داده می شود.

• `Post()`

پارامترهای ورودی:

- `Topic`: عنوان تیکت وارد شده
- `full_text`: متن تیکت
- `Attachment`: فایل آپلودی توسط کاربر در صورت وجود

## شرح کار:

اطلاعات مربوط به تیکت با استفاده از متد `send()` کلاس `SysTickets` و همچنین متد `send_subticket()` کلاس `SysSubticket` در سامانه درج می شود.

## ۲-۳-۳۲- نمایش تمام تیکت های سیستم

- System/Support/MyTicket/  
Get()

کلیه تیکت های کاربر جاری با استفاده از متد `getall()` کلاس `SysTickets` از سامانه گرفته و به کاربر نمایش داده می شود.

## ۲-۳-۳۳- نمایش یک تیکت

- System/Support/MyTickets/Show/  
Get()

پارامترهای ورودی

- `_id`

### شرح کار:

ابتدا با استفاده از متد `is_exist()` کلاس `SysTickets` بررسی می شود که آیا تیکتی با مشخصات وارد شده در سیستم وجود دارد و سپس در صورت وجود، اطلاعات تیکت با استفاده از `get_all()` کلاس `SysSubticket` استخراج می شود و به خروجی برده می شود.

- Post()

برای ارسال پیام در یک تیکت به کار می رود.

پارامترهای ورودی:

- `_id` : آی دی تیکت مربوطه
- `full_text` : متن پیام ارسالی

### شرح کار:

اطلاعات مربوط به پیام جدید در بخش تیکت ها با استفاده از `send_subticket()` متد از کلاس `SysSubticket` و همچنین `change_status()` کلاس `SysTickets` در سیستم ذخیره می شود.

## ۲-۳-۳۴- تنظیمات اعلان ها

• System/Settings/SubDomain/

تنظیمات مربوط به اعلان ها

• Get()

تنظیمات مربوط به اعلان ها با استفاده از متد `get_all()` در کلاس `SysNotificationByType` از دیتابیس استخراج شده و به کاربر نمایش داده می شود.

• Post()

پارامترهای ورودی

▪ Vals

لیستی از تنظیمات اعمال شده توسط کاربر

## شرح کار:

تک تک تنظیمات جدید ارسال شده توسط متد `update_` کلاس `SysNotificationByType` در سامانه به روز رسانی می شوند.

# فصل ۳

شرح توابع کار با دیتابیس  
سامانه



### ۳-۱ مقدمه

برای ارتباط به هر یک از جداول پایگاه داده کلاسی مجزا با پیشوند sys نوشته شده است که از ارتباط مستقیم با دیتابیس در هندلر ها جلوگیری شود، در این بخش به شرح کار هر یک از متدهای این کلاس ها می پردازیم.

توابع کار با دیتابیس به شرح زیر می باشند:

- SysProvince

برای ارتباط با جدول province

متدها:

- get\_all()

### شرح کار:

لیست کلیه استان ها را بر می گرداند.

- SysCity()

برای ارتباط با جدول City به کار می رود.

متدها:

- get\_all () : کل شهر های استان مورد نظر را بر می گرداند.

- SysSystem

برای ارتباط با جدول System

متدها:

- get\_children() : کلیه زیر سیستم های مربوط به سیستم مورد نظر را بر می گرداند.

- add\_system() : افزودن یک سیستم به سامانه

- update\_system1() : برای تغییر وضعیت سیستم مورد استفاده قرار می گیرد.

- get\_all() : دریافت کل سیستم های سامانه

- `get_all_sub_system()`: گرفتن کلیه زیر سیستم های یک سیستم
- `get_all_deactive()`: گرفتن کلیه سیستم های غیر فعال
- `get_parent()`: گرفتن آی دی سیستم اصلی یک زیر سیستم
- `get_one()`: برگرداندن اطلاعات یک سیستم خاص
- `get_all_data_of_one_sub_system()`: گرفتن اطلاعات یک زیر سیستم خاص
- `delete_system()`: حذف یکی سیستم از سامانه
- `check_for_sub_system()`: گرفتن اطلاعات مربوط به زیر سیستم های یک سیستم

## • SysRoles

برای ارتباط با جدول roles

متدها:

- `add_role()`: افزودن یک نقش جدید
- `is_exist()`: چک کردن وجود یک نقش خاص در سیستم
- `add_new_system_role()`: افزودن نقش های کاربری جدید به یک سیستم خاص
- `get_all()`: گرفتن کلیه نقش های یک سیستم
- `get_js_tree()`: گرفتن نقش های یک سیستم خاص به صورت json مورد نیاز برای نمایش در js\_tree
- `get_one()`: نمایش اطلاعات یک نقش از یک زیر سیستم
- `get_one_by_name()`: گرفتن اطلاعات یک نقش خاص از یک سیستم با نام
- `delete_role()`: حذف یکی از نقش های یک سیستم خاص
- `update_role()`: به روز رسانی اطلاعات مربوط به یکی از نقش های سیستم
- `get_guest_role_id()`: گرفتن آی دی مربوط به نقش مهمان (GUEST) یک سیستم خاص

## • SysCollections

برای ارتباط به جدول Collections

- `is_exists()`: چک کردن وجود یک مجموعه خاص در یک سیستم
- `insert_one()`: افزودن یک مجموعه جدید به یک سیستم

