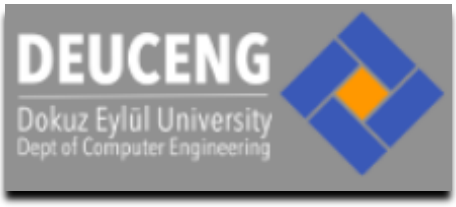


DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 3201 – DATABASE MANAGEMENT SYSTEM

ONLINE VEHICLES FOR SALE

By

Mustafa Efe Demir

Ramazan Hakan Cankul

Ali Şiyar Arslan

December, 2022

İZMİR

Page

CONTENTS

CHAPTER ONE

PROBLEM DESCRIPTION

What is vehicle for sale system?

A vehicle for sale system is a system that allows individuals or businesses to list vehicles for sale, and for potential buyers to browse and search for vehicles that meet their criteria. The system may be an online platform, such as a website or a mobile app, or it may be a physical location, such as a car dealership. Our system is based on website.

In our vehicle for sale sytem, firstly seller choose a vehicle type for advertisiment. After that, the seller typically creates a listing for their vehicle, including details such as the make, model, year, and features of the vehicle, as well as photos and a description.

Buyers can browse and search the listings to find vehicles that meet their criteria, such as a specific make or model, price range, or location. They may also be able to filter the listings by various criteria, such as the age of the vehicle or the type of fuel it uses. Once a buyer finds a vehicle they are interested in, they can contact the seller to inquire about the vehicle or arrange a test drive.

Why we should use the online vehicle system?

There are several reasons why individuals and businesses may choose to use an online vehicle for sale system:

Convenience: Online systems allow buyers and sellers to browse and list vehicles from anywhere, at any time, without the need to physically visit a dealership or other location.

Wide reach: Online systems often have a wider reach than traditional methods, allowing buyers and sellers to connect with each other from different locations.

Increased exposure: Listing a vehicle on an online platform can increase its exposure to a larger audience, increasing the chances of finding a buyer.

Easy communication: Online systems typically provide tools for buyers and sellers to easily communicate with each other, such as through messaging or email.

Efficient: Online systems can streamline the process of buying or selling a vehicle, allowing buyers and sellers to easily browse and compare listings, and facilitating communication and transactions.

Besides these advantages, there are also some disadvantages:

Limited personal interaction: Some buyers and sellers may prefer the personal interaction that comes with traditional methods of buying and selling vehicles, such as visiting a dealership or meeting in person to discuss the vehicle.

Risk of fraud: There is always a risk of fraud when conducting transactions online, and buyers may be hesitant to purchase a vehicle without seeing it in person or having the opportunity to test drive it.

Limited vehicle information: Online systems may not provide as much information about a vehicle as a physical inspection or test drive would.

Difficulty verifying the condition of the vehicle: It can be difficult to verify the condition of a vehicle from photos and descriptions alone.

Limited negotiation: Online systems may not offer as much opportunity for negotiation between buyers and sellers as traditional methods.

In conclusion, our online vehicle for sale system is a platform that allows individuals or businesses to buy and sell vehicles over the internet. This system typically have a website or app that users can access to browse available vehicles, communicate with sellers or buyers, and complete transactions.

To use the system, a seller would typically list their vehicle for sale by creating a listing that includes information about the vehicle, such as its make, model, year, condition, and price. The seller may also include photos or videos of the vehicle to help potential buyers get a better sense of what they are buying.

Buyers can then search for vehicles that meet their criteria, such as a specific make or model or a price range. They can also communicate with the seller through the system to ask questions or request additional information about the vehicle.

CHAPTER TWO

OPERATION LIST

- **def about():**about us page of website
- **def index():**main page of website
- **def register():**register page of website. The user registers in the system by entering the necessary information.
- **def login():**login page of website. The user logins in the system by entering the necessary information.
- **def logout():**logout page of website. The user logs out of the system
- **def dashboard():**dashboard page of website.The page where the user can add and manage the ads he/she added.
- **def admin():**admin page of website.Only users with admin authority can access this page. can access and manage all log records, all users and all advertisements.
- **def manageUsers():**manage users page. Only admins can access this page. has the authority to delete any user and give admin authority.
- **def manageAdverts():**manage adverts page. Only admins can access this page. has the authority to delete and manage any advert .
- **def giveAdminAuthority(id):**The user with the received id value is given admin authority.
- **def deleteUser(id):**The user with the received id value is deleted.
- **def addotomobile():**add otomobile advert page of website. The user adds advert in the system by entering the necessary information.
- **def otomobiles():**page listing all otomobile ads
- **def otomobileDetail(id):**The otomobile in the received id is shown in detail, together with its picture.
- **def searchotomobile():**Returns otomobiles that contain the given keyword in brand,series,color,model attributes

- **def sortOtomobile(by):**Sorts the otomobile ads according to the given keyword
- **def deleteotomobile(id):**deletes the otomobile advert with the given id
- **def updateotomobile(id):**update page of the otomobile ad on the given id
- **def addmotorcycles():**add motorcycle advert page of website. The user adds advert in the system by entering the necessary information.
- **def motorcycles():**page listing all motorcycle ads
- **def motorcycleDetail(id):**The motorcycle in the received id is shown in detail, together with its picture.
- **def updatemotorcycle(id):**update page of the motorcycle ad on the given id
- **def deletemotorcycle(id):**deletes the motorcycle advert with the given id
- **def searchmotorcycle():**Returns motorcycle that contain the given keyword in brand,color,model attributes
- **def sortmotorcycle(by):**Sorts the motorcyle ads according to the given keyword
- **def addwatercrafts():**add watercraft advert page of website. The user adds advert in the system by entering the necessary information.
- **def watercrafts():**page listing all watercraft ads
- **def watercraftDetail(id):**The watercraft in the received id is shown in detail, together with its picture.
- **def updatewatercraft(id):**update page of the watercraft ad on the given id
- **def deletewatercraft(id):**deletes the watercraft advert with the given id
- **def searchwatercraft():**Returns otomobiles that contain the given keyword in ,color,year attributes
- **def sortwatercraft(by):**Sorts the watercraft ads according to the given keyword
- **def addaircrafts():**add aircraft advert page of website. The user adds advert in the system by entering the necessary information.
- **def aircrafts():**page listing all aircraft ads

- **def aircraftDetail(id):**The aircraft in the received id is shown in detail, together with its picture.

- **def updateaircraft(id):**update page of the aircraft ad on the given id

- **def deleteaircraft(id):**deletes the aircraft advert with the given id

- **def searchaircraft():**Returns aircraft that contain the given keyword in color,year attributes

- **def sortaircraft(by):**Sorts the aircraft ads according to the given keyword

CHAPTER THREE
SYSTEM ARCHITECTURE, CONSTRAINTS, CHALLENGES

Flask

Flask is a lightweight Python web framework that provides a set of tools and libraries for building web applications. It is designed to be easy to use and flexible, allowing developers to build a wide range of applications, from simple websites to complex web-based systems.

Here's a brief overview of how Flask works:

A developer writes a Python script that defines a Flask application. This script includes code for setting up routes (URLs) for the application and defining the functions that will be called when those routes are accessed.

The developer runs the Flask application using a Flask development server. This server listens for HTTP requests and routes them to the appropriate function in the Flask application.

When a user accesses a route in the Flask application, the server runs the corresponding function and returns the output to the user's web browser. The function can generate HTML, XML, or any other type of content that can be sent over HTTP.

The Flask application can interact with other resources, such as a database or external APIs, as needed to generate the response to the user's request.

Overall, Flask makes it easy for developers to create web applications by providing a simple and flexible set of tools for defining routes, handling requests, and generating responses.

The architecture of Flask is based on the Model-View-Controller (MVC) pattern, which separates the application into three main components: the model, which represents the data and logic of the application; the view, which represents the user interface of the application; and the controller, which handles the communication between the model and the view.

One of the main constraints of Flask is that it is a web framework, which means that it is designed to build web applications and is not suitable for building other types of software. Additionally, Flask is a microframework, which means that it is designed to be lightweight and flexible, but may not provide as many features or as much support as a full-stack framework.

Some of the challenges that developers may face when working with Flask include:

Scaling the application to handle a large volume of traffic or a high number of concurrent users.

Managing the complexity of the application as it grows and evolves, especially if the developer is using Flask in conjunction with other tools and libraries.

Ensuring the security and stability of the application, especially in production environments.

Managing the performance of the application, especially in applications with a large volume of data or a high number of concurrent users.

Overall, Flask is a popular and widely used web framework for building web applications in Python, but it can present certain challenges for developers, especially in large or complex applications.

