Table of Contents

Login

Abstract Code

- When User clicks **Register**:
  - Navigate to **Register** page
- When User Clicks **Login**:
  - If browser/api validation succeeds for *username* and *password*, then

```
SELECT `User`.`username`, `first_name`, `last_name`, `position`
FROM `cs6400_spr18_team100`.`User`
LEFT OUTER JOIN `cs6400_spr18_team100`.`AdminUser`
ON `User`.`username` = `AdminUser`.`username`
WHERE `User`.`username` = '$username'
AND `User`.`password` = sha1('$username + $password');
```

    - If result set is empty
      - Go back to **Login** form, with failed login message.
    - Else
      - Create session for user
      - Navigate to **Main Menu** page
- Else *username* and *password* are incorrect, display **Login** with error message.

Register

Abstract Code

- User enters *FirstName, LastName*, *username, password, and confirm password* input fields
- When user clicks **Register:**
  - If browser input validation succeeds:

```
INSERT INTO `cs6400_spr18_team100`.`User` (`username`, `password`,
`first_name`, `last_name`)
VALUES ($username, SHA1('$username + $password'), '$first_name', '$last_name');
```

    - If username already exists error
      - Go back to **Registration** form, with error message.
    - If Else some other error
      - Go back to **Registration** form, with error message.
    - Else
      - Give “Successful Registration” message and redirect user to “**Login**” view
  - Else
    - Display error message on **Registration** form
      - “Passwords don’t match”
      - “Username is required”

        - "Fisrt Name is required"
        - "Last Name is required"
        - Etc.
- When **Cancel** is clicked:
    - Return to **Login** page

List My Item

Abstract Code

- When **List my Item** is clicked:
    - If input passes validation

```
INSERT INTO `cs6400_spr18_team100`.`Item` (`username`, `name`,
`description`, `returnable`, `starting_bid`, `get_it_now_price`,
`minimum_sale_price`, `category`, `condition`, `length`)
VALUES ('$username', $name', $description', $returnable, $start, $now,
$min, $category', '$condition', $length);
```

        - If db error
            - Display error message on page
        - Else
            - Navigate back to **Main Menu** page
    - Else
        - Display error message

Item Search

Abstract Code

- When **Search** is clicked
    - If all fields are blank
        - **Search** is disabled
    - If input passes browser/api validation
        - Build query based on provided input (sql example contains all input)

```
SELECT `ID`, `Item Name`, `Current Bid`, `High Bidder`, `Get It Now Price`, `Auction Ends`
FROM `HighBidder`
LEFT JOIN `Item`
ON `Item`.`item_id` = `HighBidder`.`ID`
WHERE `name` like '%$keyword%' OR `description` like '%$keyword%'
AND `Item`.`category` = $category'
AND $min >= `Current Bid` or $min >= `Item`.`starting_bid`
AND `Item`.`condition` <= 2;
```

        - If error
            - Display error on **Item Search** page
        - else
            - Navigate to **Search Results** page and display results
    - Else
        - Display error message on **Item Search** view
- When cancel is clicked
    - Return to **Main Menu** page

Edit Item Description

Abstract Code

Abstract Code:

- When user clicks **Edit Description**
    - Display **Edit Description** popup window.
    - When user clicks **Save Description**
        - If input passes browser/api validation

```
UPDATE `cs6400_spr18_team100`.`Item` SET `description`=$description'
WHERE `item_id`='$id';
```

            - If Error
                - Display error message
            - Else
                - Close popup and refresh **Item Info** page
        - Else display error message
    - When user clicks **Cancel**
        - Close popup window