- add\_collection(): افزودن مجموعه(ها) به یک سیستم
- add\_default\_collections(): افزودن مجموعه های پیش فرض به یکی سیستم
- add\_parent\_collection(): افزودن مجموعه مادر به لیست مجموعه های یک سیستم
- edit\_collection(): ویرایش اطلاعات یک مجموعه
- get\_id\_by\_groupid(): گرفتن کل مجموعه های یک سیستم خاص
- build\_tree(): برگرداندن تعداد مجموعه های مورد نیاز برای ساخت درخت js\_tree
- get\_parent(): برگرداندن اطلاعات مربوط به مجموعه مادر یک مجموعه درسیستم
- get\_rood\_id(): برگرداندن آی دی مجموعه اصلی یک سیستم
- get\_one(): برگرداندن اطلاعات یک مجموعه خاص از سیستم
- get\_system(): بررسی موجود بودن یک مجموعه در یک سیستم خاص
- get\_image\_id(): برگرداندن نام تصویر یک مجموعه خاص
- get\_all(): برگرداندن اطلاعات مربوط به مجموعه های یک سیستم خاص
- get\_children(): گرفتن زیر مجموعه های یک سیستم
- get\_children\_id(): گرفتن آی زیر مجموعه های یک سیستم خاص
- delete\_collection(): حذف یکی از مجموعه های یک سیستم خاص

## • SysCollectionRoles

برای ارتباط با جدول CollectionRoles

متدها:

- add\_role\_collections(): افزودن نقش کاربری به مجموعه خاص از سیستم
- change\_defualt\_collections(): تغییر مجموعه پیش فرض برای یک نقش کاربری در سیستم
- get\_all\_ids(): برگرداندن کلیه مجموعه های مرتبط به یک نقش خاص در سامانه
- get\_all\_ids2(): برگرداندن کلیه مجموعه های فعال مرتبط به یک نقش خاص در سامانه
- delete\_all\_role()\_\_\_: حذف کلیه مجموعه های یک نقش خاص در سیستم

## • SysUsers

برای ارتباط با جدول Users

متدها:

- get\_all\_users\_subsystem(): گرفتن کلیه کاربران یک زیر سیستم خاص از سامانه
- get\_all\_for\_login(): گرفتن مشخصات یک کاربر خاص
- get\_admin\_system(): جستجوی یک کاربر با نام کاربری خاص در یک سیستم
- get\_admin\_id\_system(): گرفتن آی دی کاربر ادمین سامانه
- update\_user(): به روز رسانی مشخصات مربوط به یک کاربر خاص
- add\_user(): افزودن کاربر جدید به یک سیستم در صورتی که از قبل با این نام کاربری به سیستم اضافه نشده باشد.
- is\_exist\_for\_login(): بررسی صحت نام کاربری و کلمه عبور کاربر
- is\_exist(): برگرداندن اطلاعات مربوط به یک کاربر خاص در سامانه
- get\_one(): بررسی وجود داشتن یک کاربر در سامانه
- check\_password(): چک کردن رمز عبور کاربری خاص از یک سیستم
- get\_all(): گرفتن اطلاعات کلیه کاربران یک سیستم خاص
- get\_all\_by\_role(): گرفتن کلیه کاربران با نقش خاص یک سیستم
- get\_guest\_users(): اطلاعات کاربران مهمان یک سیستم خاص
- get\_all\_by\_collection(): گرفتن اطلاعات کاربران مربوط به یک مجموعه خاص در یک

سیستم

- delete\_user(): حذف یکی از کاربران سیستم خاص
- update\_user(): به روز رسانی اطلاعات یک کاربر خاص از یک سیستم
- update\_password(): به روز رسانی کلمه عبور کاربر خاص از یک سیستم
- Count(): برگرداندن تعداد کاربران یک سیستم خاص
- SysUserCollections: برای ارتباط با جدول userCollection
- get\_all(): برگرداندن کلیه مجموعه های یک کاربر خاص
- get\_all(): برگرداندن کلیه مجموعه های یک کاربر خاص
- set\_user\_collections(): ست کردن مجموعه های انتخابی برای یک کاربر خاص
- delete\_all\_collection(): حذف کلیه مجموعه های یک کاربر خاص
- delete\_all\_collection2(): حذف کلیه مجموعه های یک کاربر خاص بر اساس آی دی کاربر و شماره دستگاه

- `get_all_user_col()`: برگرداندن کلید مجموعه های یک کاربر خاص

## • SysSubDomain

برای ارتباط با جدول `subdomain`

- `get_all()`: گرفتن کلید زیردامنه های سامانه
- `get_one_sub()`: برگرداندن اطلاعات یک زیر دامنه خاص
- `add_sub_domain()`: افزودن زیر دامنه به سامانه
- `delete_sub()`: حذف یک زیر دامنه خاص از سامانه
- `update_sub()`: به روز رسانی اطلاعات مربوط به یک زیر دامنه خاص

## • SysNotificationByType

برای ارتباط با جدول `Notification_by_type`

متدها:

- `get_all()`: گرفتن کلید اعلان ها
- `update()_`: به روز رسانی وضعیت یک اعلان خاص
- `get_one_status()`: گرفتن وضعیت یک اعلان خاص

## • SysUserRoles

برای ارتباط با جدول `UserRoles`

متدها:

- `get_all()`: گرفتن کلید نقش های کاربری یک کاربر خاص
- `is_editor()`: چک کردن اینکه کاربر نقش ویرایشگر در سیستم دارد یا خیر
- `add_user_roles()`: به روز رسانی نقش های کاربر خاص
- `add_user_roles2()`: افزودن نقش(های) جدید برای یک کاربر خاص
- `__delete_all_roles()`: حذف تمامی نقش های کاربر خاص در سامانه
- `__delete_all_roles_2()`: حذف کلید نقش های کاربر در سیستم خاص

- SysEditorsCollections

برای ارتباط با جدول editirscollectiona

متدها:

