***About the Sample IITBombayX Remote DRF Application***

*This application is built to Confirm some IITBombayX utilities like custom registration page, enrollment etc. from remote application external Registration with the help of IITBombayX database credentials.*

***Status:*** *In progress... Users can register and enroll in the available courses on the IITBombayX platform using IITBombayX database credentials.*

***IITBombayX Release:*** *Ironwood.master*

***Utilities List:***

* *Custom Registration Page*
* *Course Enrollment*

***About Custom Registration Page***

***Registration pages used Tables/Collections in IITBombayX Database :***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Database** | **Tables/Collections** | **Notes, If any.** |
| 1 | MySQL: edxapp | auth\_user |  |
| 2 | MySQL: edxapp | auth\_userprofile |  |
| 3 | MySQL: edxapp | custom\_reg\_form\_extrainfo | * This table is used for IITBombayX Custom Fields (States, City, Pin Code and Aadhar Id (UIDAI)) on the registration page. |
| 4 | Mongodb: cs\_comments\_service | users |  |

***1. List of fields of the “auth\_user” table in the “edxapp” MySQL database those are used in the Registration page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **fields of the “auth\_user”** | **datatype of fields** | **Label on the registration page.** | **Notes, If any.** |
| 1 | id | int(11) | -Not on page but used by the application. | It is auto\_increment and primary key will be used as reference for the user. |
| 2 | password | varchar(128) | Password | Mandatory. To encrypt it, the set\_password function of django is used. Django uses the PBKDF2 algorithm with a SHA256 hash, a password stretching mechanism recommended by NIST. |
| 3 | username | varchar(150) | Public User Name | Mandatory, unique. |
| 4 | first\_name | varchar(30) | First Name | Not used by IITBombayX. |
| 5 | last\_name | varchar(30) | Last Name | Not used by IITBombayX. |
| 6 | email | varchar(254) | Email Id | Mandatory, unique. |
| 7 | is\_active | tinyint(1) | -Not on page but used by the application. | Static Value “1”. |

**Encrypt Password Algorithm*:***

To encrypt it set\_password function of django is used. Django uses the [*PBKDF2*](https://en.wikipedia.org/wiki/PBKDF2) algorithm with a SHA256 hash, a password stretching mechanism recommended by [*NIST*](https://dx.doi.org/10.6028/NIST.SP.800-132).

**For example:**

**Consider following password**

pbkdf2\_sha256$36000$MflWfLXbejfO$tNrjk42YE9ZXkg7IvXY5fikbC+H52Ipd2mf7m0azttk=

**Let's break this down. The $ are separators:**

* pbkdf2\_sh256 means PBKDF2-SHA256, i.e. hash\_pbkf2('sha256', ...)
* *36000* is the iteration count
* MflWfLXbejfO is the salt
* tNrjk42YE9ZXkg7IvXY5fikbC+H52Ipd2mf7m0azttk= is likely the hash.

This is all the information need to validate the hash. algorithm Used:

1. hash\_pbkdf2(): to generate a new hash from the password provided by the user
2. hash\_equals(): to compare the generated hash with the stored one

Libraries used:

1. hashlib
2. base64
3. random
4. django.utils.crypto

**Password Encryption:**

*algorithm = "pbkdf2\_sha256"*

*# the default value of iterations in python3.8*

*iterations = 36000*

*# The default length of salt is 12 with the a-z, A-Z, 0-9 character set*

*length=12*

*allowed\_chars='abcdefghijklmnopqrstuvwxyz''ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'*

* *salt =''.join(random.choice(allowed\_chars) for i in range(length))*
* *digest = hashlib.sha256*
* *hash = pbkdf2(password, salt, iterations, digest=digest)*
* *hash = base64.b64encode(hash).decode('ascii').strip()*
* *encoded\_password="%s$%d$%s$%s" % (algorithm, iterations, salt, hash)*

***For code refer:***[*p*](http://gitlab.cse.iitb.ac.in/saritat/iitbombayxregistrationpage/-/blob/Ironwood.master/password.py)*assword.py*

