Database Management Systems Project Evaluation Form					
Team, Student ID, Name:		itudb2319, OnlyF1s 150200097 Mustafa Can Çalışkan	**Signature:**		
DATABASE	Object 1	Name: qualifying	**APPLICATION**	Basics	Some of the basic HTML components that used in the project: <form>, <input>, <button>, <table>, <select>, <script>.
	INSERT UPDATE DELETE SEARCH	All CRUD operations are implemented (Only admin users can perform CRUD operations, while regular users can only view the tables.). Also filtering, pagination, ordering and searching features are available.			
	Object 2	Name: drivers		Sessions	Sessions have been used.
	INSERT UPDATE DELETE SEARCH	Implementation and features are entirely identical to the "qualifying" table.		Advanced	Jinja templates and commands.
	Object 3	Name: quiz			
	INSERT UPDATE DELETE SEARCH	Implementation and features are entirely identical to the "qualifying" table.		CSS-JS	The project's overall appearance (tables, top bar, footer, panels, buttons, text bars, etc.) was styled using CSS. Additionally, numerous JavaScript functions were employed for sending forms to the Flask backend and for the proper functioning of the Blink Test.
	Other Objects	driverStandings: Implementation and features are entirely identical to the "qualifying" table.		AJAX	AJAX does not used.
	Tables	4 tables. drivers: 9 col., no foreign key qualifying: 9 col., 3 foreign key quiz: 5 col., no foreign key driverStandings: 7 col., 2 foreign key			
	Data Types	Int, varchar, date, float			
	Relationships	qualifying-races: Many to One qualifying-drivers: Many to One qualifying-constructors: Many to One driverStandings – races: Many to One driverStandings – drivers: Many to One quiz – answers: One to Many			

	Extras	Stored procedures, views, and triggers. Rankings Tables: Comparison tables that includes all dataset. It includes different type of queries that ranks the data for specific condition. As an example: perWins. This table gives the all drivers according to their win rate and season that they raced. This table ordered by the win rate. The Bulk CUD has been implemented to ease one-table-related operations. This method generates a query based on the uploaded csv file's name and content.		Extras	The project has been Dockerized to make easier work as a group. Also, this Docker environment has been supported with the Makefile to manage the Docker-based commands. ‘Contact Us’ page has been created to send emails from users to the Admin. A simple game 'Blink Test' for user interaction has been created using JavaScript. 'Quiz' section has been established to gauge their level of knowledge.
	Problems	The foundation for RabbitMQ implementation has been established, but due to time limitations, it has not been implemented.		Problems	