yadawia Documentation

Release 0.1

Mariam Maarouf

May 06, 2017

Contents:

CHAPTER 1

Introduction

Built for CCIT's E-Business Fundementals course (IS371). Its idea is simple: customer-to-customer buying and selling of goods and services, if and only if those goods and services in question are handmade.

This documentation focuses on the technical side of things rather than the business side.

Quick summary of what people should be able to do with this platform:

- Create, edit, and deactivate accounts.
- Sell handmade products.
- Buy handmade products.
- Communicate special requests to sellers.
- Search for products by their attributes.

The difference between this and a regular e-commerce platform to the user is the flexibility in making and receiving orders regarding schedules, requests, etc.

Logo Credit.

CHAPTER 2

Tools

This project uses:

- Flask (Python) for the back-end.
- AWS S3 for image uploads.
- PostgreSQL for the database.

CHAPTER 3

Installation

1. Clone the github repo.

```
$ git clone https://github.com/blaringsilence/yadawia.git
$ cd yadawia
```

2. Install virtualenv and activate it.

```
$ pip install virtualenv
$ virtualenv -p python3 venv
$ . venv/bin/activate
```

3. Install requirements.

```
$ pip install -r requirements.txt
```

4. Install PostgreSQL (see download page) and create a user and a database.

```
$ sudo -u postgres createuser -s $USER -P
$ createdb -U $USER snowdonia
```

5. Set environment variables (if running locally, add them to your venv/bin/activate script as follows):

```
export DATABASE_URL="postgresql+psycopg2://USER:PASSWORD@localhost/yadawia"
export AWS_ACCESS_KEY_ID="YOUR/YOUR IAM USER'S AWS ACCESS KEY"
export AWS_SECRET_ACCESS_KEY="YOUR/YOUR IAM USER'S SECRET ACCESS KEY"
export S3_BUCKET="YOUR BUCKET NAME HERE"
```

6. Create the tables, then populate them with the initial values (chmod a+x FILENAME before executing if file does not have execution permissions).

```
$ ./create_db.py
$ ./populate_countries.py
$ ./populate_currencies.py
$ ./populate_reasons.py
```

```
$ ./populate_categories.py
$ ./populate_methods.py
```

7. Run the app.

```
$ gunicorn yadawia:app
```

CHAPTER 4

Docs

App

Initialize configuration for the whole app and put all the parts together.

yadawia.**app = <Flask 'yadawia'>**
Initialize app.

yadawia.**assets = <flask_assets.Environment object>**
Initialize assets in app.

yadawia.**country_name_filter**(*country_id*)
TEMPLATE FILTER: Get country name from country ID.

yadawia.**csrf_protect**()
Check for csrf token in POST and DELETE requests. CSRF tokens are generated per session/login. If there's no token or the token is not equal to the one from the form, abort the request.

yadawia.**css_libs = <Bundle output=css/all.css, filters=[<webassets.filter.cssmin.CSSMin object>], contents=('css/layout.css**
Bundle library css files and minify them.

yadawia.**db**
Use SQLAlchemy as an ORM.

yadawia.**excerpt**(*text*)
TEMPLATE FILTER: Cut text and append dots if its lenght is more than 100 chars.

yadawia.**js = <Bundle output=js/all.js, filters=[<webassets.filter.slimit.Slimit object>], contents=('js/form_edit_profile.js', 'js/**
Bundle all JavaScript files and minify them.

yadawia.**jsglue = <flask_jsglue.JSGlue object>**
Initialize Flask-JSGlue which allows us to use Flask.url_for in un-rendered JavaScript.

yadawia.**name_or_username**(*userId*)
TEMPLATE FILTER: Get someone's name if set, if not then username.

yadawia.**order_total**(*order*)
TEMPLATE FILTER: Get order total given the order object.

yadawia.**paragraph**(*text*)
TEMPLATE FILTER: Replace newlines with
 in html.

yadawia.**sender**(*thread_id*)
TEMPLATE FILTER: Get sender in a 2-person message from threadID and logged in session.