***2. List of fields of the “auth\_userprofile” table in the “edxapp” MySQL database those are used in the Registration page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Fields of the *auth\_userprofile*** | **Datatype of fields** | **Label on the registration page.** | **Notes, If any.** |
| 1 | name | varchar(255) | Full Name | It is mandatory on IITBombayX registration page, but it is drop in sample application(Just for testing). When it is not set then the value of username is assign. |
| 2 | year\_of\_birth | int(11) | Year of Birth | Optional. |
| 3 | gender | varchar(6) | Gender | Optional. |
| 4 | level\_of\_education | varchar(6) | Highest level of education completed | Optional. |
| 5 | city | longtext | City | Mandatory. |
| 6 | country | varchar(2) | Country | Mandatory. |
| 7 | goals | longtext | Tell us why yo're interested in IITBombayX | Optional. |
| 8 | user\_id | int(11) | -Not on page but used by the application | It is reference key refer from auth\_user table of edxapp MySQL database. |

***3. List of fields of the “custom\_reg\_form\_extrainfo” table in the "edxapp" MySQL database those are used in the Registration page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Fields of the  *custom\_reg\_form\_extrainfo*** | **Datatype of fields** | **Label on the registration page.** | **Notes, If any.** |
| 1 | state | varchar(50) | States | Mandatory. |
| 2 | city | varchar(50) | City | Mandatory. |
| 3 | pincode | int(11) | Pin Code | Mandatory. |
| 4 | aadharid | bigint(20) | Aadhar Id(UIDAI) | Optional. |
| 5 | user\_id | int(11) | -Not on page but used by the application | It is reference key refer from auth\_user table of edxapp MySQL database. |

***4. List of fields of the “users” table in the "cs\_comments\_service" mongodb database those are used in the Registration page:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Fields of the  *users*** | **Label on the registration page.** | **Notes, If any.** |
| 1 | \_id | -Not on page but used by the application | It is reference key refer from auth\_user table of edxapp MySQL database. |
| 2 | default\_sort\_key | -Not on page but used by the application | Static Value "date". |
| 3 | external\_id | -Not on page but used by the application | It is reference key refer from auth\_user table of edxapp MySQL database. |
| 4 | username | Public User Name | Mandatory. |

***List COUNTRIES might be available in the drop-down box are:***

|  |  |  |
| --- | --- | --- |
| Sr.No. | *Drop-down Value* | *Value which is store in the* ***country*** *field of* ***auth\_userprofile*** *table of* ***edxapp*** *MySQL Database* |
| 1 | Alabama | AL |
| 2 | Alaska | AK |
| 3 | Arizona | AZ |
| 4 | Arkansas | AR |
| 5 | Armed Forces Americas | AA |
| 6 | Armed Forces Europe | AE |
| 7 | Armed Forces Pacific | AP |
| 8 | California | CA |
| 9 | Colorado | CO |
| 10 | Connecticut | CT |
| 11 | Delaware | DE |
| 12 | District Of Columbia | DC |
| 13 | Florida | FL |
| 14 | Georgia | GA |
| 15 | Hawaii | HI |
| 16 | Idaho | ID |
| 17 | Illinois | IL |
| 18 | Indiana | IN |
| 19 | Iowa | IA |
| 20 | Kansas | KS |
| 21 | Kentucky | KY |
| 22 | Louisiana | LA |
| 23 | Maine | ME |
| 24 | Maryland | MD |
| 25 | Massachusetts | MA |
| 26 | Michigan | MI |
| 27 | Minnesota | MN |
| 28 | Mississippi | MS |
| 29 | Missouri | MO |
| 30 | Montana | MT |
| 31 | Nebraska | NE |
| 32 | Nevada | NV |
| 33 | New Hampshire | NH |
| 34 | New Jersey | NJ |
| 35 | New Mexico | NM |
| 36 | New York | NY |
| 37 | North Carolina | NC |
| 38 | North Dakota | ND |
| 39 | Ohio | OH |
| 40 | Oklahoma | OK |
| 41 | Oregon | OR |
| 42 | Pennsylvania | PA |
| 43 | Rhode Island | RI |
| 44 | South Carolina | SC |
| 45 | South Dakota | SD |
| 46 | Tennessee | TN |
| 47 | Texas | TX |
| 48 | Utah | UT |
| 49 | Vermont | VT |
| 50 | Virginia | VA |
| 51 | Washington | WA |
| 52 | West Virginia | WV |
| 53 | Wisconsin | WI |
| 54 | Wyoming | WY |