Get It Now

Abstract Code

- **Upon:**
    - Clicking **Get It Now**
        - Check that Item is still for sale

```
SELECT item_id FROM (SELECT `Item`.`item_id` AS `item_id`,`Item`.`name` AS `Item
Name`,`Item`.`username` AS `Seller`,`Bid`.`username` AS `Purchaser`,max(`Bid`.`price`) AS
`Sold Price` FROM (`Bid` JOIN `Item` ON((`Bid`.`item_id` = `Item`.`item_id`))) WHERE
((now() > (`Item`.`created` + interval `Item`.`length` day)) OR (`Bid`.`price` =
`Item`.`get_it_now_price`)) GROUP BY `Item`.`item_id`) `AuctionEnded`
WHERE `item_id` = $id;
```

- If Item is no longer for sale (not returned in the result set)
  - return to **Item** page with error message “Auction has ended”
- Else
  - Insert new bid for get it now price

```
INSERT INTO
 `cs6400_spr18_team100`.`Bid` (`item_id`, `username`, `price`) VALUES
```

- Update **Item** page with get it now bid

Bid

Abstract Code

- User enters bid value
- **Upon:**
  - Clicking “**Bid On This Item**” button
    - If input is valid
      - Check that item is still for sale

```
SELECT item_id FROM (SELECT `Item`.`item_id` AS `item_id`,`Item`.`name` AS `Item
Name`,`Item`.`username` AS `Seller`,`Bid`.`username` AS `Purchaser`,max(`Bid`.`price`) AS
`Sold Price` FROM (`Bid` JOIN `Item` on((`Bid`.`item_id` = `Item`.`item_id`))) WHERE
((now() > (`Item`.`created` + interval `Item`.`length` day)) OR (`Bid`.`price` =
`Item`.`get_it_now_price`)) GROUP BY `Item`.`item_id`) `AuctionEnded`
WHERE `item_id` = $id;
```

      - If item is still for sale (result set is not empty)

```
INSERT INTO `cs6400_spr18_team100`.`Bid` (`item_id`, `username`, `price`) VALUES
($id,'$username',$bid);
```

      - Else
        - Return failure message: “Auction has ended”
        - Update “**GTBay Item for Sale**” view with new information
    - Else
      - Return failure message: “**Invalid Bid**”

View Item Info

Abstract Code

- **Upon:**
  - Clicking **ItemName**

```
SELECT `item_id`, `name`, `description`, `category`, `condition`,
`get_it_now_price`, `returnable`, (`Item`.`created` + interval
`Item`.`length` day) AS `Auction Ends`
FROM `Item` WHERE item_id = $id;
```

- Query latest 4 bids

```
SELECT `username`, `price`, `bid_time` FROM `Bid` WHERE `item_id` = $id LIMIT 4;
```

View Ratings

Abstract Code

- **Upon:**
  - Clicking "**View Ratings**"
    - Query database for derived attribute "**Average Rating**".

```
SELECT AVG(`number_of_stars`) FROM Rating AS `Average Rating` WHERE `item_id` = $id;
```

- Query database for all ratings of the "**Item**" and current "**User**"

```
SELECT `username`, `rate_time`, `number_of_stars` FROM `Rating` WHERE `item_id` = $id;
```

```
SELECT `username`, `rate_time`, `number_of_stars` FROM `Rating` WHERE `username` = $username;
```

- Navigate to "**Item Ratings**" view and display results

Rate This Item

Abstract Code

- **Upon:**
  - Clicking "**Rate This Item**"
    - If input passes browser/api validation

```
INSERT INTO `Rating` (`username`, `comment` `number_of_stars`) values ($username, $comment, $number_of_stars);
```

      - Navigate back to **Item** page
    - Else
      - Return error message to user

Delete Rating

Abstract Code

- Upon:
  - Clicking "**Delete My Rating**"

```
DELETE FROM `Rating` WHERE `username` = $username and `item_id` = $id;
```