Views

Contains all the view logic/endpoints for this app.

yadawia.views.**add_address**()
Add a new address to a logged in user's account.

yadawia.views.**cart**()
View function for cart.

yadawia.views.**cart_products**()
AJAX endpoint for cart pages: get info about products in cart.

yadawia.views.**create_product**()
Create a new product to sell by the logged in user.

yadawia.views.**deactivate_account**()
Deactivate a user's account. Can be undone by simply logging in again, as opposed to admin suspension (yet to be implemented in a view).

yadawia.views.**delete_address**()
Delete an address from a logged in user's account.

yadawia.views.**delete_review**(*productID*)
Delete a previously written review.

yadawia.views.**edit_product**(*productID*)
Edit product (edit non-image attributes and add new images).

yadawia.views.**edit_product_pics**(*productID*)
Edit product pictures (remove or re-order).

yadawia.views.**edit_profile**()
Edit profile (name, username, location) of logged in user.

yadawia.views.**edit_review**(*productID*)
Edit a previously written review.

yadawia.views.**home**()
View function for home.

yadawia.views.**login**()
View function for Login.

yadawia.views.**logout**()
Log user out.

yadawia.views.**message_thread**(*threadID*)
View for single message thread.

yadawia.views.**messages**()
View for message threads.

yadawia.views.**new_message**()
Send a new message (start a new message thread) with another user.

yadawia.views.**new_review**(*productID*)
Review a product (only available to logged in users).

yadawia.views.**order_history**()
Get someone's order history. Orders by them, and orders for them.

yadawia.views.**privacy**()
View function for privacy policy.

yadawia.views.**product**(*productID*)
View for product page given the product ID.

yadawia.views.**profile**(*username=None*)
View function for profile given a username OR if not, use username of logged in user. Returns:

- user: Object instance of type User.
- is_curr_user: Boolean = is this the logged in user's profile?
- avg_rating: user's average rating throughout their products.
- products: user's products (all if is_curr_user, available only if not).
- report_reasons: reasons to report a user if not is_curr_user.

yadawia.views.**register**()
View function for Registeration.

yadawia.views.**reply**(*threadID*)
Reply to a message thread.

yadawia.views.**report_user**()
Report a user to platform admins.

yadawia.views.**search_category**(*categoryID*)
Search in a category

yadawia.views.**search_products**()
Search products using a single query line, and a sort order/parameter.

yadawia.views.**see_message**(*threadID*)
Mark a message as seen by receiver.

yadawia.views.**settings**()
View for settings for logged in user.

yadawia.views.**sign_s3**()
Sign a request to upload to the S3 Bucket.

yadawia.views.**terms**()
View function for terms and conditions.

yadawia.views.**toggle_availability**(*productID*)
Toggle a product's availability.

yadawia.views.**update_account**()
Edit account (email, password) of logged in user.

yadawia.views.**upload_avatar**()
Upload profile picture for logged in user.

yadawia.views.**validate_field**()
Check availability of username/email. For use in registeration form.

Classes

Contains all the classes (database, exceptions, etc) created for this app.

class yadawia.classes.**Address**(*name*, *user_id*, *text*, *country_id*, *code=None*, *phone=None*)
Database model for addresses (physical). Contains:

- id: int, auto-incremented.
- name: name to assign to this address (every user can have multiple addresses).
- user_id: int, foreign key.
- country_id: string, ISO 3166-1 code, foreign key.
- city: string.
- zip/postal code: string.
- phone: string.
- text: string.

validate_name(*key*, *name_input*)
Makes sure the name doesn't have any numbers or special chars. Raises a DBException otherwise.

class yadawia.classes.**Admins**(*user_id*)
Database model for admins. Contains:

- id: int, auto-incremented.
- user_id: int, foreign key.
- date: date.

class yadawia.classes.**Category**(*name*)
Database model for categories. Contains:

- id: int, auto-incremented.
- name: string.

validate_name(*key*, *name_input*)
Makes sure the name doesn't have any numbers or special chars. Raises a DBException otherwise.

class yadawia.classes.**Country**(*country_id*, *value*)
Database model for all countries. Contains: - id: string, ISO 3166-1 code. - value: string, name.

class yadawia.classes.**Currency**(*curr_id*, *name*, *symbol=None*)
Database model for all supported currencies. Contains:

- id: string, ISO-4217 code.
- name: string.
- symbol: string.

exception yadawia.classes.**DBException**
Custom exceptions raised on the ORM level. In its 0th arg, has a human-readable message and a code.

class yadawia.classes.**Featured**(*product_id*, *notes=None*)
Database model for featured products. Contains:

- id: int, auto-incremented, primary key.
- product_id: id for the featured product.

- date: date it was featured.
- notes: editor/admin's notes on why they featured it.

exception yadawia.classes.**LoginException**
Custom exceptions raised on when logging in. In its 0th arg, has a human-readable message and a code.

class yadawia.classes.**Message**(*thread_id*, *sender_id*, *text*)
Database model for messages in a thread. Contains:

- id: int, auto-incremented.
- thread_id: int, foreign key.
- sender_id: int, foreign key. No need for receiver because thread has info.
- date: date.
- text: string.
- seen: date.

see(*userId*)
Mark the message as seen by the receiver.

class yadawia.classes.**MessageThread**(*user1*, *user2*, *title=None*)
Database model for message threads. Contains:

- id: int, auto-incremented.
- user1: int, foreign key.
- user2: int, foreign key.
- title: string. Optional.
- order_id: int, foreign key (if about an order).

getTitle()
Get thread title or 'Untitled Thread' if no title.

isParticipant(*user*)
Check if a user is a participant in a thread.

otherUser(*user*)
Given a user in a thread, find the other one.

unseen(*user*)
Get number of unseen messages relative to a user.

class yadawia.classes.**Order**(*user_id*, *address_id=None*, *payment_method_id=None*)
Database model for orders. Contains:

- id: int, auto-increment.
- user_id: int, foreign key.
- create_date: date.
- update_date: date.
- status: string. (New, Ongoing, All confirmed, On its way, Done)
- address_id: int, foreign key.
- payment_method_id: int, foreign key.

touch()
Trigger the update.

updateStatus()
Update status based on confirmation of orders/shipping (shipping not implemented yet).

class yadawia.classes.**OrderProduct**(*order_id*, *product_id*, *price*, *currency_id*, *variety_id=None*, *quantity=1*, *remarks=None*)
Database model for relationship between orders and models (many-to-many). Contains:

- id: int, auto-incremented.
- order_id: int, foreign key.
- product_id: int, foreign key.
- variety_id: int, foreign key.
- quantity: int.
- create_date: date.
- update_date: date.
- remarks: string.
- price: price at time of checkout.
- currency_id: currencyID at time of checkout.
- confirmed: boolean: is this item confirmed by the seller?

validate_quantity(*key*, *q*)
Validate that quantity is more than 0.

class yadawia.classes.**PaymentMethod**(*name*, *fee=0*, *isPercentFee=False*, *currency_id=None*)
Database model for payment methods. Contains:

- id: int, auto-incremented.
- name: string (Cash on delivery, etc).
- fee: float (extra fee taken to use this method).
- isPercentFee: Boolean (is the fee a percentage? true = yes, false = fee is flat)
- currency_id: currency for fee if it's flat

helperText()
Text to explain the payment method according to its fees.

class yadawia.classes.**Product**(*name*, *seller_id*, *description=None*, *price=None*, *currency_id=None*)
Database model for products. Contains:

- id: int, auto-incremented.
- name: string.
- seller_id: int, foreign key.
- update_date: date.
- create_date: date.
- description: string.
- price: float.
- available: boolean, default: True.

first_picture()
Get the first picture of a product (by order set in upload).

validate_price(*key*, *p*)
Makes sure the price is not less than 0.

class yadawia.classes.**ProductCategory**(*product_id*, *category_id*)
Database model for the relationship between products and categories (many-to-many). Contains:

- product_id: int, foreign key.
- category_id: int, foreign key.

class yadawia.classes.**Reason**(*text*)
Database model for report reasons. Contains:

- id: int, auto-incremented.
- text: string.

class yadawia.classes.**Report**(*sender_id*, *about_id*, *reason_id*, *message*)
Database model for reports to admins. Contains:

- id: int, auto-incremented.
- sender_id: int, foreign key.
- about_id: int, foreign key.
- reason: int, foreign key.
- date: date.

getAbout()
Return user this report is about.

getSender()
Return sender of the report.

class yadawia.classes.**Review**(*user_id*, *product_id*, *rating*, *title=None*, *text=None*)
Database model for reviews on products. Contains:

- id: int, auto-increment.
- user_id: int, foreign key.
- product_id: int, foreign key.
- rating: float.
- title: string.
- text: string.
- create_date: review date.
- update_date: update date.

validate_rating(*key*, *r*)
Makes sure the rating is between 1 and 5 with 0.5 increments only. Raises DBException otherwise.

class yadawia.classes.**Upload**(*filename*, *product_id*, *variety_id=None*, *order=0*)
Database model for product-related uploads (photo, video). Contains:

- id: int, auto-increment.
- filename: string.

- date: upload date.
- product_id: int, foreign key.
- variety_id: int, foreign key. Optional.
- order: int, default: 0

class yadawia.classes.**User**(*username*, *email*, *password*, *name=None*, *location=None*)

Database model for users. Contains:

- id: int, auto-incremented.
- username: string.
- name: string. Optional.
- email: string.
- password: string.

bought(*productID*)

Check if the user bought a certain product.

isPassword(*pw*)

Check if a string matches the stored password hash.

name_or_username()

Return name if name is set, otherwise return username.

validate_email(*key*, *em*)

Validate that the email has an @ and characters before and after it. Raises a DBException if invalid.

validate_location(*key*, *loc*)

Validate that the location contains anything but special characters. Raises a DBException if invalid.

validate_name(*key*, *name_input*)

Validate that the name contains anything but numbers and special characters. Raises a DBException if invalid.

validate_password(*key*, *pw*)

Validate that the password is at least 6 chars long. Raises a DBException if not.

validate_username(*key*, *usr*)

Validates that the username begins with a letter, is at least 2 chars long, and can only ever contain letters, numbers, or underscores. Raises a DBException if invalid.

class yadawia.classes.**Variety**(*name*, *product_id*, *price=None*, *available=True*)

Database model for varieties in products (sizes, etc). Contains:

- id: int, auto-incremented.
- product_id: int, foreign key.
- name: string. What is this variety? (e.g. Size small) No validation.
- price: float.
- available: boolean, default: True.

validate_price(*key*, *pr*)

Make sure price is not less than 0. Raises a DBException otherwise.

