Difficulty level legend:
1 : Very low difficulty
2 : Low difficulty
3 : Medium difficulty
5 : Medium difficulty with a risk of unplanned difficulty
8 : High difficulty

SPRINT #2

Issue #	User Story Title/Task Title	Description/Associated Tasks	Difficulty Level (0,1,2,3,5,8)	Priority	Risk	Assign	Sprint
#16	TASK-1.1: Set Up the database for Vehicles	-create a database for vehicles -populate the database with vehicles	**8** - This would require a fully or almost fully functional database that can be used by the front end team with	High	Low	Jad	2
#17	TASK-1.2: Design and draw all website page mockup	- Gather inspiration from multiple Car Rental websites - Design a mock-up for the Website logo and all pages of the Website - Review mockup with the team and make final	**2** - The team has already found multiple website inspirations during team meeting, thus it will be easy to cre	High	Low	Sara	2
#18	TASK-1.3: Implement search bar on Home page	- Create a component for a search bar and add it to the home page that includes the desired location, from and until dates of the reservation and search icon. - Once the search button is clicked, the user is redirected to the Catalog page where all matching search results are displayed.-	**3** - Some components can be found for this feature	Medium	Low	Irina	2
#19	TASK-1.4: Implement search bar on Home page	- Handle the incoming HTTP requests - Code the search mechanism in backend - Successfully manipulate the information in the database and return the sorted information.	**3** - some backend code or component can be found to execute this feature, however medium strength since	High	Low	Hossam	2
#20	TASK-1.5: Implement Filter Functionality on Catalog P-	- Create Filter Component - Implement user interactions for selecting and clearing filters - When a filter is applied, the search results are updated accordingly on the UI When the Filter option is cleared, displayed search results are reset to original state without any filters	**5** - Some errors in queries might occur, need to make sure to send and receive proper requests to backend	Medium	Medium	Boudour	2
#21	TASK-1.6: Implement Filter Functionality on Catalog P-	- Check the filtering options with frontend Code the logic for the database filtering system.	**3** - addition of new filters to already set database, might be time consuming if large database	Medium	Medium	Enrique	2
#22	Task-1.7: Configure Carousels for browsing options on -	- Create a "Carousel" component Integrate with API Backend to display vehicles matching the Carousel category	**2** - Need to ensure proper filtering of cars, implementing a responsive and scalable carousel might be a chi	Low	Low	Sara	2
#24	TASK-2.1: Implement Pop-Up for login/ sign-up	- Create a new component for the Pop-up window - Implement Role-based authentication: if the user is CSR or System Admin, only allow log-in with their existing credentials else, regular customer can either login with his credentials or create a new account - Handle log-in Form: prompt existing user for "username" and "password", include "Forget Password" option - Handle Sign-up Form: prompt new customer for "Full name", "email address", "username", "password", "confirm password" - Validate user inputs and display error messages gracefully	**8** - Login is a challenging task and it has to be adaptable for different accounts, incorporate a security chec	High	Medium	Irina	2
#25	TASK-2.2: Set-up the database for users	- Create a database for users only. - Each user must have a variable for: name, birthdate, profile picture, rental history	**5** - possible issue might arise during storage of profile pictures, and need protection of sensitive information	High	Medium	Hossam	2
#26	TASK-2.3: Set-up admin user account	- Write backend code to create an admin account with the specified fields: "Username", "Password", "Role" and store it in the Users' database - Design and implement backend API endpoints for CRUD operations on users, vehicles, and reservations - Implement authorization logic to check the user's role and permissions when accessing different API endpoints	**2** - Back end functionalities for account will already be set, thus it is just a matter of adjusting authorizations	Medium	Low	Enrique	2
#27	TASK-2.4: Create user-specific views based on credet-	- Determine specific permissions/ access rights associated with each user: Customer, CSR, SystemAdmin - Create separate interfaces and Navigation bars for each user role, tailored to their specific needs and permissions: - Customers: can access "My Reservations", "My Cart", and "History", modify and cancel upcoming reservations CSR: can access and confirm "Pending Reservations", access, modify and cancel upcoming reservations. System Admin: allowed to access all users, vehicles and reservations' details, add a new record, edit, and delete. Implement form validation for input fields to ensure that the data entered by the admin is valid before submitting it to the backend	**5** - mismatch between credentials might arise	Medium	Medium	Boudour	2
#28	TASK-2.5: Implement middleware for backend logic handling		**5** - back end is already set up so medium difficulty, however logic handling can be quite meticulous to ensu	High	Medium	Jad	2
#29	TASK-3.1: Prompt User to login	- Implement functionality to detect if the user is logged in when they click "book now." If logged in, proceed to the reservation form. If not logged in, display the login pop-up - After successful login, redirect the user to the reservation form where they can complete the booking process - Design and implement a "person icon" for user account management. When the user clicks on the person icon in the top bar, open the login pop-up - Ensure that the user's session is maintained across pages after login to avoid frequent logins	**3** - the creation of the pop- up is quite easy to do, the challenge might arise in frontend credentials of the us	Medium	Low	Irina	2
