

Bargain_box

Class	Responsibilities	Collaborators
asgi.py	<ul style="list-style-type: none">Entry point for compatible servers to serve the django application asynchronouslyEnables asynchronous support(such as support for running background tasks)	<ul style="list-style-type: none">settings.pyASGI server
settings.py	<ul style="list-style-type: none">Configures the projects settings (database connections, installed apps, middleware, static files paths)Sets configurations for different environments	<ul style="list-style-type: none">urls.pywsgi.pyasgi.pyDjango settings framework import
urls.py	<ul style="list-style-type: none">Defines the URL routing patterns, which directs requests to specific views and specific appsManages and includes URLs from other apps	<ul style="list-style-type: none">settings.py
wsgi.py	<ul style="list-style-type: none">Entry point for compatible web servers for the django applicationConfigures app environment	<ul style="list-style-type: none">settings.pyWSGI server

about

Class	Responsibilities	Collaborators
AboutConfig (in apps.py)	<ul style="list-style-type: none">Configure settings for home appDefine primary key behaviorSpecify the apps name and app-specific settingsRegisters the app in Django's framework	<ul style="list-style-type: none">Django app import
urls.py	<ul style="list-style-type: none">Defines URL routes for the About section pages, such as "Meet the Team," "Our Mission," and "How We Work."	<ul style="list-style-type: none">views.Meet_the_teamviews.How_we_workviews.Our_mission
Meet_the_team (in views.py)	<ul style="list-style-type: none">Handles the request for the "Meet the Team" page.Displays team member details like name, linkedin, and bio.	<ul style="list-style-type: none">urls.py
How_we_work (in views.py)	<ul style="list-style-type: none">Handles the request for the "How We Work" page.Displays information about how the company works.	<ul style="list-style-type: none">urls.py
Our_mission (in views.py)	<ul style="list-style-type: none">Handles the request for the "Our Mission" page.Displays the company's mission, vision, and	<ul style="list-style-type: none">urls.py

	core values.	
--	--------------	--

home

Class	Responsibilities	Collaborators
HomeConfig (in apps.py)	<ul style="list-style-type: none"> Configure settings for home app Define primary key behavior Specify the apps name and app-specific settings Registers the app in Django's framework 	<ul style="list-style-type: none"> Django app import
Home_page (in views.py)	<ul style="list-style-type: none"> Handle HTTP requests for home page Renders necessary data in the home.html templates 	<ul style="list-style-type: none"> Home.html foundation.html
Foundation and home html (in templates folder)	<ul style="list-style-type: none"> Template for the app, providing structure to the page like navigation top bar, search bar, ect. Also includes some scripts (for the search bar functionality) 	<ul style="list-style-type: none"> Home.html Foundation.html Static files (css)

post

Class	Responsibilities	Collaborators
PostConfig (in apps.py)	<ul style="list-style-type: none"> Configure settings for post app Define primary key behavior Specify the apps name and app-specific settings Registers the app in Django's framework 	<ul style="list-style-type: none"> Django app import
Post (in models.py)	<ul style="list-style-type: none"> Define structure of post (title, author, content, etc) Manage post validation Handle bookmarks with ManyToManyField for user association. Provide absolute URL for redirection after creation. 	<ul style="list-style-type: none"> PostListView PotDetailView PostUpdateView PostCreateView PostDeleteView User model
urls.py	<ul style="list-style-type: none"> Defines the url routes for viewing listings, viewing details of the post, creating, updating, and deleting posts, adding and removing bookmarks, and viewing bookmarks as well. Defines the mapping from each url to each view 	<ul style="list-style-type: none"> All Post Views
PostListView(in views.py)	<ul style="list-style-type: none"> Displays all the posts in a list, as well as search feature view listings 	<ul style="list-style-type: none"> Post model PostDetailView

	<ul style="list-style-type: none"> • Renders the view to the correct html template 	<ul style="list-style-type: none"> • User model • postlistview.html
PostDetailView(in views.py)	<ul style="list-style-type: none"> • Displays a specific posts details (title, author, content, etc) • Renders the details in the correct view template 	<ul style="list-style-type: none"> • Post model • User Model • postdetailview.html
PostUpdateView(in views.py)	<ul style="list-style-type: none"> • Handles the editing of an existing post. • Displays the form with current data that is editable • Validates any new data if the user changes something, and saves the updates • Redirects to success page upon success 	<ul style="list-style-type: none"> • Post model • User Model • Postupdateview.html
PostCreateView(in views.py)	<ul style="list-style-type: none"> • Handles the creation of new posts. • Displays empty form for users to fill • Validates the data and saves the data to the database • Redirects to success page upon success 	<ul style="list-style-type: none"> • Post model • User Model • postcreateview.html
PostDeleteView(in views.py)	<ul style="list-style-type: none"> • Handles deletion of user posts, only made by the logged in user • Takes user to “confirm deletion” page before deletion occurs • Upon deletion, success message displayed 	<ul style="list-style-type: none"> • Post model • User Model • postdeleteview.html
AddingBookmarkView (in views.py)	<ul style="list-style-type: none"> • Handle adding bookmarks to posts for logged-in users. • Redirect to home upon success. 	<ul style="list-style-type: none"> • Post model • User Model • home.html
BookmarkView (in views.py)	<ul style="list-style-type: none"> • Display a list of all posts bookmarked by the user. • Fetch and render bookmarked posts to bookmarked_posts.html. 	<ul style="list-style-type: none"> • Post model • User model • bookmarked_posts.html
RemovingBookmarkView (in views.py)	<ul style="list-style-type: none"> • Handle removal of bookmarks from posts. • Redirect to home upon success. 	<ul style="list-style-type: none"> • Post model • User model • home.html