**List STATES in the drop down box are:**

|  |  |  |
| --- | --- | --- |
| Sr.No. | *Drop-down Value* | *Value which is store in the* ***state*** *field of* ***custom\_reg\_form\_extrainfo*** *table of* ***edxapp*** *MySQL Database* |
| 1 | Andaman and Nicobar islands | Andaman and Nicobar islands |
| 2 | Andhra Pradesh | Andhra Pradesh |
| 3 | Arunachal Pradesh | Arunachal Pradesh |
| 4 | Assam | Assam |
| 5 | Bihar | Bihar |
| 6 | Chandigarh | Chandigarh |
| 7 | Chattisgarh | Chattisgarh |
| 8 | Dadra and Nagar Haveli | Dadra and Nagar Haveli |
| 9 | Daman and Diu | Daman and Diu |
| 10 | Delhi | Delhi |
| 11 | Goa | Goa |
| 12 | Gujarat | Gujarat |
| 13 | Haryana | Haryana |
| 14 | Himachal Pradesh | Himachal Pradesh |
| 15 | Jammu and Kashmir | Jammu and Kashmir |
| 16 | Jharkhand | Jharkhand |
| 17 | Karnataka | Karnataka |
| 18 | Kerala | Kerala |
| 19 | Lakshadweep | Lakshadweep |
| 20 | Madhya Pradesh | Madhya Pradesh |
| 21 | Maharashtra | Maharashtra |
| 22 | Manipur | Manipur |
| 23 | Meghalaya | Meghalaya |
| 24 | Mizoram | Mizoram |
| 25 | Nagaland | Nagaland |
| 26 | Orissa | Orissa |
| 27 | Pondicherry | Pondicherry |
| 28 | Punjab | Punjab |
| 29 | Rajasthan | Rajasthan |
| 30 | Sikkim | Sikkim |
| 31 | Tamil Nadu | Tamil Nadu |
| 32 | Telangana | Telangana |
| 33 | Tripura | Tripura |
| 34 | Uttar Pradesh | Uttar Pradesh |
| 35 | Uttarakhand | Uttarakhand |
| 36 | West Bengal | West Bengal |

***List of GENDER\_CHOICES in the drop down box are:***

|  |  |  |
| --- | --- | --- |
| Sr.No. | *Drop-down Value* | *Value which is store in* ***gender*** *field of* ***auth\_userprofile*** *table of* ***edxapp*** *Mysql Database* |
| 1 | Male | m |
| 2 | Female | f |
| 3 | Other/Prefer Not to Say | o |

***List of LEVEL\_OF\_EDUCATION\_CHOICES in the drop-down box are:***

|  |  |  |
| --- | --- | --- |
| Sr.No. | *Drop-down Value* | *Value which is store in the* ***level\_of\_education*** *field of* ***auth\_userprofile*** *table of* ***edxapp*** *MySQL Database* |
| 1 | Doctorate | p |
| 2 | Master's or professional degree | m |
| 3 | Bachelor's degree | b |
| 4 | Associate degree | a |
| 5 | Secondary/high school | hs |
| 6 | Junior secondary/junior high/middle school | jhs |
| 7 | Elementary/primary school | el |
| 8 | No formal education | none |
| 9 | Other education | other |

***About Course Enrollment Page***

***Course Enrollment page used Tables in IITBombayX Database :***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Database** | **Tables/Collections** | **Notes, If any.** |
| 1 | MySQL: edxapp | django\_comment\_client\_role | * This table has five entries for each course, they are added when the course is created. * Different five roles on the courses are Student, Community TA, Group Moderator, Moderator, and Administrator. * Select “id” of the course for “Student” which is treat as a role id in reference table. |
| 2 | MySQL: edxapp | django\_comment\_client\_role\_users | * Add a new row for the user with the above role id. |
| 3 | MySQL: edxapp | student\_anonymoususerid | * Check Is any entry for the user exist with the anonymous\_user\_id, user\_id and course\_id is blank? If no then add a new row, otherwise skip this step. |
| 4 | MySQL: edxapp | student\_courseenrollment | * Add a new entry with is\_active=1 and mode = “audit”. |

***1. List of fields of the “django\_comment\_client\_role” table in the “edxapp” MySQL database those are used in the Course Enrollment page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **fields of the “*django\_comment\_client\_role*”** | **datatype of fields** | **Label on the Course Enrollment page.** | **Notes, If any.** |
| 1 | id | int(11) | -Not on page but used by the application. | It is auto\_increment and primary key will be used as reference for the role id. |
| 2 | name | varchar(20) | -Not on page but used by the application. | Only the “Student” role id is useful in this application. |
| 3 | course\_id | varchar(250) | course |  |

