PROFESSIONAL

E-RIDE STAGE 8

INSTRUCTIONS:

Goal of the Project:

In class 82, you modified the code to check for the student's eligibility before issuing a book using Firebase queries. In this project, you will apply the same concept to check the user's eligibility before assigning the bike to the user and also while returning the bike.

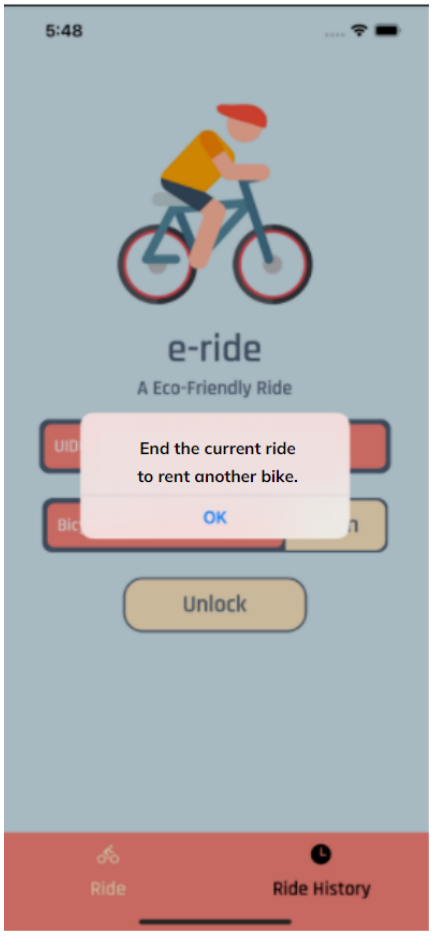
* This is a continuation of Projects 75-81. Please make sure you have completed and submitted those projects before attempting this one.

Story:

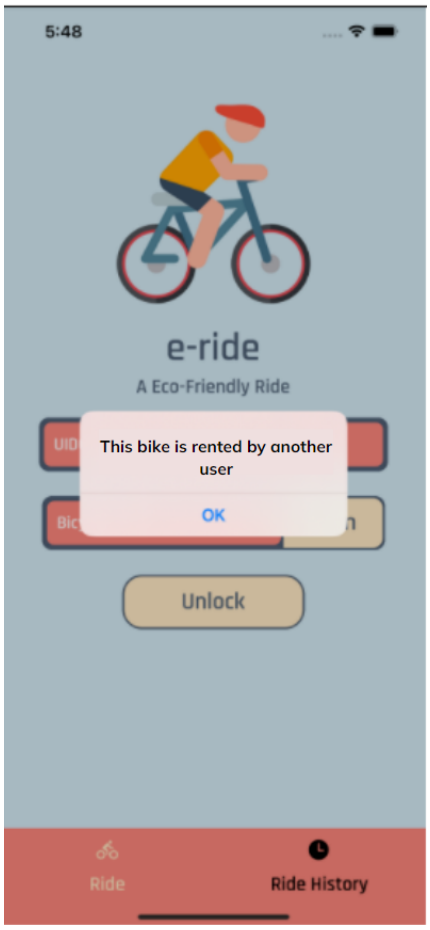
Your friend Vihaan has come across a situation where the same user is unlocking multiple bikes at the same time. He also noticed a case where the user who returned the bike was different from the one who was assigned that bike.

You need to modify the app code to refrain users from doing the same. Vihaan thus wants to update the app to run certain checks on users as well as bikes.

Here is how the message is displayed to the user on checking for the user's availability before renting the bike and while returning it. In this project, can you help Vihaan to check for the user's availability before renting a bike and also while returning it?