Auction Results

Abstract Code

- Query Auction Results

```
SELECT `Item`.`item_id`, `Item`.`name` AS `Item Name`, `Item`.`username` AS `Seller`,
`Bid`.`username` AS `Purchaser`, max(`Bid`.`price`) AS `Sold Price`,
least(coalesce(max(`Bid`.`bid_time`), (`Item`.`created` + INTERVAL `Item`.`length` DAY)),
coalesce((`Item`.`created` + INTERVAL `Item`.`length` DAY), max(`Bid`.`bid_time`)))AS
`Auction Ended` FROM `cs6400_spr18_team100`.`Item`
LEFT JOIN `cs6400_spr18_team100`.`Bid`
ON `Bid`.`item_id` = `Item`.`item_id`
WHERE NOW() > (`Item`.`created` + INTERVAL `Item`.`length` DAY) OR `Bid`.`price` =
`Item`.`get_it_now_price`
GROUP BY `Item`.`item_id`;
```

- Display Results
- When User clicks **Done**
    - Return to "**Main Menu**" view

Category Report

Abstract Code

- Display **Category, Total Items, Min Price, Max Price, Average Price**

```
SELECT name as `Category`, count(*) as `Total Items`, max(`price`) as `Max Price`,
min(`price`) as `Min Price` from (SELECT `Category`.`name`, `Bid`.`price` from
`cs6400_spr18_team100`.`Item`
join `cs6400_spr18_team100`.`Category`
on `Item`.`category` = `Category`.`name`
join `cs6400_spr18_team100`.`Bid`
on `Item`.`item_id` = `Bid`.`item_id`) `CategorySales`
group by name;
```

- When Done is clicked
    - Return back to **Main Menu**

User Report

Abstract Code

- Display **Username, Listed, Sold, Purchased, Rated**

```
select `UserRatings`.`username`, `Listed`, `Sold`, `Purchased`, `Ratings` as `Rated`
from (select `Rating`.`username` AS `username`,count(`Rating`.`item_id`) AS `Ratings` from `Rating` group
by `Rating`.`username`) `UserRatings`
left outer join (select `Purchaser` as `username`,  count(`Sold Price`) as `Purchased` from (select
`Item`.`item_id` AS `item_id`,`Item`.`name` AS `Item Name`,`Item`.`username` AS `Seller`,`Bid`.`username`
AS `Purchaser`,max(`Bid`.`price`) AS `Sold Price`,max(`Bid`.`bid_time`) AS `Auction Ended` from (`Bid` join
`Item` on((`Bid`.`item_id` = `Item`.`item_id`))) where ((now() > (`Item`.`created` + interval `Item`.`length`
day)) or ((`Bid`.`price` = `Item`.`get_it_now_price`) and (`Bid`.`price` >= `Item`.`minimum_sale_price`)))
group by `Item`.`item_id`) `ItemsSold`
group by `Purchaser`) `Purchases`
on `UserRatings`.`username` = `Purchases`.`username`
left outer join (select `Seller` as `username`,  count(`Sold Price`) as `Sold` from (select `Item`.`item_id` AS
`item_id`,`Item`.`name` AS `Item Name`,`Item`.`username` AS `Seller`,`Bid`.`username` AS
`Purchaser`,max(`Bid`.`price`) AS `Sold Price`,max(`Bid`.`bid_time`) AS `Auction Ended` from (`Bid` join `Item`
on((`Bid`.`item_id` = `Item`.`item_id`))) where ((now() > (`Item`.`created` + interval `Item`.`length` day)) or
((`Bid`.`price` = `Item`.`get_it_now_price`) and (`Bid`.`price` >= `Item`.`minimum_sale_price`))) group by
`Item`.`item_id`) `ItemsSold`
group by `Seller`) `Sellings`
on `UserRatings`.`username` = `Sellings`.`username`
left outer join (select `Seller` as `username`, count(`item_id`) as `Listed` from (select `User`.`username` AS
`Seller`,`Item`.`item_id` AS `item_id`,`Item`.`name` AS `name` from (`User` join `Item` on((`User`.`username`
= `Item`.`username`)))) `UserListings`
group by `Seller`) `Listings`
on `UserRatings`.`username` = `Listings`.`username`;
```

- When Done clicked
    - Return to **Main Menu**