МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

КЫРГЫЗСКО-ГЕРМАНСКИЙ ИНСТИТУТ ПРИКЛАДНОЙ ИНФОРМАТИКИ

**КУРСОВАЯ РАБОТА**

По предмету «Языки программирования 3»

На тему: «АИС Автоматизация учета продажи автомобилей»

Руководитель: ст.преп. Алмезбек уулу А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Консультант по немецкому языку: ст.преп. Исмаилова Г. \_\_\_\_\_\_\_

Нормоконтроль: ст.преп. Турганов К.Б. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнили ст.гр AIN-2-20: Кубатбеков А.К.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Куштарбеков К.К. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Бакаев Д.Ж. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Бишкек 2021

**Inhalt**

[**Einführung** 2](#_Toc91189866)

[**Relevanz** 2](#_Toc91189867)

[**Ziel** 2](#_Toc91189868)

[**Sphäre** 2](#_Toc91189869)

[**Kontotypen** 3](#_Toc91189870)

[**Hauptteil** 4](#_Toc91189871)

[**Datenbankstruktur** 4](#_Toc91189872)

[**Frontend** 8](#_Toc91189873)

[**Spring Security** 10](#_Toc91189874)

[**Authentifizierungssystem** 13](#_Toc91189875)

[**Beschreibung der Controller** 14](#_Toc91189876)

[**Admin** 19](#_Toc91189877)

[**Fazit** 21](#_Toc91189878)

[**Ergebnisse** 21](#_Toc91189879)

[**Spezielle Anforderungen** 21](#_Toc91189880)

[**Literatur** 22](#_Toc91189881)

# **Einführung**

# **Relevanz**

Der Autokauf gehört heute zu den wichtigsten Anschaffungen im Leben eines Menschen. Die Leute sind sehr vorsichtig beim Autokauf. Schließlich ist das Auto bereits ein fester Bestandteil des Lebens der Städter. Laut Statistik gab es 2019 in Kirgisistan mehr als 1,3 Millionen Autos. Das bedeutet, dass die Zahl potenzieller Käufer sehr groß ist. Vergessen Sie auch nicht, dass der Automarkt jeden Tag wächst, es werden immer mehr Neuwagen aus anderen Ländern zum Verkauf angeboten, und gleichzeitig werden auch alte Autos verkauft.

Das Hauptproblem besteht jedoch darin, dass das Autokaufsystem nicht automatisiert ist. Alle Käufer müssen auf den Automarkt gehen und stundenlang unter der sengenden Sonne das richtige Auto auswählen. Auch Autoverkäufer haben es schwer, tagelang auf potenzielle Käufer zu warten, die ihr Auto auf dem Automarkt kaufen wollen. All diese Faktoren machen den Autokauf nicht zum angenehmsten Erlebnis.

# **Ziel**

Entwicklung einer Webanwendung mit dem Spring Boot-Webframework in Java, um den Kaufprozess zu vereinfachen. Die Website sollte modern aussehen und eine benutzerfreundliche und intuitive Oberfläche haben.

Aufgaben:

1. Eine Marktanalyse für den Bedarf unseres Produkts machen
2. Die Struktur der Datenbank für die Webanwendung überlegen
3. Ein Layout für die Site erstellen
4. Code für Registrierung und Autorisierung mit Spring Security schreiben
5. Webanwendungscode für das Layout schreiben
6. Administratorkonto hinzufügen

# **Sphäre**

Der Umfang unserer Studienarbeit ist der Online-Autoverkauf. Heute finden die meisten Käufe und Verkäufe von Autos in der Kirgisischen Republik offline auf dem Automarkt statt. Aber es gibt auch einige Websites, die Autos verkaufen und kaufen. Heute nimmt mashina.kg den Spitzenplatz unter diesen Websites ein.