- get\_all(): گرفتن کلیه مجموعه های یک کاربر ویرایشگر خاص
- add\_editors\_collection(): ذخیره مجموعه های جدید برای کاربر ویرایشگر
- delete\_all\_editors\_collection()\_\_\_ حذف کلیه مجموعه های مرتبط با یک کاربر ویرایشگر

- SysEditorsSetting

برای ارتباط با جدول EditorSetting

متد ها:

- add\_setting(): افزودن تنظیمات برای کاربر ویرایشگر
- get\_one(): گرفتن اطلاعات تنظیمات مربوط به یک کاربر ویرایشگر خاص
- update\_setting(): به روز رسانی تنظیمات مربوط به کاربر ویرایشگر
- delete\_setting(): حذف تنظیمات مربوط به یک کاربر ویرایشگر خاص

- SysNews

برای ارتباط با جدول News

- add\_news(): افزودن خبر جدید
- add\_news\_site(): افزودن خبر جدید
- update\_news(): به روز رسانی اطلاعات مربوط به یک خبر
- update\_status\_news(): به روز رسانی وضعیت (فیلد Status) یک خبر خاص.
- get\_news(): برگرداندن اطلاعات مربوط به یک خبر خاص
- get\_news\_p4(): گرفتن اطلاعات مربوط به یک خبر خاص
- total\_count\_p4(): برگرداندن تعداد کل خبر های مربوط به یک سیستم خاص

- `total_count_search_p4()`: برگرداندن تعداد کل خبر ها مربوط به مجموعه ای خاص در سیستم مورد نظر
- `get_news_by_search()`: گرفتن خبرهای مربوط به مجموعه ای خاص
- `get_news_by_search_p4()`: گرفتن خبرها مربوط به مجموعه ای خاص از یک سیستم
- `get_by_news_site_id()`: گرفتن کلیه خبرها مربوط به یک سایت
- `delete_by_news_site_id()`: حذف کلیه خبر ها مربوط به سایت موردنظر
- `update_by_news_site_id()`: به روزرسانی اطلاعات اخبار سایت مورد نظر
- `get_one()`: گرفتن اطلاعات مربوط به یک خبر خاص
- `delete_news()`: حذف یک خبر خاص از یک سیستم از سامانه
- `delete_subnews()`: حذف یک خبر خاص از سامانه
- `get_count()`: گرفتن تعداد خبرهای مربوط به هر مجموعه یا کلیه مجموعه های سیستم مورد نظر

#### • SysNewsCollections

جهت ارتباط با جدول NewsCollection

متد ها:

- `add_one()`: افزودن یک مجموعه به خبر
- `add_many()`: افزودن چند مجموعه به خبر
- `update_many()`: به روزرسانی مجموعه های یک خبر خاص
- `get_collection_id_by_news()`: گرفتن مجموعه های یک خبر خاص
- `get_count_by_collection_id()`: برگرداندن تعداد اخبار مربوط به یک مجموعه خاص
- `delete_collections()`: حذف کلیه مجموعه های یک خبر

#### • SysNewsVisit

برای ارتباط با جدول newsvisit

- `last_visit()`: اطلاعات مربوط به آخرین بازدیدکنندگان یک سیستم خاص

### • SysNewsFile

- add\_one(): برای افزودن اطلاعات یک فایل به خبر
- update\_file\_name(): به روز رسانی نام فایل مربوط به یک خبر
- update\_file\_name\_p4(): به روزرسانی نام تصادفی یک فایل از یک خبر
- get\_all\_dict(): گرفتن اطلاعات کلیه فایل ها مربوط به یک خبر از سیستم
- get\_all\_dict\_api(): گرفتن اطلاعات کلیه فایل ها مربوط به یک خبر از سیستم به همراه لینک کامل تصویر خبر
- get\_one(): گرفتن اطلاعات مربوط به یک فایل خاص از یک خبر یک سیستم
- delete\_bash\_files(): حذف فایل های اضافی از سرور یک سیستم خاص
- delete\_bash\_files\_p4(): حذف کلیه فایل های اضافی از سرور

### • SysNewsLink

برای ارتباط با جدول NewsLink

متد ها:

- add\_one(): افزودن لینک جدید به یک خبر
- get\_all\_dict(): گرفتن اطلاعات مربوط به لینک های یک خبر
- get\_all\_dict\_api(): گرفتن اطلاعات مربوط به لینک های یک خبر
- update\_link(): به روز رسانی اطلاعات مربوط به لینک یک خبر
- delete\_links(): حذف لینک(ها) مربوط به یک خبر
- delete\_links\_news\_site(): حذف کلیه لینک های خبری یک خبر خاص

### • SysNewsPic

جهت ارتباط با جدول newspic

متد ها:

- add\_one(): اضافه کردن یک تصویر به خبر
- add\_one\_pic\_site(): اضافه کردن یک تصویر جدید به خبر در یک سیستم
- add\_many(): اضافه کردن چندی عکس به یک خبر خاص در یک سیستم
- get\_all(): گرفتن تصاویر مربوط به یک خبر خاص



- `get_one2()`: گرفتن لینک دانلود تصاویر مربوط به یک خبر
- `get_all_dict()`: گرفتن اطلاعات کامل تصاویر مربوط به یک خبر
- `get_one()`: گرفتن اطلاعات مربوط به یک تصویر خاص از خبر
- `delete_pics()`: حذف تصاویر از سرور و دیتابیس
- `delete_bash_pics()`: حذف تصاویر از سرور و دیتابیس

## • SysGcmUsers

برای ارتباط با جدول GcmUsers

متدها:

- `get_all_users_for_news()`: برگرداندن کلیه کاربران یک مجموعه خاص
- `get_all_users_for_poll()`: برگرداندن کلیه کاربران یک سیستم خاص
- `get_all_users_reg_id()`: برگرداندن کلیه کاربران ریجستر شده در سامانه