HTML (HyperText Markup Language)

HTML (HyperText Markup Language) is a markup language used to structure and format content on the web. It is used to define the structure and layout of a web page, and to specify how content should be displayed in a web browser.

Here's a brief overview of how HTML works:

A developer writes an HTML document using a text editor or other tool. The document consists of a series of elements, which are represented by tags. Each tag defines a specific type of content, such as a paragraph, a heading, or a link.

The developer saves the HTML document with a .html file extension and uploads it to a web server.

When a user accesses the HTML document in their web browser, the browser reads the HTML code and uses it to render the page on the user's screen. The browser interprets the tags and uses them to determine how the content should be displayed, such as what font to use, what color to make the text, and how to lay out the page.

The browser also follows any links that are defined in the HTML document, allowing the user to navigate to other pages or resources on the web.

Overall, HTML provides a way for developers to structure and format content on the web, and for web browsers to display that content to users.

HTML (HyperText Markup Language) is a markup language used to structure and format content on the web. It is not a programming language and does not have a specific architecture or runtime environment. Instead, it is interpreted by web browsers and used to render the content of web pages.

One of the main constraints of HTML is that it is primarily used to structure and format content, rather than to define the behavior or functionality of a web page. This means that developers need to use other technologies, such as JavaScript, to add interactivity and dynamic behavior to a web page.

Some of the challenges that developers may face when working with HTML include:

Ensuring compatibility with different web browsers, as different browsers may interpret HTML differently.

Managing the complexity of larger HTML documents, especially as the content grows and evolves.

Ensuring the accessibility of the HTML content, especially for users with disabilities or specialized needs.

Managing the security of the HTML content and protecting against vulnerabilities such as cross-site scripting (XSS) attacks.

Overall, HTML is a fundamental building block of the web and is widely used to structure and format content on the internet. However, it can present certain challenges for developers, especially in large or complex applications.

CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) is a stylesheet language used to describe the look and formatting of a document written in HTML. It is used to specify how HTML elements should be displayed on the web page, such as the font, color, and layout of text, as well as the appearance of other elements like images and forms.

Here's a brief overview of how CSS works:

A developer writes a CSS stylesheet using a text editor or other tool. The stylesheet consists of a series of rules that specify how different HTML elements should be styled. Each rule consists of a selector, which specifies which elements the rule applies to, and a declaration, which specifies the styling properties that should be applied to those elements.

The developer links the CSS stylesheet to the HTML document using a <link> element in the <head> of the HTML document. When a user accesses the HTML document in their web browser, the browser reads the HTML code and uses it to render the page on the user's screen. The browser also reads the linked CSS stylesheet and applies the styles specified in the stylesheet to the HTML elements on the page.

The CSS styles are used to control the appearance of the HTML elements, such as the font, color, and layout of the text, as well as the appearance of other elements like images and forms.

Overall, CSS provides a way for developers to specify the look and formatting of a web page, and for web browsers to apply those styles when rendering the page for users.

CSS, or Cascading Style Sheets, is a stylesheet language used for describing the look and formatting of a document written in HTML or XML. It is a simple but powerful tool that allows developers to separate the content of a website from its presentation, making it easier to maintain and update.

The architecture of CSS consists of a set of rules that specify how elements in an HTML or XML document should be displayed. These rules, called "styles," can be applied to individual elements, groups of elements, or the entire document. Styles can be defined in a number of ways, including inline (within the HTML element itself), in a separate stylesheet file, or in a <style> element within the document's head.

CSS has a number of constraints, or limitations, that developers must consider when using it. One of the main constraints is that CSS styles are applied to elements based on their position in the document, rather than their content or meaning. This means that developers must be careful when organizing their HTML elements in order to achieve the desired layout and style.

There are several key aspects of the architecture and constraints of CSS that you should be aware of as a developer:

- Cascading: One of the main features of CSS is its ability to cascade, or "flow," from one style rule to another. This means that multiple style rules can be applied to the same element, and the order in which they are applied determines which rule takes precedence.

- Specificity: When there are multiple style rules that apply to the same element, the specificity of the rule determines which rule takes precedence. A rule with a higher specificity will override a rule with a lower specificity.

- Inheritance: Another key feature of CSS is inheritance, which allows style rules to be passed down from parent elements to child elements. This means that if you apply a style rule to a parent element, it will also be applied to all of its child elements unless you specify otherwise.

- Box model: CSS uses a box model to describe the layout of elements on a webpage. Each element is represented as a box, with dimensions that include the width, height, padding, border, and margin. Understanding the box model is essential for creating complex layouts using CSS.

Some of the challenges of using CSS include:

- Maintaining consistency: It can be difficult to ensure that styles are consistently applied across all pages of a website, especially if the site is large or has multiple contributors.

- Debugging: Debugging CSS can be time-consuming, as it is not always clear which styles are being applied to an element or why they are not being applied as expected.

- Cross-browser compatibility: Different web browsers can interpret and render CSS differently, so developers must ensure that their styles are compatible with a variety of browsers and devices.

- Responsive design: With the increasing use of mobile devices to access the web, developers must design websites that are responsive and adapt to different screen sizes and resolutions. This requires careful planning and testing to ensure that styles are applied correctly on different devices.

JavaScript

JavaScript is a programming language that is commonly used to add interactivity and dynamic behavior to web pages. It is a client-side scripting language, which means that it is executed by the web browser on the user's computer, rather than on the server where the web page is hosted.

Here's a brief overview of how JavaScript works:

A developer writes a JavaScript script using a text editor or other tool. The script consists of a series of statements that define the actions that should be performed by the browser.

The developer embeds the JavaScript script in an HTML document using a <script> element. The script can be included either in the <head> of the HTML document or in the <body> of the document.

When a user accesses the HTML document in their web browser, the browser reads the HTML code and uses it to render the page on the user's screen. The browser also reads the embedded JavaScript script and executes the code in the script.

The JavaScript code can interact with the HTML elements on the page, such as by changing the content of a text element, modifying the appearance of an image, or responding to user input.

Overall, JavaScript provides a way for developers to add interactivity and dynamic behavior to web pages, and for web browsers to execute that code and provide a more engaging and interactive experience for users.

The architecture of JavaScript is based on the execution of scripts in a web browser or other runtime environment. JavaScript is a client-side language, which means that it is executed by the web browser on the user's computer, rather than on the server where the web page is hosted.

One of the main constraints of JavaScript is that it is a single-threaded language, which means that it can only execute one task at a time. This can be a constraint for developers who need to perform multiple tasks concurrently or who need to execute tasks in parallel.

Some of the challenges that developers may face when working with JavaScript include:

Ensuring compatibility with different web browsers, as different browsers may implement JavaScript differently.

Managing the complexity of larger JavaScript applications, especially as the codebase grows and evolves.

Ensuring the performance and efficiency of the JavaScript code, especially in applications with a large volume of data or a high number of concurrent users.

Managing the security of the JavaScript code and protecting against vulnerabilities such as cross-site scripting (XSS) attacks.

Overall, JavaScript is a powerful and widely used programming language for building web applications, but it can present certain challenges for developers, especially in large or complex applications.

MySQL

MySQL is a popular open-source relational database management system (RDBMS). It is used to store and manage data for a wide variety of applications, such as websites, online stores, and business applications.

Here's a brief overview of how MySQL works:

A developer creates a database and defines its structure using SQL (Structured Query Language). This includes creating tables to store data, and defining the columns in each table and the data types of those columns.

The developer writes SQL statements to insert, update, delete, and retrieve data from the database. These statements are executed by the MySQL server, which stores the data in the tables and returns the results to the developer.

The MySQL server can be accessed from a variety of programs and languages, including web servers, application servers, and programming languages like PHP, Python, and Java.

The MySQL server manages the data in the database, including tasks such as indexing the data for faster searching, and ensuring the consistency and integrity of the data.

Overall, MySQL provides a way for developers to store and manage data in a structured and organized way, and to access and manipulate that data using SQL statements.

The architecture of MySQL is based on a client-server model, in which the MySQL server handles the storage and management of data, and clients connect to the server to access and manipulate the data. The MySQL server is written in C and C++ and runs on a variety of platforms, including Linux, Unix, and Windows.

One of the main constraints of MySQL is that it is a relational database management system (RDBMS), which means that it stores data in tables and uses SQL (Structured Query Language) to manipulate the data. This can be a constraint for developers who are used to working with NoSQL databases, which use a different data model and query language.

Some of the challenges that developers may face when working with MySQL include:

- Scaling the database to handle a large volume of data or a high number of concurrent users.

- Ensuring the performance and reliability of the database, especially in high-traffic applications.

- Maintaining the security of the database and protecting against data breaches.

- Managing the complexity of the database schema and SQL queries as the application grows and evolves.