***2. List of fields of the “django\_comment\_client\_role\_users” table in the “edxapp” MySQL database those are used in the Course Enrollment page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **fields of the “django\_comment\_client\_role\_users”** | **datatype of fields** | **Label on the Course Enrollment page.** | **Notes, If any.** |
| 1 | id | int(11) | -Not on page but used by the application. | It is auto\_increment and primary key. |
| 2 | role\_id | int(11) | -Not on page but used by the application. | It is foreign key reference from the ‘id’ field of `django\_comment\_client\_role` table. |
| 3 | user\_id | int(11) | -Not on page but used by the application. | It is foreign key reference from the ‘id’ field of `auth\_user` table. |

***3. List of fields of the “student\_anonymoususerid” table in the “edxapp” MySQL database those are used in the Course Enrollment page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **fields of the “student\_anonymoususerid”** | **datatype of fields** | **Label on the Course Enrollment page.** | **Notes, If any.** |
| 1 | id | int(11) | -Not on page but used by the application. | It is auto\_increment and primary key. |
| 2 | anonymous\_user\_id | varchar(32) | -Not on page but used by the application. |  |
| 3 | course\_id | varchar(255) | course |  |
| 4 | user\_id | int(11) | -Not on page but used by the application. | It is foreign key reference from the ‘id’ field of `auth\_user` table. |

***Algorithm used to generate anonymous\_user\_id:***

*# include the secret key as a salt, and to make the ids unique across different LMS installs.*

* *hasher = hashlib.md5()*

*# SECRET\_KEY is the “*SECRET\_KEY” of IITBombayX available in the /edx/app/edxapp/lms.auth.json" file.

* *hasher.update(SECRET\_KEY.encode('utf-8'))*
* *hasher.update(text\_type(user\_obj.id).encode('utf-8'))*
* *anonymous\_user\_id = hasher.hexdigest()*

**For code refer:** [*anonymous\_user\_id*](http://gitlab.cse.iitb.ac.in/saritat/iitbombayxregistrationpage/-/blob/Ironwood.master/anonymous_user_id.py)*.py*

***4. List of fields of the “student\_courseenrollment” table in the “edxapp” MySQL database those are used in the Course Enrollment page:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **fields of the “*student\_courseenrollment*”** | **datatype of fields** | **Label on the Course Enrollment page.** | **Notes, If any.** |
| 1 | id | int(11) | -Not on page but used by the application. | It is auto\_increment and primary key. |
| 2 | course\_id | varchar(255) | course |  |
| 3 | is\_active | tinyint(1) | -Not on page but used by the application. | Static Value “1”. |
| 4 | mode | varchar(100) | -Not on page but used by the application. | Static Value "audit". |
| 5 | user\_id | int(11) | -Not on page but used by the application. | It is foreign key reference from the ‘id’ field of `auth\_user` table. |

***About the*** *course mode/*enrollment track*:*

*Also called certificate type, enrollment track, course seat, course track, course type, enrollment mode, or seat type.*

*The enrollment track specifies the following items about a course.*

* The type of certificate, if any, that learners receive if they pass the course.
* Whether learners must verify their identity to earn a certificate, using a webcam and a photo ID.
* Whether the course requires a fee.
* audit: This is the default enrollment track when learners enroll in a course. This track does not offer certificates, does not require identity verification, and does not require a course fee.
* professional: This enrollment track is only used for specific professional education courses. The professional enrollment track offers certificates, requires identity verification, and requires a fee. Fees for the professional enrollment track are generally higher than fees for the verified enrollment track. Courses that offer the professional tracks do not offer a free enrollment track.
* verified: This enrollment track offers verified certificates to learners who pass the course, verify their identities, and pay a required course fee. A course that offers the verified enrollment track also automatically offers a free non-certificate enrollment track.
* honor: This enrollment track was offered in the past and offered an honor code certificate to learners who pass the course. This track does not require identity verification and does not require a fee. Note, however, that as of December 2015, edx.org no longer offers honor code certificates.

***Instruction to Install and Run package***

1. **Setting up a new environment:**

Before we do anything else we'll create a new virtual environment, using venv. This will make sure our package configuration is kept nicely isolated from any other projects we're working on.