## • SysSystemLog

برای ارتباط با جدول SyestemLog

متدها:

- `add_log()`: اضافه کردن لاگ جدید
- `SysSuggestions`: برای ارتباط با جدول suggestions
- `Add()`: افزودن نظر جدید
- `get_new_count()`: برگرداندن تعداد پیام های خوانده نشده
- `get_all()`: برگرداندن لیست انتقادات و پیشنهادات یک سیستم خاص
- `get_all_p4()`: برگرداندن لیست انتقادات و پیشنهادات یک سیستم خاص
- `get_one()`: برگرداندن اطلاعات مربوط به یک پیام خاص
- `delete_p4()`: حذف یک پیام خاص از سیستم
- `update_p4()`: به روز رسانی اطلاعات مربوط به وضعیت یک پیام خاص
- `get_one_p4()`: برگرداندن اطلاعات مربوط به یک پیام خاص زیر سیستم
- `Update()`: به روز رسانی اطلاعات مربوط به یک پیام از سیستم خاص

- `change_status()`: تغییر وضعیت یک سیستم خاص در سامانه

## • SysTags

برای ارتباط به جدول `tags`

متدها:

- `Add()`: افزودن کلیدواژه
- `add_many()`: افزودن همزمان چند کلیدواژه
- `get_all()`: دریافت کلیه کلیدواژه های یک سیستم
- `get_one()`: گرفتن اطلاعات مربوط به یک کلیدواژه
- `get_one_by_name()`: جستجوی یک کلیدواژه در یک سیستم بر اساس نام
- `Update()`: به روز رسانی اطلاعات مربوط به یک کلیدواژه

## • SysNewsTag

ارتباط با جدول `newstag`

- `Add()`: افزودن کلیدواژه به خبر
- `get_all()`: برای برگرداندن خبر های یک تگ و یا تگ های انتخابی برای یم خبر خاص
- `delete_by_news_id()`: حذف کلیه تگ های یک خبر خاص
- `delete_by_tag_id()`: حذف یک تگ خاص از یک خبر

## • SysNewsComments

برای ارتباط با جدول `newsComment`

متدها:

- `Add()`: افزودن کامنت جدید برای یک خبر
- `get_all()`: برای گرفتن کلیه کامنت های ثبت شده در یک سیستم
- `get_one()`: نمایش اطلاعات یک دیدگاه خاص

- delete\_by\_id(): حذف یک کامنت خاص از سیستم
- Update(): به روز رسانی اطلاعات مربوط به یک دیدگاه
- new\_news\_comment\_count(): گرفتن تعداد کامنت های جدید یک سیستم

#### • SysNewsLikes

برای ارتباط با جدول newslikes

متد ها:

- add\_like(): افزودن لایک به یک خبر از یک سیستم
- delete\_like(): حذف لایک یک کاربر از یک خبر از سیستم
- get\_all(): گرفتن کلیه افرادی که پست های یک سیستم را لایک کرده اند.
- get\_one\_news\_like\_count(): گرفتن تعداد لایک های یک خبر خاص
- is\_user\_liked(): چک کردن اینکه یک کاربر خبری خاص را پسندیده است یا خیر.
- delete\_by\_id(): حذف یک لایک از خبر

#### • SysMessages

برای ارتباط با جدول messages

متد ها:

- Send(): ارسال پیام به یک کاربر
- get\_system(): گرفتن اطلاعات سیستم مربوط به یک پیام
- delete\_msg\_p4(): حذف یک پیام از سیستم
- send\_many(): برای ارسال جمعی پیام
- get\_all()\_: گرفتن همه پیامهای کاربران سامانه
- get\_all\_for\_system(): گرفتن همه پیام های سیستم یک خاص
- get\_all\_for\_user(): گرفتن پیام های مربوط به یک کاربر خاص
- get\_new\_messages\_count(): گرفتن پیام های جدید ارسالی کاربران
- get\_one(): گرفتن اطلاعات مربوط به یک پیام
- Update(): به روز رسانی اطلاعات مربوط به یک پیام خاص

- update\_all\_messages\_from\_user(): تغییر وضعیت پیام های یک کاربر خاص به خواننده شده
- change\_status(): تغییر وضعیت یک پیام خاص
- get\_all\_sent\_for\_system(): گرفتن فهرست پیام های ارسالی به کاربران در یک سیستم خاص

#### • SysPollQuestion

برای ارتباط با جدول pollQuestion

متد ها :

- Insert(): برای افزودن نظرسنجی جدید
- children\_question(): گرفتن سوالات مربوط به یک نظر سنجی خاص
- update\_status(): تغییر وضعیت یک نظرسنجی خاص
- Delete(): حذف یک نظرسنجی از سیستم
- get\_all(): گرفتن کلیه نظرسنجی های ثبت شده درسیستم

#### • SysPoolItem

برای ارتباط با جدول poolitem

متد ها:

- insert\_list\_item(): برای درج گزینه(های) جدید برای یک نظرسنجی
- get\_all(): گرفتن همه گزینه های یک نظرسنجی

#### • SysPollAnswer

برای ارتباط با جدول pollAnswer

متد ها:

- get\_all()

گرفتن لیست پاسخ های مربوط به یک سوال

- SysTickets

برای ارتباط با جدول tickets

متدها:

- Send() : ارسال تیکت جدید
- date\_update() : به روز رسانی تاریخ آخرین آپدیت تیکت
- change\_status() : عوض کردن وضعیت تیکت خاص
- is\_exist() : بررسی وجود داشتن یک تیکت خاص در یک سیستم

- SysSubTickets

برای ارتباط با جدول subticket

- send\_subticket() : ارسال پیام برای یک تیکت خاص
- get\_all() : دریافت پیام های مربوط به یک تیکت خاص