- Overall, MySQL is a powerful and widely used database management system, but it can present certain challenges for developers, especially in large or complex applications.

CHAPTER FOUR
USED TECHNOLOGY, TOOLS AND PROGRAMMİNG LANGUAGES

HTML: We used HTML for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

CSS: We used CSS for describe the look and formatting of a document written in HTML.

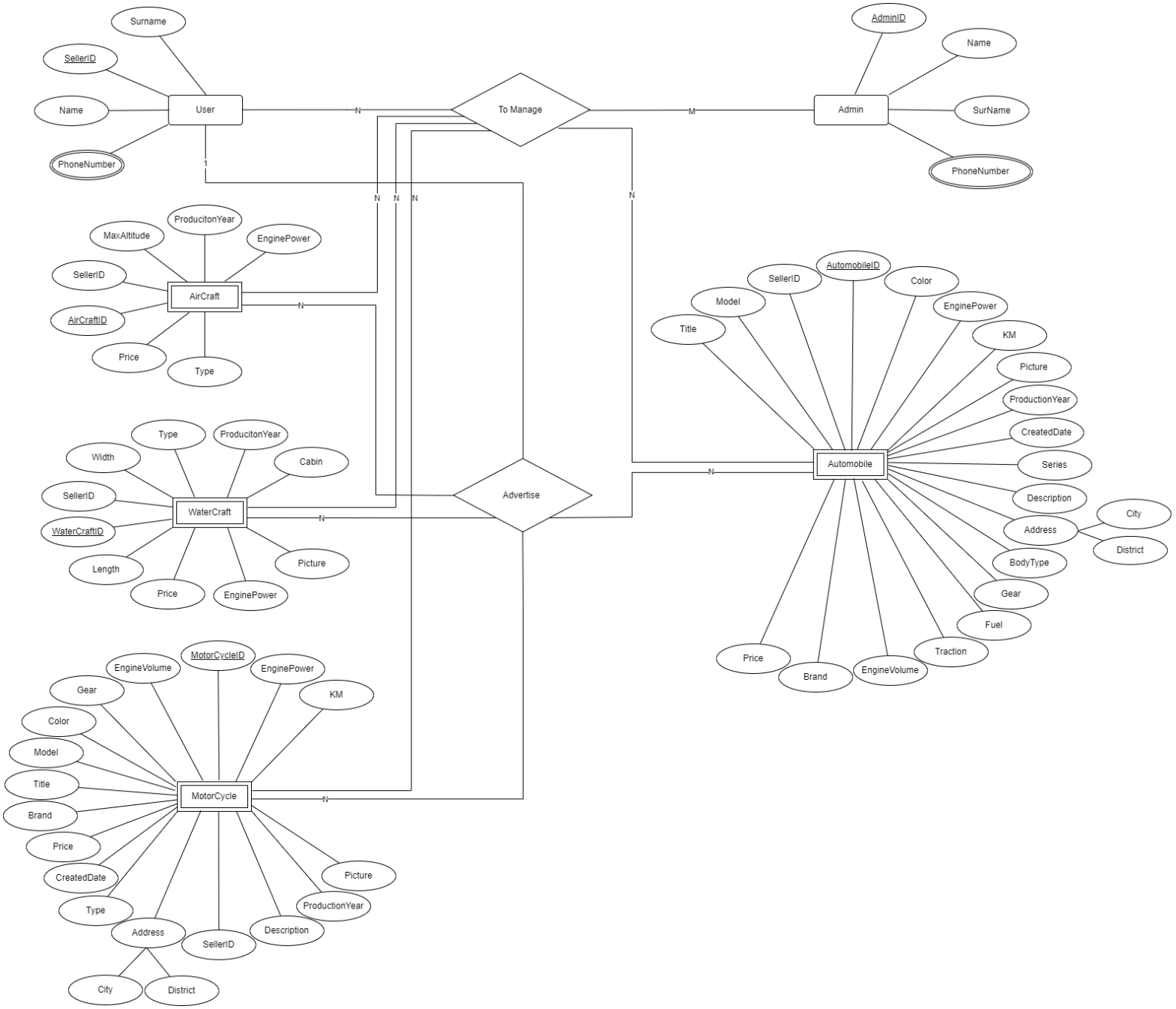
JAVASCRIPT: We used JavaScript create interactive effects within web browsers.

FLASK: We used flask to build web applications.

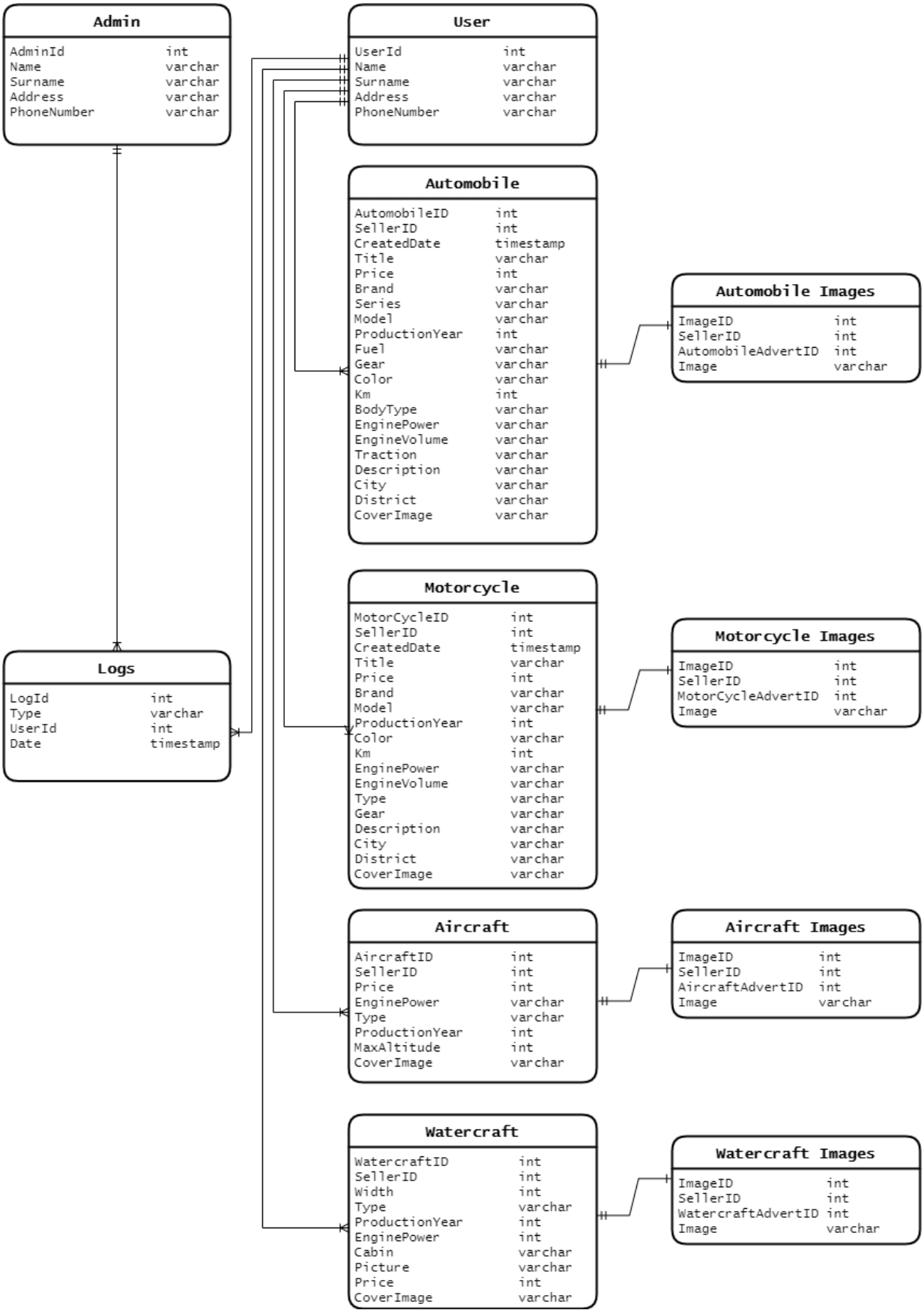
MYSQL: We used MYSQL in web applications to store and retrieve data.