* 1. **To create a new environment:**

*python3 -m venv <env\_name>*

* 1. **Activate created environment:**

*source <env\_name>/bin/activate*

1. **Copy the package:**

Now that we're inside a virtual environment, we can copy our package.git clone

* 1. **Git Clone:**

*git clone http://<user\_name>@gitlab.cse.iitb.ac.in/saritat/iitbombayxregistrationpage.git*

* 1. **Change Directory:**

*cd iitbombayxregistrationpage/*

* 1. **Checkout Git branch**

*git checkout Ironwood.master*

1. **Package Settings:**
   1. **In edxapp MySQL Database of the IITBombayX:**
      1. Use *'*edxapp*'* database:

*mysql -u root -p*

Use *'*edxapp*'* *;*

* + 1. Create a user with all privileges granted:

*create user '<db\_user\_name>'@'<api\_server\_ip>' IDENTIFIED BY '<db\_password>';*

*GRANT ALL PRIVILEGES ON \*.\* TO '<db\_user\_name>'@'<api\_server\_ip>' WITH GRANT OPTION ;*

*FLUSH PRIVILEGES;*

* 1. ***Change database setting of IITBombayXRegistrationPage package:***

We'll need to edit our mysql user credential to ‘DATABASES’ dictionary . Let's edit the './*IITBombayXRegistrationPage*/settings.py' file:

*vi ./IITBombayXRegistrationPage/settings.py*

**For example:**

*DATABASES = {*

*'default': {*

*'ENGINE': 'django.db.backends.mysql',*

*'NAME': 'edxapp',*

*'USER': '****<db\_user\_name>****',*

*'PASSWORD':'****<db\_password>****',*

*'HOST':'****<IITBombayX\_MySQL\_IP>****',*

*'PORT':'****<IITBombayX\_MySQL\_PORT>****',*

*}*

*}*

* 1. ***Package Server Setting:***

*We'll need to add our server IP to 'ALLOWED\_HOSTS' list. Let's edit the 'edXRegistration/settings.py' file:*

*vi edXRegistration/settings.py*

**For example:**

ALLOWED\_HOSTS = ['127.0.0.1']

1. ***Install package requirements*:**

We can install our package requirements inside a virtual environment.

*pip install -r requirement.txt*

1. ***IITBombayX Server*** ***Configurations:***

Now, we set IITBombayX server credentials of Mongodb, MySql databases and EDXAPP\_EDXAPP\_SECRET\_KEY in the 'config.yml' file, let’s edit it

*vi config.yml*

*Note: The “*EDXAPP\_EDXAPP\_SECRET\_KEY” is the key of IITBombayX copied from the file /home/edx/my-passwords.yml.

1. ***IITBombayX Login Server IP:***

vi registration/templates/navbar.html

**For example:**

<li class="nav-item navbar-right">

<a class="nav-link active" href="http://<IITBombayX\_IP>/login?next=%2F">Login</a>

</li>

1. **Run server:**

Let us start up the development web server:

*python manage.py runserver <server\_IP>:<port>*

***Test cases***

***Test cases of the Sample Custom Registration page for the IITBombayX Ironwood.Master Instance:***

1. In the browser hit *<server\_IP>:<port> URL.*
2. Click on“Register” menu on the top-right side of the page.
3. Fill registration page information of new user and click on “Create Account” Button at bottom of the page.

**Case 1:** Age should be more or equal to the 13.

**Case 2:**  Age should be less than the 13.

To verify:

1. Click on“Login” menu on top-right side of page, which is used to login on the Ironwood.master instance using the newly created user’s credentials.
2. Verify the user’s all profile details in the user dashboard on the Ironwood.master instance.
3. Check the user's profile details on the instructor dashboard.
4. Check the user's account details on the instructor dashboard.
5. Enroll the login user for a course.
6. Sign-Out the login user from IITBombayX platform.

***Test cases of the Course Enrollment page for the IITBombayX Ironwood.Master Instance:***

1. In the browser hit *<server\_IP>:<port> URL.*
2. Click on“Enroll” menu on top-left side of page.
3. Select a Learner from the drop-down learner list.
4. Select a Course from the drop-down course list.
5. Click on Enroll Button.

To verify:

1. Click on“Login” menu on top-right side of page, which is used to login on the Ironwood.master instance using the user’s credentials.
2. Click on the User Dashboard on the IITBombayX Ironwood.master platform.
3. Check the “My Courses” list, Confirm the course available in the list.