- SysForum

برای ارتباط با جدول forum

متدها:

- add\_forum() : افزودن فروم جدید به سیستم
- update\_forum() : به روز رسانی اطلاعات مربوط به یک فروم خاص در سیستم
- get\_all() : گرفتن کلیه فروم های یک سیستم خاص
- forum\_delete() : حذف یکی از فروم های سامانه
- update\_status() : به روز رسانی اطلاعات مربوط به یک فروم خاص
- get\_one() : گرفتن اطلاعات مربوط به یک فروم خاص

## • SysForumTopic

forum topic برای ارتباط با جدول

متد ها :

▪ get\_all()

گرفتن اطلاعات مربوط به موضوعات فروم

▪ get\_one() : گرفتن اطلاعات یک موضوع فروم خاص

▪ topic\_delete() : حذف کردن یکی از موضوعات فروم

▪ update\_status() : به روز رسانی وضعیت یکی از موضوعات

▪ update\_topic() : به روز رسانی اطلاعات مربوط به یک موضوع

▪ add\_topic() : افزودن موضوع جدید به فروم در یک سیستم خاص

## • SysForumPost

froumPost برای ارتباط با جدول

متد ها:

▪ get\_all() : گرفتن کلیه پست های ارسالی در یک موضوع خاص

▪ get\_all\_pending() : گرفتن کلیه پست های منتظر بررسی

▪ get\_one() : گرفتن اطلاعات یکی از پست های بخش فروم ها

▪ delete\_by\_id() : حذف اطلاعات مربوط به یک پست

▪ Update() : به روزرسانی اطلاعات مربوط به یک پست

▪ update\_p4() : به روزرسانی اطلاعات مربوط به یک پست

▪ send\_post() : افزودن پست جدید برای یک سیستم در یک موضوع برای یک فروم

خاص

## • SysSystemPermissions

system\_permissions برای ارتباط با جدول

متد ها:

▪ get\_all() : گرفتن کلیه سطوح دسترسی سیستم

▪ get\_one() : گرفتن مشخصات یکی از سطوح دسترسی سیستم

- SysSystemPlans

برای ارتباط با جدول system\_Plans

متد ها:

- Insert(): درج پلن جدید در سامانه
- get\_one(): گرفتن اطلاعات مربوط به یک پلن

- SysPlanPermissions

برای ارتباط با جدول plan\_permissions

متد ها:

- get\_all\_perm\_id(): گرفتن آی دی تمامی سطوح دسترسی پلن ها
- get\_all\_perms\_by\_plan(): گرفتن مشخصات سطوح دسترسی مربوط به یک پلن خاص

- SysPermissions

برای ارتباط با جدول ppermissions

متد ها:

- add\_user\_permission(): افزودن سطح دسترسی جدید برای کاربر
- get\_all\_perms(): گرفتن کلیه سطوح دسترسی کاربران
- delete\_all\_perms()\_\_: حذف کلیه سطوح دسترسی یک کاربر در یک سیستم خاص

- SysPlans

برای ارتباط با جدول plans

متد ها:

- get\_one(): گرفتن مشخصات یکی از پلن های سیستم

- SysPlanFeatures

برای ارتباط با جدول plan\_features

متد ها:

- get\_one() : گرفتن اطلاعات مربوط به خصوصیات یکی از پلن ها

- SysSystemPlanFeatures

برای ارتباط با جدول system\_plan\_feature

متد ها:

- Insert() : افزودن خصوصیت جدید برای یک پلن برای یک سیستم
- copy\_plan\_features() : کپی کردن خصوصیات یک پلن از یک سیستم
- get\_one() : گرفتن اطلاعات مربوط به یکی از پلن های سیستم



# فصل ۴

نتیجه گیری

## نتیجه گیری

استفاده از فریم ورک ها در طراحی و توسعه اپلیکیشن های مبتنی بر وب، نقش به سزایی دارد. سادگی و سهولت در استفاده و همچنین آماده بودن اکثر کتابخانه های مورد نیاز برای توسعه نرم افزار، از جمله مزیت های آن به شمار می رود.

فریم ورک تورنادو نیز به دلیل سادگی و سبکی و نصب ساده، جزء فریم ورک های محبوب به شمار می رود. توسعه سامانه خبرنامه به دلیل نیاز به سرعت اجرای بالا، با استفاده از این فریم ورک صورت گرفته است.

استفاده از پایگاه داده MySQL نیز به دلیل ماهیت رابطه ای بودن آن، برای این سامانه بسیار مناسب بود. همچنین استفاده از پایگاه داده Redis برای Session ها و بحث Caching یکی از عوامل تاثیرگذار در سرعت اجرای نرم افزار می باشد.

## مراجع

- [1] <http://www.tornadoweb.org/en/stable/web.html>
- [2] <https://dev.mysql.com/>

## آدرس پروژه روی Git

این پروژه به صورت عمومی بر روی ریپوزیتوری خبرنامه در آدرس زیر قابل دسترسی می باشد:

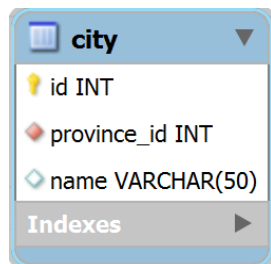
<https://gitlab.com/Reza-S4/Khabarnameh-Bozorgmehr.git>

## پیوست ۱

فهرست و شرح جداول استفاده شده (به ترتیب حرف الفبا):

نام جدول:

• City



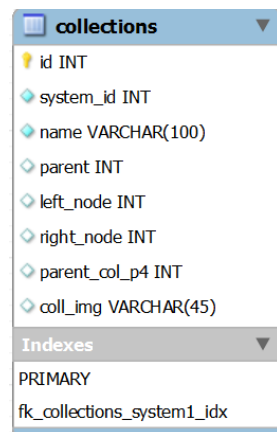
city	
id	INT
province_id	INT
name	VARCHAR(50)
Indexes	

شرح مختصر داده ها:

- فهرست شهر های ایران به همراه آی دی استان مربوطه در این جدول وجود دارد.