CHAPTER FIVE
ER DIAGRAMS AND SCHEMA

ER DIAGRAM



SCHEMA


Admin
AdminId int
Name varchar
Surname varchar
Address varchar
PhoneNumber varchar
User
UserId int
Name varchar
Surname varchar
Address varchar
PhoneNumber varchar
Automobile
AutomobileID int
SellerID int
CreatedDate timestamp
Title varchar
Price int
Brand varchar
Series varchar
Model varchar
ProductionYear int
Fuel varchar
Gear varchar
Color varchar
Km int
BodyType varchar
EnginePower varchar
EngineVolume varchar
Traction varchar
Description varchar
City varchar
District varchar
CoverImage varchar
Automobile Images
ImageID int
SellerID int
AutomobileAdvertID int
Image varchar
Logs
LogId int
Type varchar
UserId int
Date timestamp
Motorcycle
MotorCycleID int
SellerID int
CreatedDate timestamp
Title varchar
Price int
Brand varchar
Model varchar
ProductionYear int
Color varchar
Km int
EnginePower varchar
EngineVolume varchar
Type varchar
Gear varchar
Description varchar
City varchar
District varchar
CoverImage varchar
Motorcycle Images
ImageID int
SellerID int
MotorCycleAdvertID int
Image varchar
Aircraft
AircraftID int
SellerID int
Price int
EnginePower varchar
Type varchar
ProductionYear int
MaxAltitude int
CoverImage varchar
Aircraft Images
ImageID int
SellerID int
AircraftAdvertID int
Image varchar
Watercraft
WatercraftID int
SellerID int
Width int
Type varchar
ProductionYear int
EnginePower int
Cabin varchar
Picture varchar
Price int
CoverImage varchar
Watercraft Images
ImageID int
SellerID int
WatercraftAdvertID int
Image varchar

CHAPTER SIX
SQL STATEMENTS

- CREATE PROCEDURE IF NOT EXISTS getUserInfo(IN user_name VARCHAR(255))
  BEGIN
    SELECT * FROM users WHERE username = user_name;
  END

- INSERT INTO users(name,email,username,password,phone_number) VALUES(%s,%s,%s,%s,%s)

- INSERT INTO logs(type,user_id) VALUES(%s,%s)

- SELECT * FROM admins WHERE user_id=%s

- INSERT INTO logs(type,user_id) VALUES(%s,%s)

- call getUserInfo(%s)

- SELECT * FROM otomobiles WHERE seller_id = %s

- SELECT * FROM motorcycles WHERE seller_id = %s

- SELECT * FROM watercrafts WHERE seller_id = %s

- SELECT * FROM aircrafts WHERE seller_id = %s

- CREATE VIEW IF NOT EXISTS AllAdmins AS

- SELECT logs.id ,logs.type,logs.date,logs.user_id,users.username FROM logs
  INNER JOIN users
  ON logs.user_id=users.id
  ORDER BY logs.date DESC;

- SELECT users.id,users.username,users.email,admins.user_id FROM users
  LEFT JOIN admins
  ON admins.user_id=users.id;

- SELECT * FROM otomobiles

- SELECT * FROM motorcycles

- SELECT * FROM watercrafts

- SELECT * FROM aircrafts

- INSERT INTO admins(user_id) VALUES(%s);

- DELETE FROM users WHERE id=%s;

- INSERT INTO otomobiles(seller_id,title,price,brand,series,model,year,fuel,gear,color,km,body_type,engine_power,engine_volume,traction,city,district,description) VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)

- SELECT * FROM otomobiles WHERE seller_id = %s ORDER BY created_date DESC LIMIT 1

- UPDATE otomobiles SET cover_image =%s WHERE id=%s

● INSERT INTO otomobile_images(seller_id,otomobile_advert_id,img) VALUES(%s,%s,%s)

● SELECT * FROM otomobiles where id =%s

● SELECT * FROM otomobile_images WHERE otomobile_advert_id=%s

● SELECT * FROM users WHERE id=%s

● Select * from otomobiles where brand like '%" + keyword +"%' OR series like '%"+ keyword+"%' OR color like '%"+ keyword +"%' OR model like '%"+ keyword +"%'

● SELECT * FROM otomobiles ORDER BY "+index+" "+direction+";

● SELECT * FROM admins where user_id=%s

● SELECT * FROM otomobiles where seller_id=%s AND id=%s

● DELETE FROM otomobiles where id =%s

● SELECT * FROM otomobiles where seller_id=%s AND id=%s

● UPDATE otomobiles SET title =%s , price=%s , brand=%s , series=%s , model=%s , year=%s , fuel=%s , gear=%s , color=%s , km=%s , body_type=%s , engine_power=%s , engine_volume=%s , traction=%s , city=%s , district=%s , description=%s WHERE id =%s

● INSERT INTO motorcycles(seller_id,title,price,brand,type,model,year,gear,color,km,engine_power,engine_volume,city,district,description) VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)

● SELECT * FROM motorcycles WHERE seller_id = %s ORDER BY created_date DESC LIMIT 1

● UPDATE motorcycles SET cover_image =%s WHERE id=%s

● INSERT INTO motorcycle_images(seller_id,motorcycle_advert_id,image) VALUES(%s,%s,%s)

● SELECT * FROM motorcycles

● SELECT * FROM motorcycles where id =%s

● SELECT * FROM motorcycle_images WHERE motorcycle_advert_id=%s

● SELECT * FROM users WHERE id=%s

● SELECT * FROM motorcycles where seller_id=%s AND id=%s

● UPDATE motorcycles SET title =%s , price=%s , brand=%s , type=%s , model=%s , year=%s , gear=%s , color=%s , km=%s , engine_power=%s , engine_volume=%s , city=%s , district=%s , description=%s WHERE id =%s

● SELECT * FROM admins where user_id=%s

● SELECT * FROM motorcycles where seller_id=%s AND id=%s

● DELETE FROM motorcycles where id =%s

● Select * from motorcycles where brand like '%" + keyword +"%' OR model like '%"+ keyword +"%' OR color like '%"+ keyword +"%'

● SELECT * FROM motorcycles ORDER BY "+index+" "+direction+";

● INSERT INTO watercrafts(seller_id,title,price,type,year,color,width,length,city,district,description) VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)

● SELECT * FROM watercrafts WHERE seller_id = %s ORDER BY created_date DESC LIMIT 1

● UPDATE watercrafts SET cover_image =%s WHERE id=%s

● INSERT INTO watercraft_images(seller_id,watercraft_advert_id,image) VALUES(%s,%s,%s)

● SELECT * FROM watercraft_images WHERE watercraft_advert_id=%s

● SELECT * FROM users WHERE id=%s

● SELECT * FROM watercrafts where seller_id=%s AND id=%s

● UPDATE watercrafts SET title =%s , price=%s ,   type=%s ,   year=%s , color=%s , width=%s , length=%s ,   city=%s , district=%s , description=%s WHERE id =%s

● SELECT * FROM admins where user_id=%s

● SELECT * FROM watercrafts where seller_id=%s AND id=%s

● DELETE FROM watercrafts where id =%s

● Select * from watercrafts where year like '%" + keyword +"%' OR color like '%"+ keyword +"%'

● SELECT * FROM watercrafts ORDER BY "+index+" "+direction+";

● INSERT INTO aircrafts(seller_id,title,price,type,year,color,width,max_altitude,city,district,description) VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)

● SELECT * FROM aircrafts WHERE seller_id = %s ORDER BY created_date DESC LIMIT 1

● UPDATE aircrafts SET cover_image =%s WHERE id=%s

● INSERT INTO aircraft_images(seller_id,aircraft_advert_id,image) VALUES(%s,%s,%s)

● SELECT * FROM aircraft_images WHERE aircraft_advert_id=%s

● SELECT * FROM users WHERE id=%s

● SELECT * FROM aircrafts where seller_id=%s AND id=%s

● UPDATE aircrafts SET title =%s , price=%s ,   type=%s ,   year=%s ,   color=%s , width=%s , max_altitude=%s ,   city=%s , district=%s , description=%s WHERE id =%s