#30	TASK-3.2: Implement a reservation form interface	- Create a Form component with a summary of Vehicle details, entered location, dates, checkboxes for adding extra equipment, and text areas for special notes. - Implement client-side form validation to ensure that required fields are filled out correctly before submitting the reservation - Implement functionality to submit the reservation form data to the backend API endpoint for processing	**3** - form handling like such have been done before in previous projects, however filtering with backend migh	Medium	Medium	Boudour	2
#31	TASK-3.3: Set-Up the database for Reservations	Create the database for reservations with corresponding criteria (status, vehicle id, customer id, starting date, days rented, total price, rating, reservation id)	**3** - some previous databases will already be set, so it will be easy to implement, medium difficult because o	High	Low	Enrique	2
#32	TASK-3.4: Implement middleware for user permissions	on reservations	**8** - user permissions are a very meticulous detail to configure and debug , the use of middleware will be dor	Medium	High	Hossam	2
#33	TASK-3.5: Design and Code the reservation status on -	- Design Reservation Status Component: show the status in a visually clear format, such as a badge or a labelled text - Display the reservation status as "Pending" immediately after the user submits a reservation request Update the reservation status to "Confirmed" when the CSR approves the request. This should trigger a notification to the user's email address	**3** - the reservation tag labeling is quite easy in itself, a posible challenge in the automated email communic	Medium	Medium	Irina	2
#34	TASK-4.1: Access existing reservations on customer a-	- Create "MyReservations" component - Implement logic to fetch existing reservation data from the backend API - Populate the reservation display component with its data: ID, date, location, and status. Handle Empty state: Display a message when user has no reservations	**5** - the data fetching and their display are quite easy to put in practice assume the database was built corre	Medium	Low	Sara	2
#35	TASK-4.2: CSR access to pending customer reservati-	- Create a "Requested Reservations" component for CSR interface - Populate the reservation log interface with the data fetched from the backend - Implement a 'Details' button to allow CSR to view detailed information about each pending reservation. - Provide buttons for CSR to approve or reject each pending reservation - Update reservation status accordingly - When a reservation is approved, move it to "Upcoming reservations" list When a reservation is rejected, prompt the CSR to enter a reason for rejection then move it "Rejected Reservations" list	**5** - same as issue #34 above	High	Medium	Sara	2
#36	TASK-4.3: Set up an interactive table for modifying cur -	- Create "Upcoming Reservations" component - Design a table that displays confirmed reservations - Include editable fields for location, dates and status - Implement "Save" button for CSR to save changes and "Cancel" button to discard changes. Implement client-side validation for the editable fields to ensure that the data entered by the CSR is valid before saving	**5**	Medium	Medium	Boudour	2
#37	TASK-4.4: Implement the middleware for the CSR tabl-	Code the permission for a CSR to delete an existing reservation Code the permission for a CSR to edit an existing reservation	**3** - database is already set and it is a question of playing in the parameters of each user's access. Difficulty	High	Medium	Hossam	2
#38	TASK-4.5: Code the backend for confirming the pendi	Code the change of status from pending to accept/deny from CSR account.	**3** - same justification as issue #37	High	Medium	Jad	2
#41	TASK-5.1: Configure Payment Portal		**8** - payment portals will require many interactions with the database, storage and protection of sensitive inf	High	Low	Enrique	2
#42	TASK-5.2: Handle payment confirmation and unsuccee-	- Create "MyCart" component - Design a checkout process that allows users to add the confirmed reservation to their cart - Provide "Proceed to Payment" button for customer and display the payment process - Prompt customer for Card number, card holder, exp date and cvv - If the payment is successful, update the reservation status to "Paid" and send an email confirmation If the payment is unsuccessful, provide an error message to the user and allow them to try again or choose a different payment method	**8** - same justification as in issue # 41 above. Plus additional difficulty in the logic handeling of unsuccessful	Low	High	Boudour	2
#43	TASK-6.1: Implement Review Prompt	- Create "Review" component - Design a form for users to leave a rating for each car they reserved. Include fields for rating (e.g., 1-5 stars) and comments - Update the reservation status to "Reviewed" Show the ratings and comments left by the user in their reservation history	**3** - multiple review components can be found to be added in our website, however potential difficulty in data	Medium	Low	Irina	2
#44	TASK-6.2: Implement Review Visibility on Products	- Create a "Reviews" section on the product page - Display the average rating and the number of reviews for each product Display individual reviews for each product. Show the rating, comments	**3** - same justification as the issue # 43 above	Medium	Low	Sara	2
#45	TASK-6.3: Develop Backend Logic for Average Rating-	- Code a simple mechanism that multiplies the number of ratings with the average rating of a vehicle, add the new rating to it and divide it by the number of old ratings + 1 g Handle http requests regarding the ratings	**2** - this task is a simple arithmetical equation that has been done previous tutorials and assignments	Medium	Low	Enrique	2