نام جدول:

• Collections



collections	
id	INT
system_id	INT
name	VARCHAR(100)
parent	INT
left_node	INT
right_node	INT
parent_col_p4	INT
coll_img	VARCHAR(45)
Indexes	
PRIMARY	
fk_collections_system1_idx	

شرح مختصر داده ها:

- نام مجموعه های تعریف شده برای اخبار در این جدول نگه داری می شود.

نام جدول:

• Collections\_rules

collections_roles	
id	INT
role_id	INT
collection_id	INT
status	ENUM('yes','no')
Indexes	
PRIMARY	
fk_collections_roles_roles1_idx	
fk_collections_roles_collections1_idx	

شرح مختصر داده ها:

- فهرست مجموعه های (collection) های مجاز برای هر گروه کاربری در این جدول مشخص می شود.

نام جدول:

• Editors\_collections

editors_collections	
id	INT
user_id	INT
collection_id	INT
Indexes	
PRIMARY	
fk_editors_collections_users1_idx	
fk_editors_collections_collections1_idx	

شرح مختصر داده ها:

- فهرست مجموعه های مجاز برای هر کاربر با نقش ویرایشگر در سامانه در جا مشخص می شود.

نام جدول:

• Editors\_setting

editors_setting	
id	INT
user_id	INT
edit_news	ENUM(...)
direct_publish	ENUM('YES', 'NO')
Indexes	
PRIMARY	
user_id_UNIQUE	
fk_editors_setting_users1_idx	

2شرح مختصر داده ها:

- تنظیمات مربوط به کاربران با نقش ویرایشگر در سامانه در این جدول نگه داری میشود.

نام جدول:

• Manager

manager	
id	INT
username	VARCHAR(45)
passwd	VARCHAR(45)
first_name	VARCHAR(45)
last_name	VARCHAR(45)
Indexes	
PRIMARY	

شرح مختصر داده ها:

- اطلاعات مربوط به کاربران با نقش مدیریت در این جا نگه داری می شود.

نام جدول:

• Messages

messages	
id	INT
system_id	INT
user_id	INT
text	TEXT
date	TIMESTAMP
status	ENUM(...)
msg_type	ENUM('s2u', 'u2s')
type	ENUM('sms', 'advice')
Indexes	
PRIMARY	
fk_messages_users1_idx	
fk_messages_system1_idx	

شرح مختصر داده ها:

- این جدول برای نگه داری اطلاعات مربوط به بخش پیام ها ایجاد شده است.

نام جدول:

• News

news	
id	INT
system_id	INT
user_id	INT
title	VARCHAR(255)
web_text	TEXT
text	TEXT
_type	ENUM(...)
public	TINYINT(1)
status	ENUM(...)
date	TIMESTAMP
visit	INT
like	INT
comment	INT
news_site_id	VARCHAR(45)
source_news_site	TEXT
Indexes	
PRIMARY	
fk_news_users1_idx	
fk_news_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به بخش اخبار(پست ها)ی سامانه در این جدول نگه داری می شود.

نام جدول:

News\_collections •

news_collections	
id	INT
news_id	INT
collection_id	INT
Indexes	
PRIMARY	
fk_news_collections_news1_idx	
fk_news_collections_collections1_idx	

شرح مختصر داده ها:

- این جدول برای مشخص کردن دسته بندی(های)(collection) هر خبر(news) ایجاد شده است.

نام جدول:

News\_comments •

news_comments	
id	INT
system_id	INT
news_id	INT
user_id	INT
text	VARCHAR(150)
confirmed	ENUM('yes', 'no')
date	TIMESTAMP
Indexes	
PRIMARY	
fk_new_comments_news1_idx	
fk_new_comments_users1_idx	
fk_news_comments_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به دیدگاه های هر خبر(پست) در این جدول نگه داری می شود.

نام جدول:

News\_favorite •

news_favorite	
id	INT
news_id	INT
date	TIMESTAMP
user_id	INT
system_id	INT
Indexes	
PRIMARY	
fk_news_favorite_news1_idx	
fk_news_favorite_users1_idx	
fk_news_favorite_system1_idx	

شرح مختصر داده ها:

- این جدول برای نگه داری اطلاعات اخباری است که کاربران در لیست اخبار دلخواه خود قرار داده اند.



نام جدول:

• News\_file

news_file	
id	INT
file_name	VARCHAR(45)
file_random_name	VARCHAR(45)
news_id	INT
system_id	INT
file_type	VARCHAR(45)
Indexes	
PRIMARY	
fk_news_file_news1_idx	
fk_news_file_system1_idx	

شرح مختصر داده ها:

- برای نگه داری اطلاعات فایل هایی که برای هر خبر (پست) آپلود شده اند.

نام جدول:

• News\_likes

news_likes	
id	INT
system_id	INT
user_id	INT
news_id	INT
date	TIMESTAMP
Indexes	
PRIMARY	
system_id_UNIQUE	
fk_news_likes_news1_idx	
fk_news_likes_users1_idx	
fk_news_likes_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به اخبار (پست ها) ی که کاربران پسندیده اند (like) در این جدول نگه داری میشود.

نام جدول:

• News\_link

news_link	
id	INT
link_name	VARCHAR(45)
link_address	TEXT
news_id	INT
system_id	INT
random_id	VARCHAR(45)
Indexes	
PRIMARY	
fk_news_link_news1_idx	
fk_news_link_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به پیوند های هر خبر (link) در این جدول نگه داری می شوند.