● SELECT * FROM admins where user_id=%s

● SELECT * FROM aircrafts where seller_id=%s AND id=%s

● DELETE FROM aircrafts where id =%s

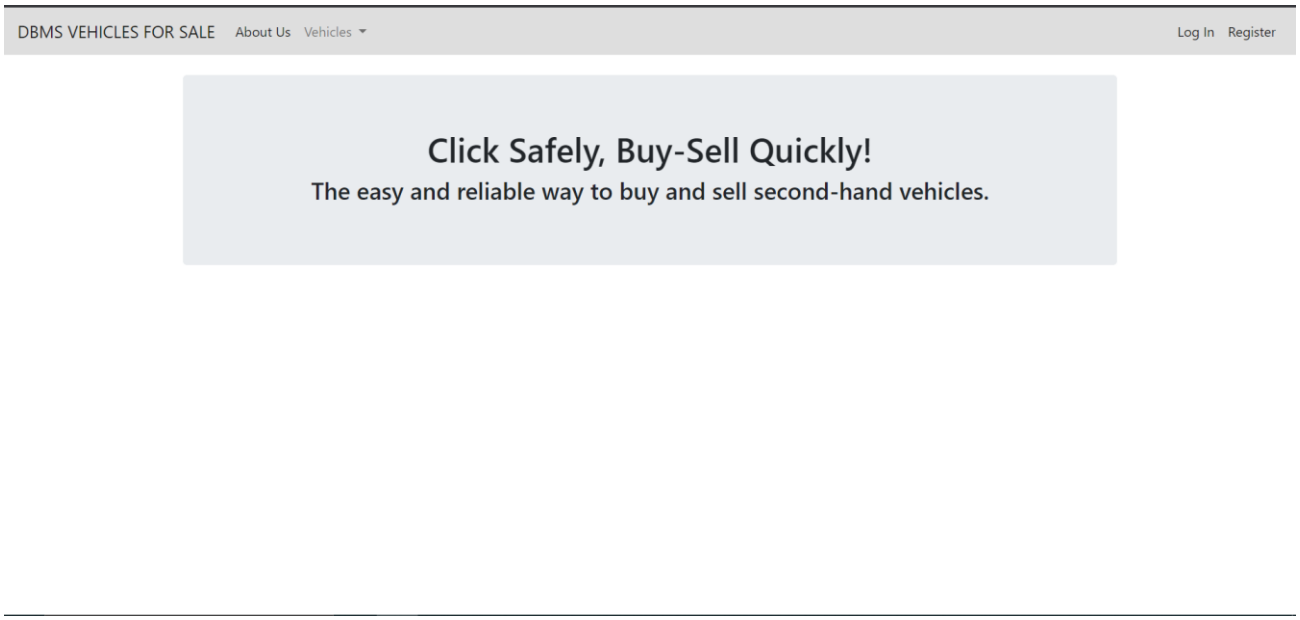
● Select * from aircrafts where year like '%" + keyword +"%' OR color like '%"+ keyword +"%'

● SELECT * FROM aircrafts ORDER BY "+index+" "+direction+";

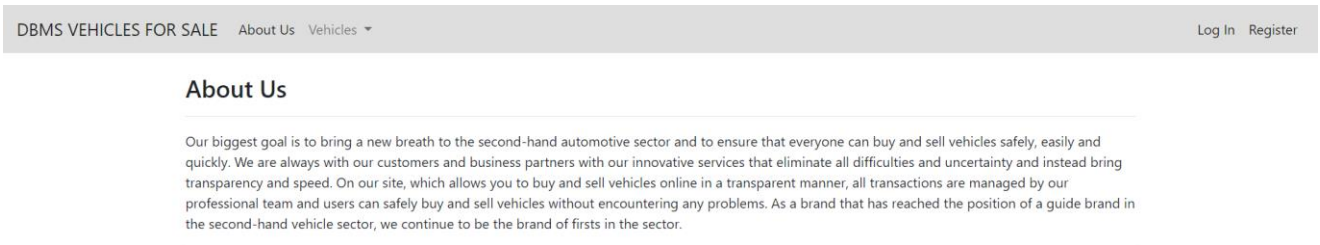
CHAPTER SEVEN

SCREESHOTS

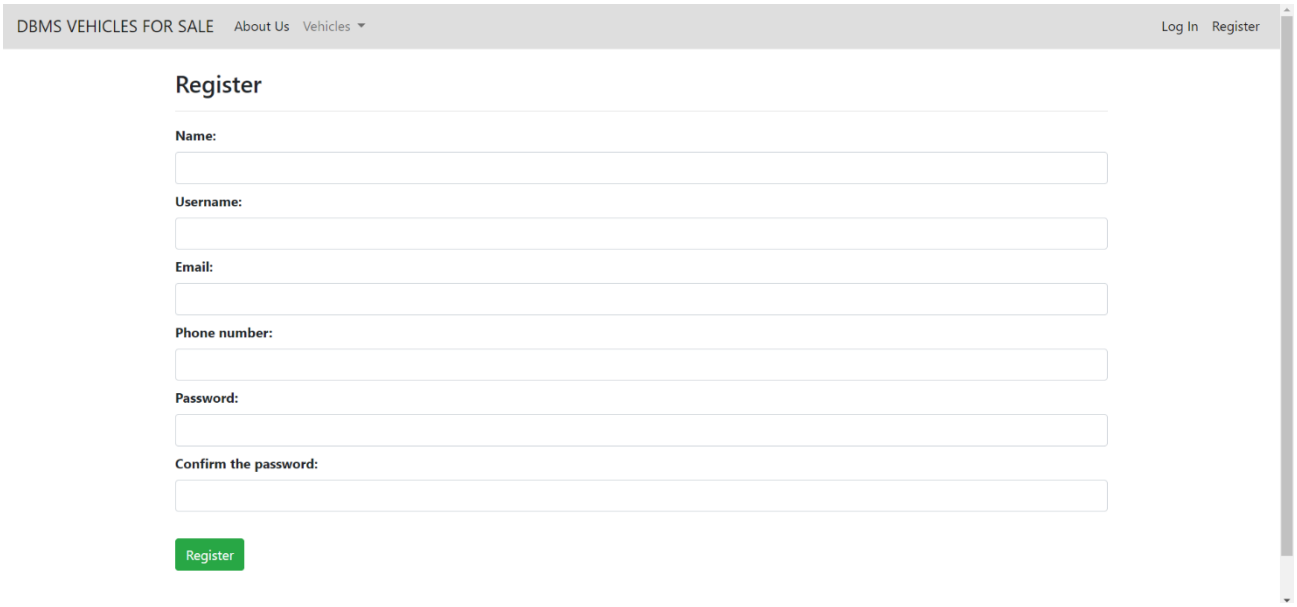
Home Page: main page of a website that is the first page a visitor sees when they visit the site. It includes links to the other pages on the site and provide a brief overview of the site's purpose and content.



About Us Page: It includes a brief history of the organization, an overview of its mission and values, and details about its products, services, or other areas of focus



Register Page: It is a page that allows users to create a new account for the site. It includes a form that asks the user to provide certain information, such as their name, email address, and a password. Once the user has completed the form and submitted it, a new account will be created for them and they will be able to log in to the site using the username and password they provided.



Login Page: It is a page that allows users to access their account on the site. It includes a form that asks the user to enter their username and password. Once the user has entered their login information and submitted the form, they will be logged in to their account and can access the features or content that are restricted to registered users.

Dashboard Page: The page that allows the user to add advertisements and to list and manage the advertisements they have added.

DBMS VEHICLES FOR SALE About Us Vehicles Dashboard Log Out

Dashboard

Welcome, ali

Add Advert

Your otomobile adverts:

Id	Image	Brand	Series	Model	Title	Year	Color	Price			
82		Mercedes	G	G63	-E VISION- 2022 MERCEDES G63 AMG HATASIZ BAYİ ÇIKIŞLI	2012	black	15,250,000	Go To Advert	Update	Delete

Your motorcycle adverts:

Id	Image	Brand	Model	Type	Title	Year	Color	Price			
3		Yamaha	YZF R25	supersport	2016 MODEL DÜŞÜK KM FIRSAT ARACI	2015	white	86,000	Go To Advert	Update	Delete