Application displaying the message on checking the user's eligibility while starting the ride



Application displaying the message on checking the user's eligibility while ending the ride

***This is just for your reference. We expect you to apply your own creativity to the project.**

PROFESSIONAL

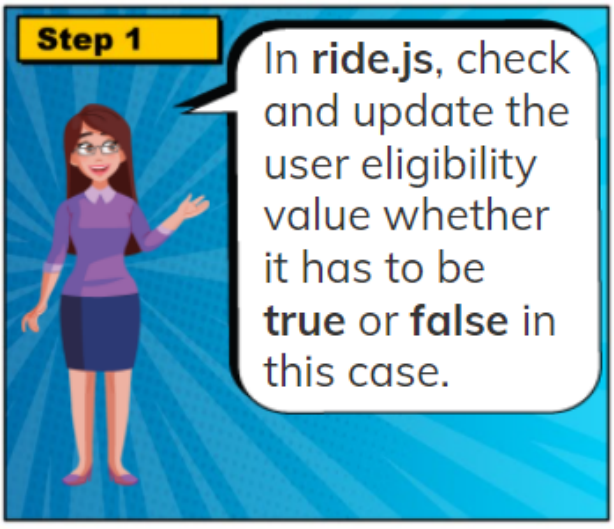
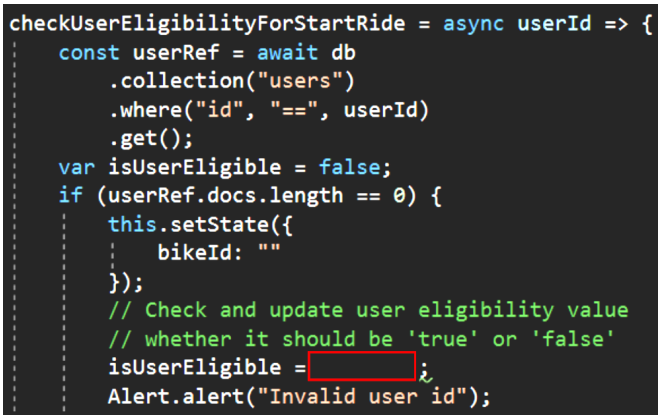
E-RIDE STAGE 8

WhiteHat Jr
Live Online Coding for Kids

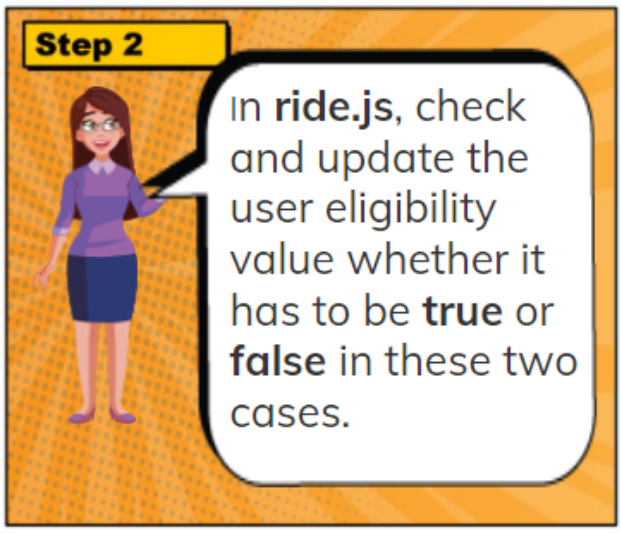
Getting Started:

1. Use the template on **GitHub** by downloading it from this link.
2. **Unzip** the downloaded zip folder.
3. Rename the unzipped folder as **Project 82.**
4. Open Command Prompt:
    a. For Windows OS, type **cmd** in the Run box.
    b. For macOS, type **Terminal** in the search bar.
5. Type **cd Project 82** (you have to give the full path on your computer).
6. Type **npm install**.
7. Open the project folder in **VS Code**.
8. Run the code by typing **expo start** in **Command Prompt/Terminal.**
9. Start editing your code in **Ride.js**.

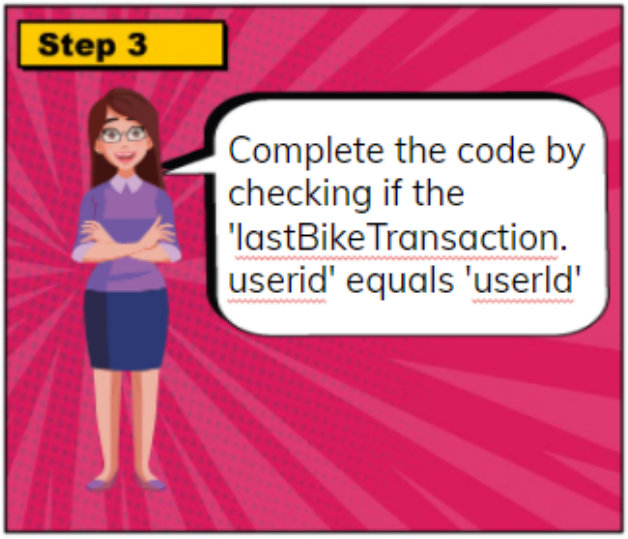
Specific tasks to complete the project:

Things to do	Code Blocks
	

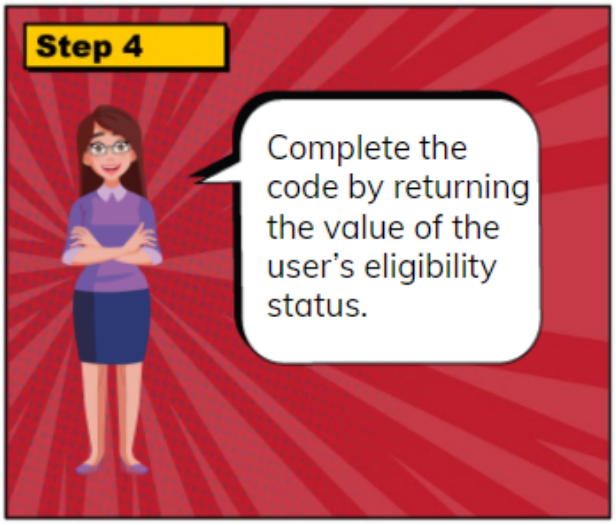

```
checkUserEligibilityForStartRide = async userId => {
    const userRef = await db
        .collection("users")
        .where("id", "==", userId)
        .get();
    var isUserEligible = false;
    if (userRef.docs.length == 0) {
        this.setState({
            bikeId: ""
        });
        // Check and update user eligibility value
        // whether it should be 'true' or 'false'
        isUserEligible = ;
        Alert.alert("Invalid user id");
```



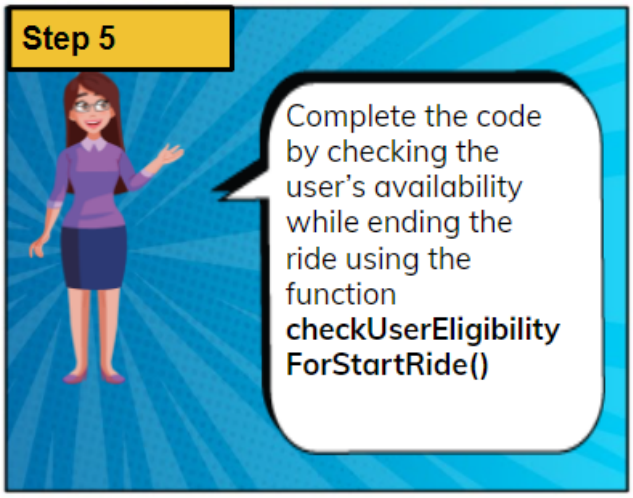

```
    Alert.alert("Invalid user id");
} else {
    userRef.docs.map(doc => {
        if (!doc.data().bike_assigned) {
            // Check and update user eligibility value
            // whether it should be 'true' or 'false'
            isUserEligible = [          ];
        } else {
            // Check and update user eligibility value
            // whether it should be 'true' or 'false'
            isUserEligible = [          ];
            Alert.alert("End the current ride to rent another bike.");
            this.setState({
                bikeId: ""
            });
```




```
checkUserEligibilityForEndRide = async (bikeId, userId) => {
    const transactionRef = await db
        .collection("transactions")
        .where("bike_id", "==", bikeId)
        .limit(1)
        .get();
    var isUserEligible = "";
    transactionRef.docs.map(doc => {
        var lastBikeTransaction = doc.data();
        // Check if 'lastBikeTransaction.userid' equals 'userId'
        if ([                    ]) {
            isUserEligible = true;
```