user_authentication

Class	Responsibilities	Collaborators
userAuthenticationConfig (in apps.py)	<ul style="list-style-type: none"> • Configure settings for userauthentication app • Define primary key behavior • Registers the app in Django's framework 	<ul style="list-style-type: none"> • Django app import
user_account_registration (in views.py)	<ul style="list-style-type: none"> • Handles the http requests for user registration • Validates the users data, making sure all fields are correct including strong password, valid 	<ul style="list-style-type: none"> • User model • UserRegistrationForm • User_account_signout

	email, unique username	
user_account (in views.py)	<ul style="list-style-type: none"> Display users account details Fetch users details and organize on page according to the profile view template 	<ul style="list-style-type: none"> User model
User_account_signout (in views.py)	<ul style="list-style-type: none"> Handles user sign out functions Upon success, success message displayed 	<ul style="list-style-type: none"> Authentication framework user_account_registration
PasswordResetView (in views.py)	<ul style="list-style-type: none"> Handles the password reset form rendering and processing 	<ul style="list-style-type: none"> Django authentication framework
PasswordResetEmailSent View (in views.py)	<ul style="list-style-type: none"> displays a confirmation page after the password reset email is sent 	<ul style="list-style-type: none"> Django authentication framework
PasswordResetConfirmView (in views.py)	<ul style="list-style-type: none"> displays the form to reset the password using a token 	<ul style="list-style-type: none"> Django authentication framework
PasswordResetComplete View (in views.py)	<ul style="list-style-type: none"> displays a confirmation page once the password is successfully reset 	<ul style="list-style-type: none"> Django authentication framework

user_posts

Class	Responsibilities	Collaborators
userPostsConfig (in apps.py)	<ul style="list-style-type: none"> Configure settings for user posts app Define primary key behavior Registers the app in Django's framework 	<ul style="list-style-type: none"> Django app import
urls.py	<ul style="list-style-type: none"> Defines the url routes for accessing the posts including listing and detailed views. Defines the mapping from each url to each view for the user posts app 	<ul style="list-style-type: none"> view_posts
View_posts (in views.py)	<ul style="list-style-type: none"> Displays list of posts Fetches and presents the data of the post loaded into the correct template for users to view 	<ul style="list-style-type: none"> Post model urls.py

user_profile

Class	Responsibilities	Collaborators
userProfileConfig (in apps.py)	<ul style="list-style-type: none"> Configure settings for user_profile app Define primary key behavior Specify the apps name and app-specific settings 	<ul style="list-style-type: none"> Django app import

UpdateUserTableForm (in forms.py)	<ul style="list-style-type: none"> Defines the form for updating the users information such as name and email. Validates the user's new data, and saves any updates to the database 	<ul style="list-style-type: none"> Profile model
UpdateProfileTableForm (in forms.py)	<ul style="list-style-type: none"> Provides form for updating the profile picture Validates the user's new profile data and saves any updates to the database 	<ul style="list-style-type: none"> Profile model
Profile (in models.py)	<ul style="list-style-type: none"> Defines the profiles database table that links a user to their profile image Relates the profile to the user using a one-to-one relationship 	<ul style="list-style-type: none"> updateUserTableForm Signals.py
signals.py	<ul style="list-style-type: none"> Handle the creation of a user profile when a new user is created Handles deletion of user profile resources such as profile picture (the user deletes them) 	<ul style="list-style-type: none"> Django signals Profile model
urls.py	<ul style="list-style-type: none"> Routes the URLs to profile related views like viewing and editing the profile 	<ul style="list-style-type: none"> views.view_profile views.edit_profile views.delete_profile
view_profile (in views.py)	<ul style="list-style-type: none"> Handles the HTTP GET request to display the profile page of the user Renders and fetches the users profile data (only for logged in users) Lists active users listings 	<ul style="list-style-type: none"> User model Profile model Post model
edit_profile (in views.py)	<ul style="list-style-type: none"> Handles the HTTP GET request by provides the form for editing a logged in user's profile Handles the HTTP POST request by validating the form to make sure information is correct, and saves the updates. Also includes editing profile picture email and username 	<ul style="list-style-type: none"> User model Profile model updateUserTableform updateProfileTableForm
delete_profile (in views.py)	<ul style="list-style-type: none"> Handles the HTTP GET request to display the delete profile page of the user Handles the HTTP DELETE request which in turn logs the user out and deletes their records from the database 	<ul style="list-style-type: none"> User model Profile model

user_registration

Class	Responsibilities	Collaborators
User_registration_config	<ul style="list-style-type: none"> Configure settings for user_registration app 	<ul style="list-style-type: none"> Django app import

(in apps.py)	<ul style="list-style-type: none"> • Define primary key behavior • Specify the apps name and app-specific settings 	
register (in views.py)	<ul style="list-style-type: none"> • Render the user registration form • Process the user registration form • Validate user_data with a successful form • Handles GET requests to show registration page • Handles POST requests to process the data from the form 	<ul style="list-style-type: none"> • UserRegisterForm • User Model (creates user) • Messages (for message feedback (success,fail))
UserRegistrationForm (in forms.py)	<ul style="list-style-type: none"> • Defines the fields in the form and the order in which they appear in registration (username, email, password) • Validates user input, making sure password is strong, email is valid, username not taken. • Handle data cleaning and transformation before saving to the database 	<ul style="list-style-type: none"> • User Model • Register (View) • Django forms import