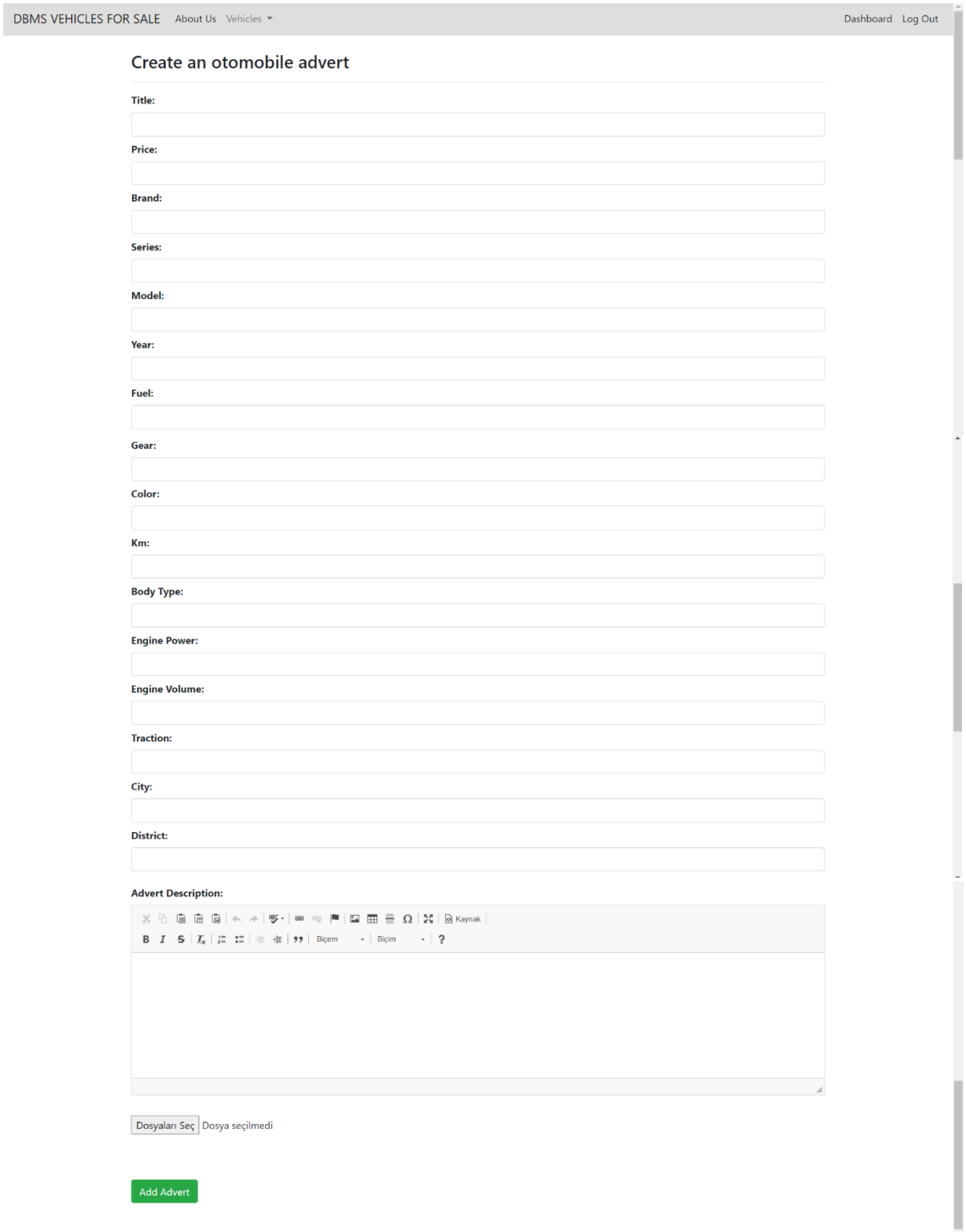
Your watercraft adverts:

Id	Image	Title	Year	Color	Price			
2		2014 | Azimut 70 | İstanbul Yachting	2014	white	27,645,000	Go To Advert	Update	Delete
3		Sahibinden Dolu Dolu Tekne	2019	white	850,000	Go To Advert	Update	Delete

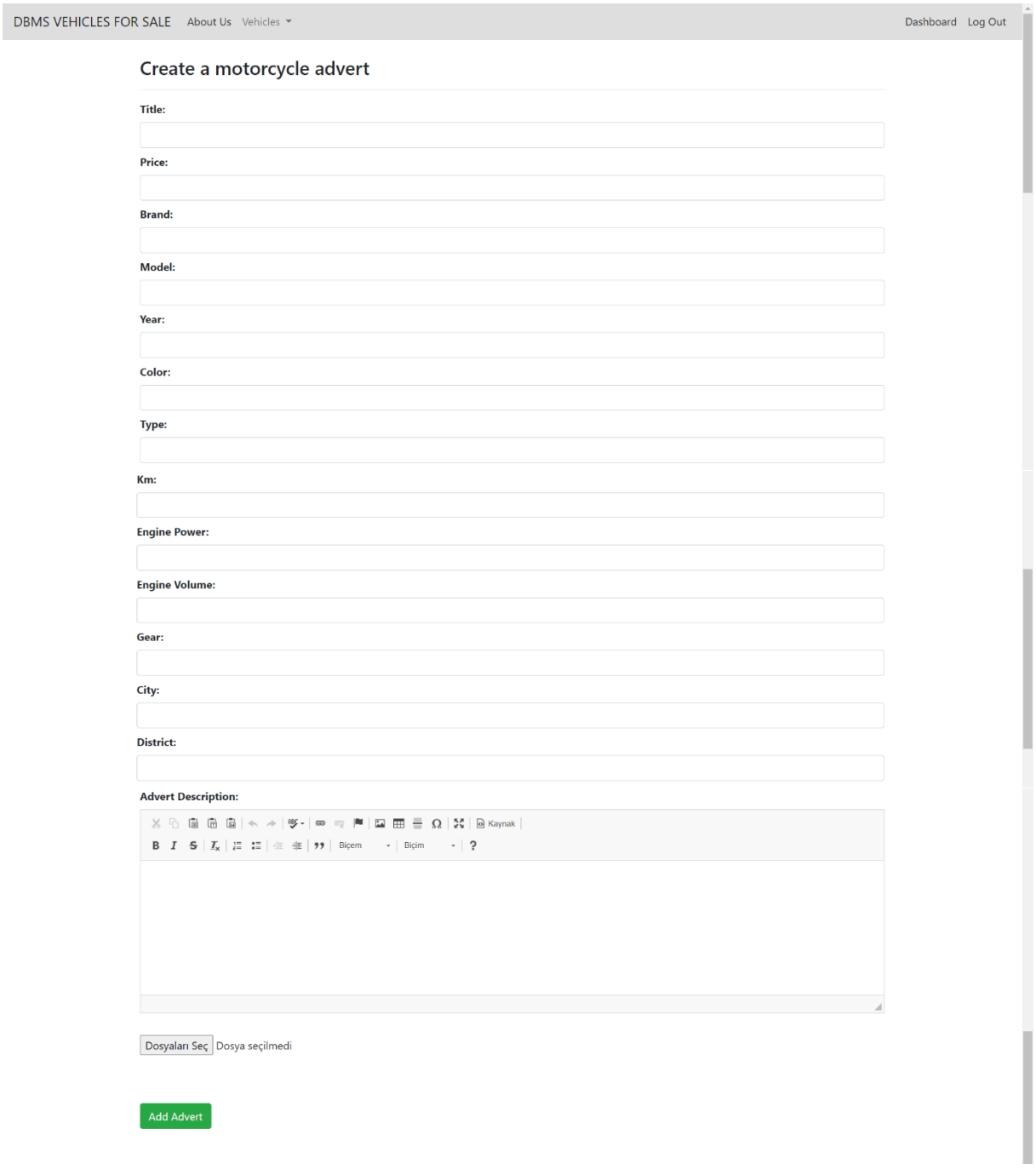
Your aircraft adverts:

Id	Image	Title	Year	Color	Price			
2		ERÇAL'DAN 2008 Diamond DA42 Twin Star Uçak- Airplane	2008	white	14,950,000	Go To Advert	Update	Delete
3		ZAMAN EN ÖNEMLİ DEĞERDİR.	2010	white	65,000,000	Go To Advert	Update	Delete

