Preliminary Planning

Brief

Create a web application where the user can create a ticket or upvote an existing ticket to request a change either due to a bug or for a new feature.

If the ticket is requesting a new feature the user must pay sum of money for the request/upvote. , and an index organised by upvotes of each ticket.

Add any additional pages that would help you attract users to the Issue Tracker

Bonus: add additional features, such as a blog, extra perks for active participants, add pages describing your fictional application.

Requirements

- Write a README.md file for your project that explains what the project does and the need that it fulfills. It should also describe the functionality of the project, as well as the technologies used. Detail how the project was deployed and tested and if some of the work was based on other code, explain what was kept and how it was changed to fit your need. A project submitted without a README.md file will FAIL.

- The project must be a brand new Django project, composed of multiple apps (an app for each reusable component in your project).

- The project should include an authentication mechanism, allowing a user to register and login, and there should be a good reason as to why the users would need to do so.

- At least one of your Django apps should contain some e-commerce functionality using Stripe.

- Include at least one form with validation that will allow users to create and edit models in the backend (in addition to the authentication mechanism).

- The project will need to connect to a database (e.g., sqlite or Postgres) using Django's ORM

- The UI should be responsive, use either media queries or a responsive framework such as Bootstrap to make sure that the site looks well on all commonly-used devices.

- As well as having a responsive UI, the app should have a great user experience.
- The frontend should contain some JavaScript logic to enhance the user experience.
- Whenever relevant, the backend should integrate with third-party Python/Django packages, such as Django Rest Framework, etc.
- Make sure to test your project extensively. In particular, make sure that no unhandled exceptions are visible to the users, under any circumstances. Use automated Django tests wherever possible. For your JavaScript code, consider using Jasmine tests.
- Use Git & GitHub for version control. Each new piece of functionality should be in a separate commit.
- Deploy the final version of your code to a hosting platform such as Heroku.

User Story

User Point of View

“As a user I want to be able to create tickets for bugs/features, upvote and view existing tickets, see how many hours are spent working on bugs/features per day/week/month and see what the highest voted tickets are”

User Information

Who is using the app?
A person using the app.

What is it they want?
To create, upvote and view tickets in order to request changes and bug fixes and be sure their money is well spent.

Why do they want it?
So the product they enjoy will improve and they can get the features they want added quicker.

Database Schema

Tables

1. Users - username, number of comments, userid, email , password, first name, last, role (3 by default)

2. Tickets - username, ticket id, description, title, date created, value(if feature), number of comments, type (1 or 2), upvotes, status, userid, updated at
3. Ticket type is 1 or 2 monetary value for 2 is null and 1 cannot be null say 1 doller at least, title (bug or feature)
4. Comments: ticket id, comment, date created, userid, updated at, id
5. Roles: id (1- 3) , title (is the role e.g. admin) like a reference point so it tells the database what the numbers mean

On backend, if something is updated make sure its the same author, superadmin can also edit it, explain roles on readme. Admin cant update ticket or comment, can onlu set statuts, superadmin can do anything, a page for all users only superadmin can use/see it it otherwise 404, superadmin can see list of all users, where he can be promoted, or delete a user, only a user can edit their own details

Main Project Functionality Outline

Create Ticket Functionality

Page
Create Ticket Page

Description
Create the backend code and frontend form to allow users to add new tickets to the site.

Idea Sketchpad
Ticket has data: id, number of upvotes, creators name, description, title, comments (an array in an array of key/values, key = username, value = comment e.g. { [id1: [Kim: ‘Hello World”], id2 : [Cian: ‘I love this!”]] }), current status of the ticket (to do, doing, done).

User Registration Functionality

Page
Navbar, Registration Page

Description
Create the backend code front end interface to enable users to create a username and have their created tickets and comments associated with that username and store the username in the database.

Idea Sketchpad

Unique usernames, user has data: username, created tickets [id :[ticket data]], comments: [commentid: comment], number of comments, number of tickets, money paid?,

Login Functionality

Page
Navbar, Login Page

Description
Enable the user to log into their existing account.

Detailed Ticket Page Functionality

Page
Detailed Ticket Page

Description
Create a detailed view for each ticket, that would just show all attributes for that ticket,

Idea Sketchpad
Each recipe generates its own page via ‘id’ e.g. when the user clicks a ‘view button’ it will append the ‘id’’ to the URL bar and fill in the data in a base.html template unique to the recipe, use the same page outline as the Add ticket Page just with un-editable fields.

Edit/Delete Ticket Page Functionality

Page
Detailed Ticket Page or Edit Ticket Page

Description
Allow for editing and deleting of the ticket

Idea Sketchpad
Same as add more or less except with fields autofilled

Ticket Index Functionality

Page
Graph Page

Description

The user should be able to view the amount of time per day/week/month dedicated to developing new features (think simple graph where 8 hours is the time span and the points are tasks) also should list all tickets a la github

Idea Sketchpad

Research Graph, graph x axis is date, week or month,