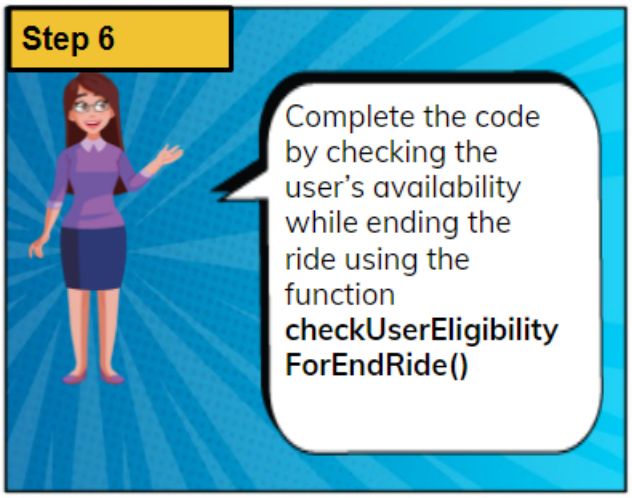

```
        } else {
            isUserEligible = false;
            Alert.alert("This bike is rented by another user");
            this.setState({
                bikeId: ""
            });
        }
    });
    // Return the user's eligibility status
    return                ;
};
```



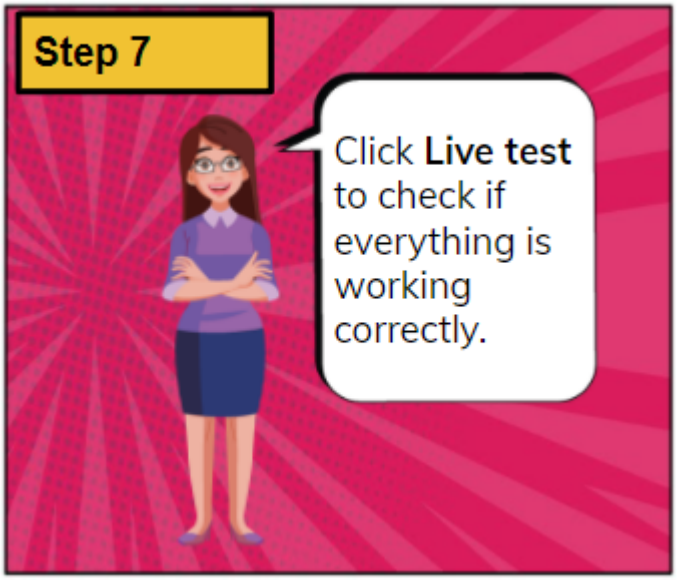

```
handleTransaction = async () => {
    var { bikeId, userId } = this.state;
    await this.getBikeDetails(bikeId);
    await this.getUserDetails(userId);

    var transactionType = await this.checkBikeAvailability(bikeId);

    if (!transactionType) {
        this.setState({ bikeId: "" });
        Alert.alert("Kindly enter/scan valid bike id");
    } else if (transactionType === "under_maintenance") {
        this.setState({
            bikeId: ""
        });
    } else if (transactionType === "rented") {
        // Check user availability for start ride
        var isEligible =                ;

        if (isEligible) {
            var { bikeType, userName } = this.state;
            this.assignBike(bikeId, userId, bikeType, userName);
```

PROFESSIONAL

E-RIDE STAGE 8




```
} else {
    // Check user availability for end ride
    var isEligible = [          ];

    if (isEligible) {
        var { bikeType, userName } = this.state;
        this.returnBike(bikeId, userId, bikeType, userName);
        Alert.alert("We hope you enjoyed your ride");
        this.setState({
            bikeAssigned: false
        });

        // For Android users only
        // ToastAndroid.show(
        //   "We hope you enjoyed your ride",
        //   ToastAndroid.SHORT
        // );
    }
```



Submitting the Project:

1. **Upload** your completed project to your **GitHub** account. Here is a video on how to do this: https://vimeo.com/561338335/aa2b0db66e

2. Enable **GitHub** pages for the repository. After you have performed this step, wait for a few minutes for your project to be live. **See the video given below:** https://vimeo.com/561338446/a7e3084fb4

3. Copy the GitHub link and submit it in the **Student Dashboard-> Projects** panel against the correct class number.

REMEMBER... Try your best, that's more important than being correct.
After submitting your project your teacher will send you feedback on your work.

———— xxx ———— xxx ———— xxx ———— xxx ———— xxx ————