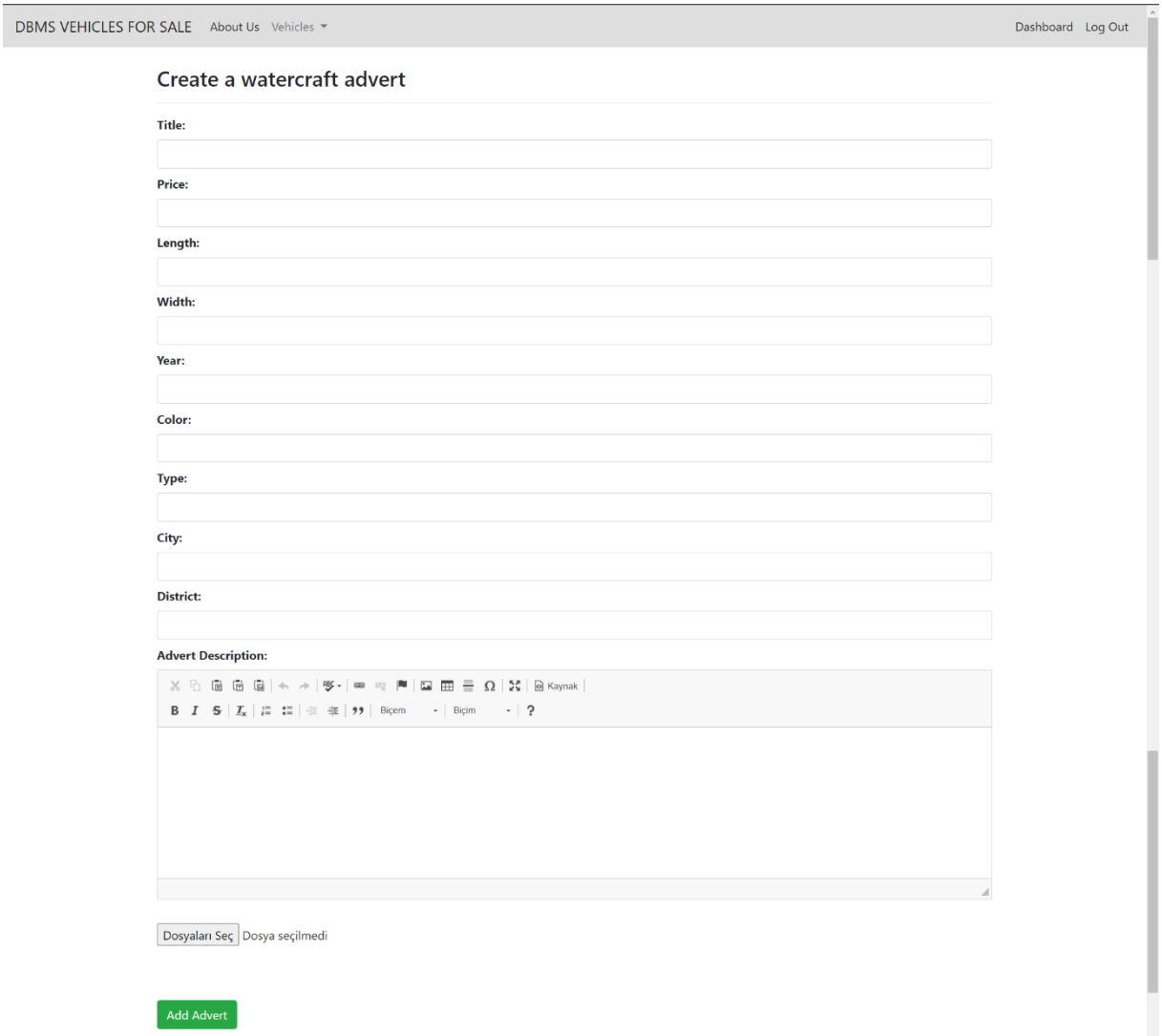
Create an otomobile advert page: It is a page that allows users to create and publish an advertisement for an otomobile on the site. This page include a form that asks the user to provide information about the otomobile they are selling, such as the make, price, year, and color.



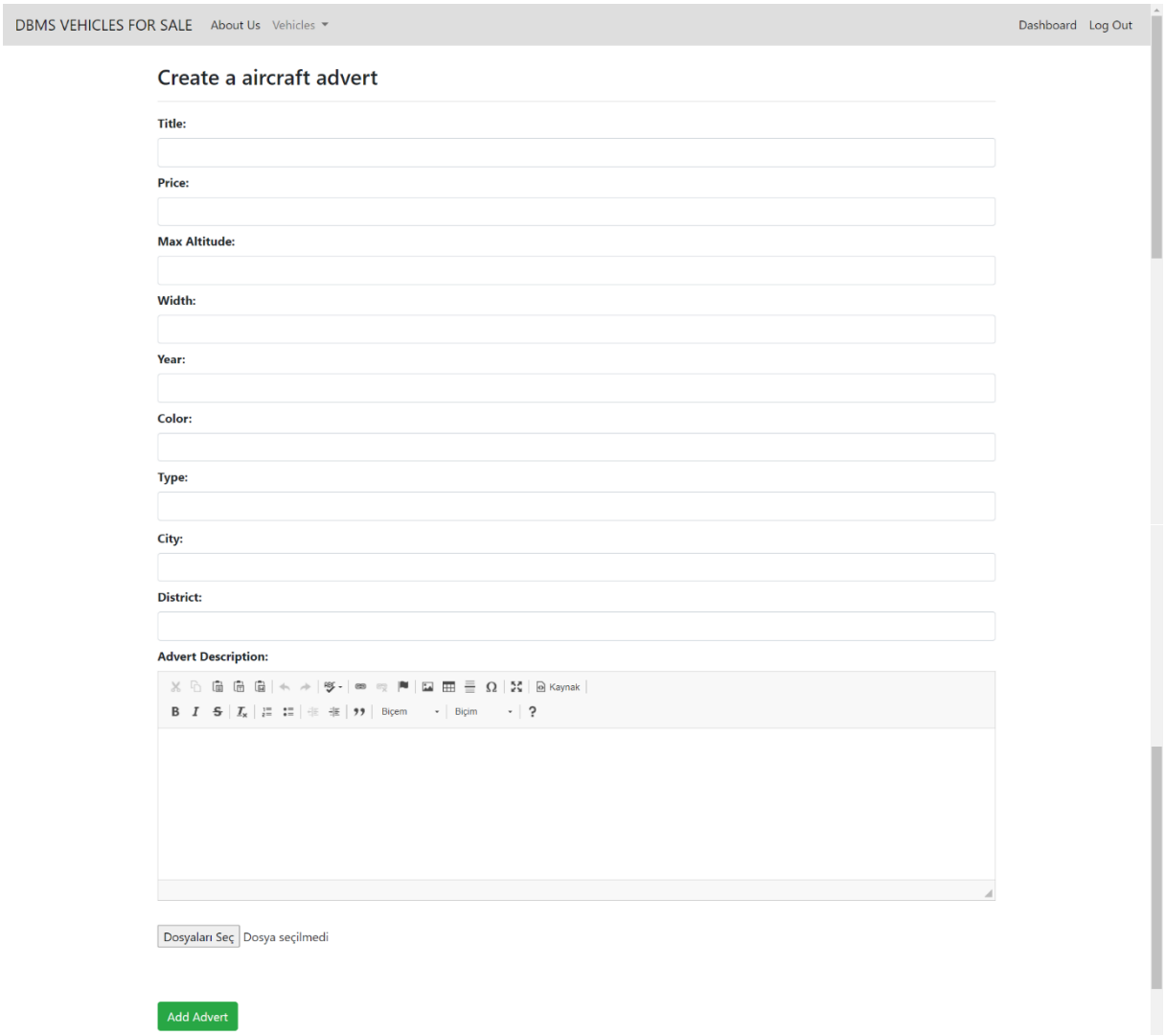
Create an motorcycle advert page: It is a page that allows users to create and publish an advertisement for an motorcycle on the site. This page include a form that asks the user to provide information about the motorcycle they are selling, such as the make, price, year, and color.



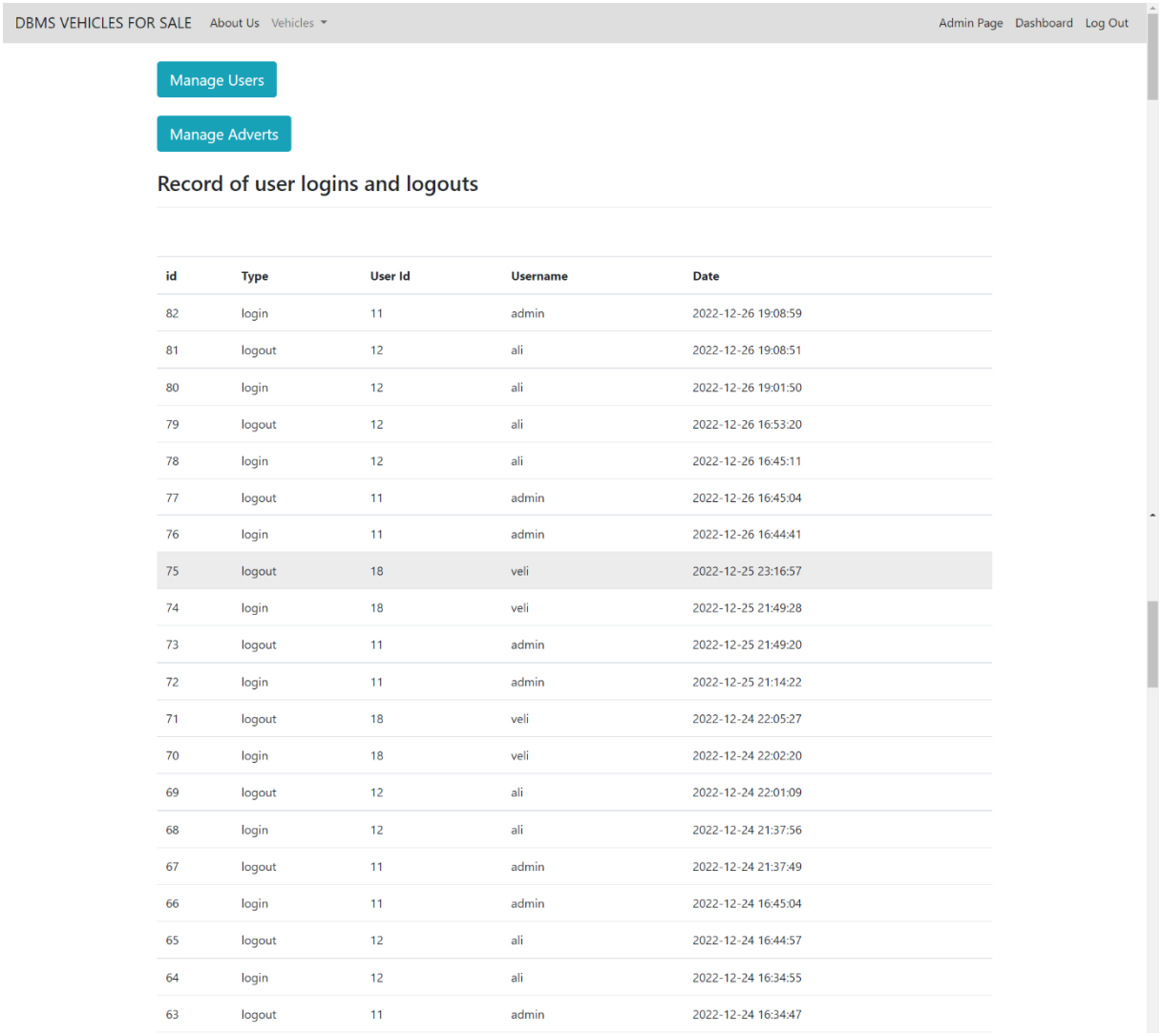
Create an watercraft advert page: It is a page that allows users to create and publish an advertisement for an watercraft on the site. This page include a form that asks the user to provide information about the watercraft they are selling, such as the make, price, year, and color.



