CSC322

Professor Wei

12/11/24

Jezlea Ortega, Aprajita Srivastava, Nadia, Suhana Lama, and Tahsina Khan

Final system report

1. Introduction (1 or 2 paragraphs)

Packages and platforms used in this system so anyone can install and run your system from scratch

To run the project, we need to run the backend from the repository

Platforms Used: Node.Js for backend, React for frontend and MongoDB for the database

Steps to run backend:

1. Open the terminal in VS Code and run the following command to initialize a Git repository:
```
git init
```

2. Connect your local repository to the remote repository by running:
```
git remote add origin https://github.com/aprajita0/E-bidding_322Project.git
```

3.Retrieve the project files from the remote repository by running:
```
git pull origin main
```

4: Change into the server directory by executing:
```
cd server
```

5: In the server folder, add the env file

ENV FILE CONTENT:
```
MONGO_URI =
"mongodb+srv://admin:1234@e-biddingdb.kjq3q.mongodb.net/?retryWrites
=true&w=majority&appName=e-biddingDB"
JWT_SECRET =
"c369d8afdec2f56d08ca002fac7915ed00f0320bed587e5581efdd0b15a35d908fe
750aabcda578bfb75fd81130046c89e908285a6d90f846b382ff6e3faf573"
```

6: Run the commands below in the terminal:

```
npm i bcrypt cors dotenv express jsonwebtoken mongoose nodemon
```

7. Once all Node packages are installed run:

```
npm start dev
```

Have the backend running and then in a new terminal run commands:

```
npm install
npm install @mui/material @emotion/react @emotion/styled
npm start
```

2. 1-1 correspondence to the required system features listed in the system spec: finished, partially done, or unable to do it

1. There are three types of users: visitors (V), users (U) and super users (S) - COMPLETE
2. A V can browse the listing of items provided by U's and provide comments. V can apply to be U, which needs to be approved by S. To avoid automatic robots, a question should be answered by V in the application process to decide whether V is a human being: asking a random arithmetic question that robots cannot handle - COMPLETE
3. A U can deposit/withdraw money from one's own account, can list any items/services s/he owns and the asking price for sale or rent can list any items/services s/he needs and the price range s/he can accept; can bid on an item s/he is interested in and the deadline of the bid, the bidder U should have adequate money in the system to be qualified; the owner U decides to accept which U to sell/rent the item/service - COMPLETE
4. Once the owner and buyer/renter Us agree, the item is removed from the listing, and the corresponding price is transferred from the buyer/renter to the owner. - COMPLETE
5. After the transaction is done, the buyer/renter can rate the owner: from 1 - worst to 5 - best. The owner can rate the renter after the transaction. U's do not know who rates them so they cannot retaliate or praise back. No one else can rate. A U can directly complain to S about the Us/he bought/rented items. - COMPLETE
6. A U whose rating less than 2 evaluated by at least three U's is suspended, which can only be active again by paying a $50 fine or re-activated by S. A U whose average ratings (>=3 times) are less than 2, viewed as too mean, or greater than 4, viewed as too generous, is suspended - COMPLETE
7. A U suspended three times is forced out of the system; A U can choose to quit from the system by applying to S. - COMPLETE
8. A U is a VIP if 1) with more than $5,000 in the account; 2) conducted more than 5 transactions; 3) without any complaints; A VIP receives a 10% discount for every transaction; violations of the previous two conditions will move the VIP back to an ordinary U. VIP has all powers U enjoys. VIP who is supposed to be suspended will not be suspended but will be put back to an ordinary U - COMPLETE

9. Provide a personalized GUI so that when a U logs in, information related to him/her based on her/his previous records is shown - INCOMPLETE
10. One creative feature worthy of 10% of the system points is required, e.g., a live bidding session (a game - like feature with time - sensitive constraints) organized and participated by VIPs for items owned by VIPs, a feature of special creativity will receive a special bonus (up to 10%). - COMPLETE

3. contributions of each team member and suggested (if ever) bonus or penalty to certain team members

Jezlea:

- Home page design and navigation bar setup
- Redirection based on role for every button and page clicked
- Login page and ability for user to login to an account
- Registration page and ability for users to register for an account
- U, S, and VIP profile pages with access to their respective granted permissions (e.g. S have an area to resolve complaints).
- Browse listings page for all roles with the ability for visitors, VIP, S, U to add comments and get comments
- S page which allows users to view a list of visitors applying to be a user and approve
- S ability to see suspended accounts and reactive them,
- Ability for all roles to access money total and deposit or withdraw money (checking on withdraw there is enough money in the account).
- U/S/VIPs ability to list any items/services s/he owns and the asking price for sale or rent can list any items/services s/he needs and the price range s/he can accept; can bid on an item s/he is interested in and the deadline of the bid. Included section and fixed connection for the owner U to decide to accept which U to sell/rent the item/service.
- U ability and page to file a complaint form to super user about an item purchased or rented.
- U and VIP’s ability to directly complain to S about the Us/he bought/rented items, and ability for S to resolve these complaints.
- Creative feature - browse raffle listings page, ability for VIP’s to enter other VIP's raffles if they have a transaction with the VIP, and ability for VIP to create a raffle.
- S design/functionality to see and approve quit request applications
- U design/functionality to see and optionally submit a quit request
- Fixed check-vip to ensure when vip conditions are not met on login or after the account balance is < 5000, the users role is changed and they have to log back in.
- Fixed ban user to disallow users to log in (i.e. they are forced out the system) when banned.
- Performed complete/extensive testing to ensure system requirements were implemented and functioning as per the spec requirements.