نام جدول:

News\_pic •

news_pic	
id	INT
system_id	INT
news_id	INT
pic_name	VARCHAR(45)
Indexes	
PRIMARY	
fk_news_pic_news1_idx	
fk_news_pic_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به تصاویری که برای هر خبر(پست) انتخاب شده اند در اینجا نگه داری می شود.

نام جدول:

News\_tag •

news_tag	
id	INT
news_id	INT
tag_id	INT
Indexes	
PRIMARY	
news_id_UNIQUE	
fk_news_tag_tags1_idx	
fk_news_tag_news1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به کلیدواژه(tag)های انتخابی کاربران بر اخبار در این جدول می باشد.

نام جدول:

News\_visit •

news_visit	
id	INT
news_id	INT
date	TIMESTAMP
user_id	INT
system_id	INT
Indexes	
PRIMARY	
fk_news_visit_news1_idx	
fk_news_visit_users1_idx	
fk_news_visit_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به بازدید هر یک از کاربران از اخبار هر کدام از مجموعه ها در این جدول نگه داری میشود.

نام جدول:

• Nofitication\_by\_type

notification_by_type	
id	INT
_type	ENUM(...)
name	VARCHAR(45)
status	ENUM('yes', 'no')
Indexes	
PRIMARY	

شرح مختصر داده ها:

- انواع اعلان ها(notification)ی ارسالی اپلیکیشن برای کاربران در این جدول نگه داری می شود.

نام جدول:

• Permissions

permissions	
id	INT
system_id	INT
user_id	INT
perm_name	VARCHAR(100)
Indexes	
PRIMARY	
fk_permissions_users1_idx	
fk_permissions_system1_idx	
fk_permissions_system_permissions1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به سطوح دسترسی کاربران سامانه در این جدول نگه داری می شود.

نام جدول:

• Plans

plans	
id	INT
plan_name	VARCHAR(45)
plan_type	ENUM(...)
price	VARCHAR(20)
Indexes	
PRIMARY	

شرح مختصر داده ها:

- این جدول شامل پلن های مالی در نظر گرفته شده برای سامانه می باشد.

نام جدول:

Plan\_features •

plan_features	
id	INT
plan_id	INT
max_users	INT
active_days	INT
sub_system	TINYINT
Indexes	
PRIMARY	
fk_plan_features_plans1_idx	

شرح مختصر داده ها:

- این جدول شامل جزئیات مربوط به هر پلن می باشد.

نام جدول:

Plan\_permissions •

plan_permissions	
id	INT
plan_id	INT
permission_id	VARCHAR(100)
Indexes	
PRIMARY	
fk_plan_permissions_plans1_idx	
fk_plan_permissions_system_permissions1_idx	

شرح مختصر داده ها:

- این جدول شامل اطلاعات مربوط به سطوح دسترسی برای هر پلن است. که در نهایت مشخص کننده امکاناتی است که از سوی ادمین برای هر کدام از پلن ها در نظر گرفته شده است.

نام جدول:

Poll\_answer •

poll_answer	
id	INT(11)
user_id	INT(11)
question_id	INT(11)
item_id	INT(11)
date	TIMESTAMP
Indexes	
PRIMARY	
user_id_UNIQUE	
fk_answer_users1_idx	
fk_answer_question1_idx	
fk_answer_item1_idx	

شرح مختصر داده ها:

- پاسخ کاربران به نظرسنجی های مطرح شده در سامانه در این جدول قرار می گیرد.

نام جدول:

• Poll\_item

poll_item	
id	INT(11)
question_id	INT(11)
item	VARCHAR(45)
Indexes	
PRIMARY	
fk_item_question1_idx	

شرح مختصر داده ها:

- گزینه های مطرح شده در هر یک از نظر سنجی های سامانه در این جدول قرار دارد.

نام جدول:

• Poll\_questions

poll_question	
id	INT(11)
system_id	INT(11)
question	TEXT
date	TIMESTAMP
status	ENUM(...)
parent_question	INT
Indexes	
PRIMARY	
fk_question_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به هر نظرسنجی مطرح شده توسط کاربران سامانه در این بخش قرار دارد.

نام جدول:

• Province

province	
id	INT
name	VARCHAR(45)
Indexes	
PRIMARY	

شرح مختصر داده ها:

- فهرست کلیه استان های کشور در این جدول قرار دارد.

نام جدول:

• Roles

roles	
id	INT
system_id	INT
name	VARCHAR(100)
title	VARCHAR(100)
Indexes	
PRIMARY	
fk_roles_system1_idx	

شرح مختصر داده ها:

- فهرست کاربران (نقش‌ها) قابل تعریف در سامانه در این جدول نگه داری می شود.

نام جدول:

• Slider\_images

slider_images	
id	INT(11)
system_id	INT(11)
title	VARCHAR(255)
name	VARCHAR(45)
date	TIMESTAMP
Image_type	ENUM(...)
Indexes	
PRIMARY	
fk_slider_images_system1_idx	

شرح مختصر داده ها:

- تنظیمات مربوط به تصاویر اسلایدر هر یک از سیستم ها در این جدول نگه داری میشود.

نام جدول:

• Subticket

subticket	
id	INT(11)
txt	VARCHAR(1000)
file	VARCHAR(100)
date	TIMESTAMP
type	VARCHAR(45)
tickets_id	INT(11)
users_id	INT(11)
Indexes	
PRIMARY	
fk_subticket_tickets1_idx	
fk_subticket_users1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به محاورات انجام شده در هریک از تیکت های ایجاد شده توسط هر کاربر در این جدول قرار می گیرد.