Create an aircraft advert page: It is a page that allows users to create and publish an advertisement for an aircraft on the site. This page include a form that asks the user to provide information about the aircraft they are selling, such as the make, price, year, and color.

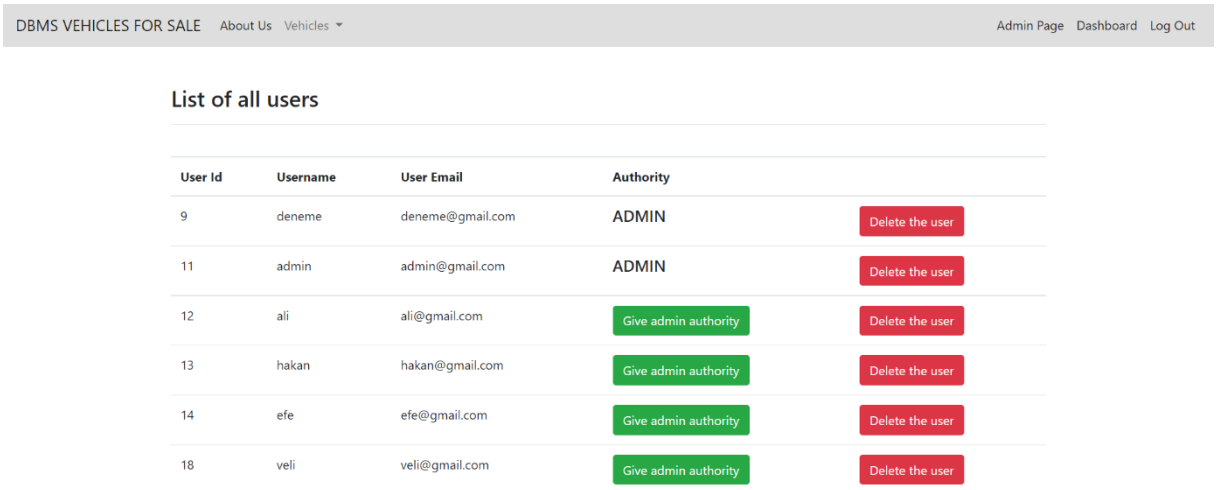


Admin Page: It is only accessible to the website's administrators, or those who have been granted special access to manage the site's content and settings.



id	Type	User Id	Username	Date
82	login	11	admin	2022-12-26 19:08:59
81	logout	12	ali	2022-12-26 19:08:51
80	login	12	ali	2022-12-26 19:01:50
79	logout	12	ali	2022-12-26 16:53:20
78	login	12	ali	2022-12-26 16:45:11
77	logout	11	admin	2022-12-26 16:45:04
76	login	11	admin	2022-12-26 16:44:41
75	logout	18	veli	2022-12-25 23:16:57
74	login	18	veli	2022-12-25 21:49:28
73	logout	11	admin	2022-12-25 21:49:20
72	login	11	admin	2022-12-25 21:14:22
71	logout	18	veli	2022-12-24 22:05:27
70	login	18	veli	2022-12-24 22:02:20
69	logout	12	ali	2022-12-24 22:01:09
68	login	12	ali	2022-12-24 21:37:56
67	logout	11	admin	2022-12-24 21:37:49
66	login	11	admin	2022-12-24 16:45:04
65	logout	12	ali	2022-12-24 16:44:57
64	login	12	ali	2022-12-24 16:34:55
63	logout	11	admin	2022-12-24 16:34:47

Manage users page(admin): The page where the admin manages users. can delete users or give admin privileges.



Manage Adverts page (admin): The page where the admin manages the adverts, can delete and view the adverts.

DBMS VEHICLES FOR SALE About Us Vehicles Admin Page Dashboard Log Out

Manage Adverts

All otomobile adverts:

Id	Image	Brand	Series	Model	Title	Year	Color	Price		
82		Mercedes	G	G63	-E VISION- 2022 MERCEDES G63 AMG HATASIZ BAYİ ÇIKIŞLI	2012	black	15,250,000	Go To Advert	Delete
83		Volkswagen	Passat	1.6 TDI BlueMotion Comfortline	2017MODEL LANSMAN RENK İÇİ BEJ+LED FAR+R LİNE GÖRÜNÜM+F1 DSG	2017	black	728,000	Go To Advert	Delete
84		Mercedes	C	C180 amg	İLK SAHİBİNDEN MERCEDES C180 (Hatasız-Boyasız)	2016	red	1,450,000	Go To Advert	Delete

All motorcycle adverts:

Id	Image	Brand	Model	Type	Title	Year	Color	Price		
3		Yamaha	YZF R25	supersport	2016 MODEL DÜŞÜK KM FIRSAT ARACI	2015	white	86,000	Go To Advert	Delete
4		Yamaha	YZF R1M	supersport	ÖZEN MOTOR - YAMAHA YZF R1M - 2022 - CARBON- 125 KM ! TR'DE TEK	2020	grey	880,000	Go To Advert	Delete
6		BMW	S1000rr	supersport	ULUS MOTOR'DAN ÖZEL ÜRETİM ORJINAL CARBON HP4 MOTOSPORT	2012	white	375,000	Go To Advert	Delete

All watercraft adverts:

Id	Image	Title	Year	Color	Price		
2		2014 | Azimut 70 | İstanbul Yachting	2014	white	27,645,000	Go To Advert	Delete
3		Sahibinden Dolu Dolu Tekne	2019	white	850,000	Go To Advert	Delete
4		TEMİZ 2018 QUICKSILVER 645 CRUISER MERCURY 150 HP	2018	black	1,000,000	Go To Advert	Delete

All aircraft adverts:

Id	Image	Title	Year	Color	Price		
2		ERÇAL'DAN 2008 Diamond DA42 Twin Star Uçak-Airplane	2008	white	14,950,000	Go To Advert	Delete
3		ZAMAN EN ÖNEMLİ DEĞERDİR.	2010	white	65,000,000	Go To Advert	Delete
4		ATEC 321 FAETA NG	2022	red	2,500,000	Go To Advert	Delete

CHAPTER EIGHT

ADDITIONAL PROPERTIES

● It is checked whether the email entered by the user while registering on the website has the correct form.

● While the password entered by the user is saved in the database, it is encrypted from beginning to end. Even the database admin cannot see it. Thus, it does not constitute a situation contrary to the law on the protection of personal data.

● The user cannot update or delete an advertisement that does not belong to him/her through the website pages. However, if a request such as editotomobile/82 or deleteotomobile/82 is found in the url section and this user does not have a vehicle with an id of 82, this request will also be blocked.

● On the code side, we have a decorator function called login_required. This function checks whether there is a user currently logged into the system. Before functions where user login is required, control is provided on a single line with @login_required. With this development, if the user makes requests such as /dashboard , /deleteotomobile/X, /editotomobile/X to the url section without logging into the system, the system will reject them and the user will be directed to the login screen.

● The ck editor was used to advert descriptions to the postings. In this way, it was possible to change the font of the text.