Aprajita:

- Set up MongoDB for storing all data.

- Added roles for visitors, superusers, and users.
- Created all 13 models for the APIs.
- Worked on and completed authentication with JSON Web Token and designed and tested the login and register APIs.
- Ability to suspend VIPs by returning to User role (frontend)
- Ability for VIPs to receive a 10% discount on all transactions (frontend)
- Designed and tested API endpoints for /withdraw-balance, /deposit-balance, /get-balance, /get-listing, and /get-listing/:id.
- Designed and tested API endpoints for /add-notification, /user-get-listing, /add-raffle, /get-listings, and /get-raffle (fetches all raffles except those belonging to the currently logged-in user).
- Fixed and tested /accept-bid, /suspend-reguser, /pay-fine, and /check-vip.
- Designed and tested API endpoints for /rate-transactions, /get-listing-rating, and /apply-reg-user.
- Designed and tested API endpoints for /unban-user, /get-visitor-applications, and /get-transactions (retrieves all successful transactions for a user to enable filing a complaint).
- Fixed and added API endpoints for /check-account-status, /unban-user, and /deny-bid.
- Designed and tested APIs for /enter-raffle, /win-raffle, and /get-raffle/:id.
- Fixed and added /add-notification, /fix-ratings, /get-raffle-entries, and read-status.

Nadia:

- Designed and tested API endpoint for `/buy-listing`.
    - Purpose: Handles the purchase of listings.
    - Note: This endpoint will be removed after merging with other APIs.
- Designed and tested API endpoint for `/add-comment`.
    - Purpose: Allows users to add comments to listings or items.
- Designed and tested API endpoint for `/add-complaint`.
    - Purpose: Enables users to submit complaints about listings or items.
- Worked on and fixed API endpoint for `/accept-bids`.
    - Purpose: Manages bid acceptance for listings.
        - For buying-type listings:
            - The owner of the listing loses money in their account.
            - The person placing the bid gains money.
- Designed and tested API endpoint for `/check-vip`.
    - Purpose: Verifies if a user has VIP status.
- Designed and tested API endpoint for `/get-complaint`.
    - Purpose: Retrieves:
        - Accounts of users filing complaints.
        - Details of complaints.

        - Status: Fixed to ensure accurate retrieval of complaint data.
- Worked on and fixed API endpoint for `/unsuspend-account`.
    - Purpose: Unsuspends users previously suspended, without imposing fines.
    - Update: After unsuspension:
        - All user ratings are removed to prevent immediate re-suspension upon logging back in.
- Designed and tested API endpoint for `/get-suspended-account`.
    - Purpose: Retrieves information about suspended accounts, including:
        - The number of suspensions associated with each account.
- Fixed and updated logic for `/get-complaint`.
    - Purpose: Ensured accurate retrieval of user account and complaint details.
- Fixed and tested logic for `/unsuspend-account`.
    - Purpose: Updated to remove all ratings for a user after unsuspension, ensuring compliance with system rules.

Tahsina:

FRONTEND (all done using apis created by aprajita, suhana & nadia)

- Visitor Profile Page design and ability to apply to be a user with a CAPTCHA question
- U, S, and VIP ability to bid on an item & place a deadline on bid while checking for sufficient account balance
- U, S and VIP ability to accept and deny bids, and once a bid is accepted, the listing disappears from Browsing Listings page
- Anonymous rating system- Seller ability to rate Buyer via a modal once bid is accepted, and Buyer ability to rate Seller by viewing pending ratings on U, S & VIP profiles
- Ability to identify if at least three users gave them a less than 2 rating or their Average Rating is < 2 (too mean) or > 4 (too generous) and perform a suspension on Login
- Suspension Page where suspended users can choose to pay a $50 fine or wait for a Super User to reactivate them
- U and VIP ability to pay $50 fine, which deducts money from balance and then reactivates their account
- Ability to detect whether users have been suspended three times and perform a ban which notifies them that they have been banned and then forces them out of the system
- U ability to apply to quit/deactivate their account, S ability to approve those applications
- Check if User meets the conditions for becoming a VIP (>5000 balance, >=5 transactions, and 0 complaints) and promote to VIP on Login (reroutes to VIP page)
- Ensured that VIPS have all abilities Users do
- Creative feature- Raffle System for VIPS - made a progress bar that shows that VIPs have entered in at least one raffle, ability to choose a Raffle winner after creating a raffle, and ability to see the history of the raffles the VIP has created as well as the raffles they've entered

- Made sure that VIPs get a notification if they win a bid, and press a button that marks notification as “read” so they don’t continue receiving it
- Testing to make sure specs work

BACKEND:
- Updated /suspend-reguser api (previously only checked if ratings were less than 2) and /rate-transactions
- Helped fix deny-bid (called wrong “owner_id” field which didn’t exist)
- Tested /check-vip and /get-user-ratings
- Designed and tested /quit-requests api (retrieved all quit requests)

Suhana:
- Designed and tested APIs for /get-listing, /bid-listing, and /add-listing.
- Designed and tested APIs for /suspend-reguser, /approve-reguser and fixed /bid-listing.
- Designed and tested APIs for /get-bids, /read-notify, and /pay-fine.
- Designed and tested APIs for /get-notif, /deny-user and fixed /pay-fine.
- Designed and tested APIs for /user-quit-request and /approve-quit.
- Designed and tested APIs for /get-user-ratings.

4. git address of your repo: https://github.com/aprajita0/E-bidding_322Project

5. (optional) remarks or suggestions
No remarks or suggestions from the team.