Helpers

Contains helper functions (decorators, others) used by other parts of the app.

yadawia.helpers.**anonymous_only**(*f*)
Decorator function to ensure user is NOT logged in before a page is visited.

yadawia.helpers.**assetsList**(*app*, *folder='js'*, *extension='js'*, *exclusions=[]*)
Get list of files of a specific extension in a folder inside the static directory.

yadawia.helpers.**authenticate**(*f*)
Decorator function to ensure user is logged in before a page is visited.

yadawia.helpers.**create_edit_product**(*create=True*, *productID=None*)
Function to create or edit a product (used in views).

yadawia.helpers.**curr_user**(*username*)
True if this username is that of the logged in user, false otherwise.

yadawia.helpers.**disable_user**(*username*, *suspended=False*)
Disable a user.

yadawia.helpers.**enable_user**(*username*, *was_suspended=False*)
Enable a user.

yadawia.helpers.**generate_csrf_token**(*force=False*)
Create a CSRF-protection token if one doesn't already exist in the user's session (or force it, as done per login) and put it there.

yadawia.helpers.**get_presigned_post**(*filename*, *filetype*)
Use boto3 the AWS Python SDK to generate a presigned post for S3.

yadawia.helpers.**get_random_string**(*length=32*)
Generate a random string of length 32, used in `generate_csrf_token()`

yadawia.helpers.**get_upload_url**(*filename*)
Return url to uploaded file.

yadawia.helpers.**is_allowed_in_thread**(*threadID*)
Given a threadID, is the signed in user allowed in the thread?

yadawia.helpers.**is_logged_in**()
Is the user logged in?

yadawia.helpers.**is_safe**(*url*)
Is the URL safe to redirect to?

yadawia.helpers.**login_user**(*username*, *password*)
Function to login user through their username. Sets:

- session['logged_in'] to True.
- session['username'] to the username.
- session['userId'] to the user ID.

Raises LoginException (represented as e here) if:

- User with that username does not exist (e.args[0]['code'] = 'username')
- Password is incorrect (e.args[0]['code'] = 'password')

yadawia.helpers.**logout_user**()
Log user out.

yadawia.helpers.**no_special_chars**(*string*, *allowNumbers=False*, *optional=True*, *allowComma=False*)
Function to check if a string has no special characters.

yadawia.helpers.**public**(*obj*, *keys*)
Pass a db class object and a list of keys you don't want returned (e.g. password hash, etc) and get a filtered dict.

yadawia.helpers.**redirect_back**(*endpoint*, ***values*)
Helper function to redirect to 'next' URL if it exists. Otherwise, redirect to an endpoint.

yadawia.helpers.**suspend_user**(*username*)
Suspend a user.

yadawia.helpers.**unsuspend_user**(*username*)
Unsuspend a user.

yadawia.helpers.**user_exists**()
Given user is logged in, do they exist in db?

yadawia.helpers.**valid_photo**(*photo_type*, *photo_size*)
Given a photo_type and a photo_size, make sure it's an image under MAX_PHOTO_SIZE in app.config

yadawia.helpers.**validate_name_pattern**(*name_input*, *allowNumbers=False*, *optional=True*)
Validate name pattern, given that generally names do not have special chars.

Error handlers

How to handle common HTTP errors.

yadawia.errorhandlers.**bad_request**(*e*)
Render error template with the message: Bad Request and no details.

yadawia.errorhandlers.**forbidden**(*e*)
Render error template with the message: Forbidden and no details.

yadawia.errorhandlers.**gone**(*e*)
Render error template with the message: Page Gone and no details.

yadawia.errorhandlers.**internal_error**(*e*)
Render error template with the message: Internal Server Error and no details.

yadawia.errorhandlers.**not_found**(*e*)
Render error template with the message: Page Not Found and details.

yadawia.errorhandlers.**render_error**(*code*, *msg*, *det='Oops..'*)
Render template with variables appropriate to the error provided.

yadawia.errorhandlers.**unauthorized**(*e*)
Render error template with the message: Unauthorized and details.

Python Module Index

y

Index