نام جدول:

• Sub\_domain

sub_domain	
id	INT
site_id	VARCHAR(45)
sub_domain	VARCHAR(45)
Indexes	
PRIMARY	

شرح مختصر داده ها:

- آدرس زیر دامنه های استفاده شده برای هر یک از سایت ها در این جدول نگه داری می شود.

نام جدول:

• Suggestion

suggestions	
id	INT
user_id	INT
system_id	INT
text	VARCHAR(500)
date	TIMESTAMP
status	ENUM(...)
read_by	VARCHAR(250)
Indexes	
PRIMARY	
fk_suggestions_system1_idx	
fk_suggestions_users1_idx	

شرح مختصر داده ها:

- پیشنهادات ارسالی توسط کاربران در هر کدام از سامانه ها در این جدول نگه داری می شود.

نام جدول:

• System

system	
id	INT
name	VARCHAR(255)
pname	VARCHAR(255)
pic	VARCHAR(45)
color	VARCHAR(10)
address	VARCHAR(100)
admin_username	VARCHAR(50)
tel	VARCHAR(20)
province_id	INT
city_id	INT
status	ENUM(...)
confirmed	ENUM('yes', 'no')
reg_date	TIMESTAMP
email	VARCHAR(50)
parent_system_id	INT(11)
Indexes	
PRIMARY	
fk_system_city1_idx	
fk_system_province1_idx	
fk_system_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به هر کدام از سیستم های استفاده کننده از سامانه در این جدول نگه داری می شود.

نام جدول:

• System\_permissions

system_permissions	
name	VARCHAR(100)
title	VARCHAR(100)
handlers	VARCHAR(500)
url	VARCHAR(100)
default	TINYINT
icon	VARCHAR(45)
parent	VARCHAR(100)
perm_order	INT
Indexes	
PRIMARY	
name_UNIQUE	

شرح مختصر داده ها:

- کلیه اطلاعات سطوح دسترسی هر یک از سیستم های سامانه در این جدول نگه داری می شود.

نام جدول:

• System\_plans

system_plans	
id	INT
system_id	INT
plan_id	INT
Indexes	
PRIMARY	
system_id_UNIQUE	
fk_system_plans_system1_idx	
fk_system_plans_plans1_idx	

شرح مختصر داده ها:

- اطلاعات این جدول، مشخص کننده پلن انتخابی برای هر یک از سیستم های سامانه می باشد.

نام جدول:

• System\_plan\_features

system_plan_features	
id	INT
system_id	INT
max_users	INT
active_days	INT
sub_system	TINYINT
Indexes	
PRIMARY	
fk_system_plan_fetures_system1_idx	

شرح مختصر داده ها:

- مشخص کننده جزئیات مربوط به پلن های هر یک از سیستم های سامانه می باشد.



نام جدول:

• Tags

tags	
id	INT
system_id	INT
name	VARCHAR(45)
date	TIMESTAMP
Indexes	
PRIMARY	
fk_tags_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به کلیه کلمات کلیدی که توسط کاربرای وارد شده اند.

نام جدول:

• Tickets

tickets	
id	INT(11)
topic	VARCHAR(250)
status	VARCHAR(45)
system_id	INT(11)
priority	VARCHAR(45)
last_update	TIMESTAMP
Indexes	
PRIMARY	
fk_tickets_system1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به تیکت های ایجاد شده توسط کاربران سامانه در این جدول نگه داری می شود.

نام جدول:

• Users

users	
id	INT
system_id	INT
username	VARCHAR(100)
password	VARCHAR(50)
first_name	VARCHAR(100)
last_name	VARCHAR(255)
status	ENUM(...)
Indexes	
PRIMARY	
fk_users_system1_idx	

شرح مختصر داده ها:

- این جدول شامل کلیه اطلاعات مربوط به کاربران سامانه می باشد.

نام جدول:

• User\_collections

user_collections	
id	INT
user_id	INT
collection_id	INT
device_id	VARCHAR(45)
Indexes	
PRIMARY	
fk_user_collections_users1_idx	
fk_user_collections_collections1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به مجموعه اخبار (collection) هایی است که توسط کاربران انتخاب شده اند.

نام جدول:

• User\_roles

user_roles	
id	INT
user_id	INT
role_id	INT
Indexes	
PRIMARY	
fk_user_roles_users_idx	
fk_user_roles_roles1_idx	

شرح مختصر داده ها:

- اطلاعات مربوط به نقش هر یک از کاربران در سامانه در این جدول نگه داری می شود.



## ضمیمه ۱

نحوه راه اندازی پروژه به صورت سریع:

```
import tornado.ioloop
import tornado.web
class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")
def make_app():
    return tornado.web.Application([
        (r"/", MainHandler),
    ])
if __name__ == "__main__":
    app = make_app()
    app.listen(8888)
    tornado.ioloop.IOLoop.current().start()
```

نمونه کلاس برای مدل های دیتابیس:

```
class SysSettings:
    def __init__(self, _id=None, name=None, value=None):
        self.id = _id
        self.name = name
        self.value = value

    def insert(self):
        try:
            x = Settings.insert(
                name=self.name,
                value=self.value
            ).execute()
            if x:
                return x
            else:
                return False
        except Exception, e:
            return False

    def get_one(self):
        try:
            x = Settings.get(Settings.name == self.name)
            if x:
                return x.value
            else:
                return None
        except Exception, e:
            return None
```



**Bozorgmehr University of Qaenat  
Engineering College**

**Thesis of Bachelor of Science of software engineering**

**KhabarNameh Notification System**

Supervisor:  
**Seyyed Mojtaba Banaei**

By:  
**Reza Safaeiyan**

**Spring 2017**