**Overview:**
The goal of the project is to create a website that five college students can use to buy, sell and give away items. It would aim to serve the Amherst community specifically, for example selling books for classes that are specific to these colleges. The website aims to verify users both via university email and through user reviews.

**Team Members:**

- Vedant Puri
- Shubham Mehta
- Sagar Thapar
- Zachary Kiihne
- Yifan Zhu

**Github Repository:** https://github.com/vedantpuri/ffs-web
**Design Overview:**

- Data models: Our project consists of 3 models.

  The first and foremost is the Product model. This model consists of the following fields: title, description, price, image, owner and status. The names of these fields are quite self-explanatory (e.g. title is the name of the product, image pertains to the picture of the product, etc.). The status field can take 2 values, name “Available” and “Sold.” Additionally, it is important to note that the images of products are stored at ‘ffs-web/static_cdn/media_root/products.’ Thus, whenever a new product image is uploaded, it will be stored in this folder. Furthermore, it is important to note that the owner field is defined as a Foreign Key. We do this because a single owner can have many products, resulting in a one to many relationship.

  Moving forward, let us discuss the User model. The user model has the following fields: first_name, last_name, bio, password, college, email, star_count and image. It is important to note here that we do not have a username field because we will be using the user’s email for his or her username. Also, just as with products, the image field corresponds to the user’s profile picture, which are stored at ‘ffs-web/static_cdn/media_root/user_profile_picture.’ Furthermore, since our application is restricted to the 5-college area, the college field can only take the values ‘UMass Amherst’, ‘Amherst College’, ‘Smith College’, ‘Mount Holyoke College’, and ‘Hampshire College.’

  Lastly, the Flag model consists of 2 fields. The first is the user field, which is defined as a foreign key since a single user can flag many items. Similarly, we have a products field, defined as a many to many field. Only these 2 fields are necessary for this model since we are simply keeping track of the items that have been flagged by a certain user.

- URL routes:

  1. localhost:8000/ corresponds to the home page of our website
  2. localhost:8000/search/ corresponds to the search results page
  3. localhost:8000/flagged/ corresponds to the flagged items page
  4. localhost:8000/upload/ corresponds to the upload product page
  5. localhost:8000/user/ corresponds to the profile view page
  6. localhost:8000/admin/ corresponds to the Django admin site

  These URLs can be accessed via the navigation bar as well.

- UI views:

  Home Page / Landing Page : This is the main page of our website. The search box currently does not take input

  Search Results Page: Currently displays the results for a search for ‘Textbook’

  Upload Product Page: This form is currently inactive

  Flagged Items Page: Currently displays the flagged items for the user whose email is ‘sthapar@umass.edu’

  Profile View Page: Currently displays the profile for the user whose name is ‘Samuel Jackson’

***Important Note:*** Firstly our team name folder in the git repository is CS326TeamH. Secondly, we have not yet included a login/registering page as the project requirement mentioned to not include any forms. However we have a UI for it ready in project1 if you would like to see. But for this project we haven’t redirected the Sign-In/Register button on the nav bar to any view and hence you will get an error if you try to click that.

**Problems/Successes:**

Problems

- Making all the views coherent with each other.
- Configuring the media root properly to store and render the images correctly.
- Updating the database: pulling the code didn’t pull changes from the db.sqlite file.

Successes

- Coordinating and distributing work.
- Creating a proper base template that worked well with all our views.

**Individual Writeups**

***Format:***

***Member name:Contribution %***

***Contribution explained in detail [bullets]***

**Vedant Puri: 20%**

- Fix all the inconsistencies in the UI from project 1. (primarily made a consistent navbar and footer)
- Collaborate to configure django and create models.
- Create the base template (base.html)
- Fix annoying image rendering issue. (in both user page and products)
- Link the tabs on the navigation bar appropriately.
- Collaborated to create and render the user view.
- Fix minor styling issues.
- Managed and monitored the project overall and solved redundancies.
- Update the README

**Yifan Zhu: 20%**

- Recreate upload page and user profile page from project 1.
- Collaborate to create models
- Collaborate to create and render the user view.
- Help to modify base template
- Fix star rating in user view

- Draw UML

**Shubham Mehta: 20%**

- Studied Django in more depth in an effort to add to our team's knowledge base
- Collaborated with team mates to configure Django and create models
- Formatted html to extend base.html
- Collaborated on image rendering and developing search logic
- Worked on url mappings and views of the project
- Developed logic to retrieve flag counts from database

**Sagar Thapar: 20%**

- Created and rendered the flagged items view.
- Collaborated to create and render the models and views for search functionality.
- Collaborated to create search logic.
- Collaborated to create other models as well.
- Fixed minor errors in data entries.
- Formatted html to extend base.html

**Zachary Kiihne: 20%**

- Populated database with 30+ item entries
- Incorporated landing and upload pages into new html framework
- Created URL mapping for landing page and upload page
- Created the write up
- Wrote overview and contributed to successes and failures section of the write up