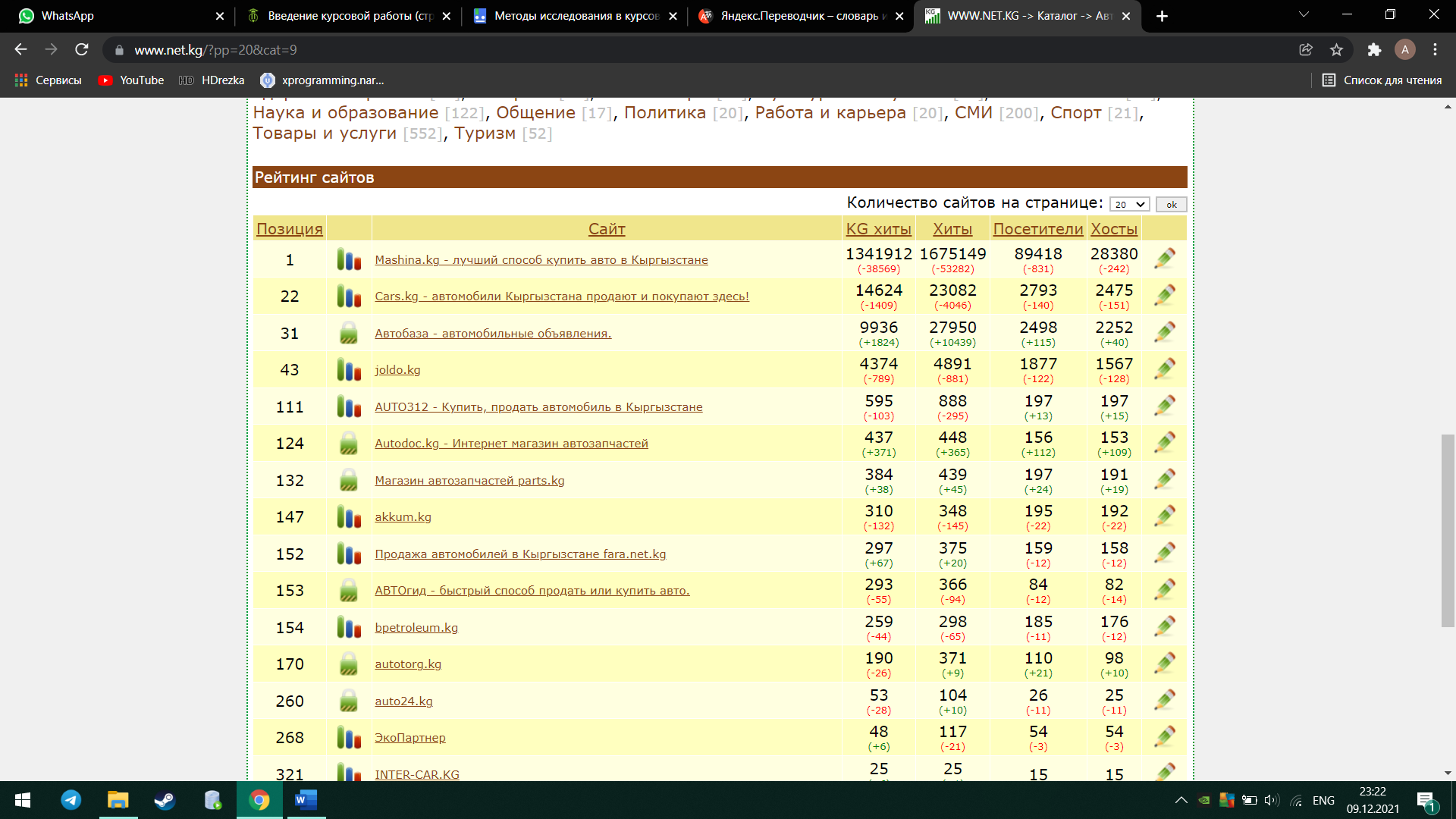


Abbildung 1. Die statistischen Daten der Webseiten nach dem Verkauf der Autos.

Laut Statistik der net.kg – Website ist zu erkennen, dass diese Website ein sehr großes Publikum hat. Die Site hatte etwa 90.000 Besucher pro Tag und es gibt 1.657.149 Besuche pro Tag. Das sind unglaubliche Zahlen, außerdem stehen sie unter allen Websites in Kirgisistan an erster Stelle. Andere Autohändler-Sites haben ebenfalls einen hohen Stellenwert.

Die Site mashina.kg hat ein sehr großes Publikum, aber dennoch ist die Site selbst ziemlich veraltet und hat ein ziemlich veraltetes Aussehen. Es hat eine zu komplexe Benutzeroberfläche, zu viele unnötige Details, die verwirrend sein können. Alles, was die Site bietet, ist eine große Datenbank mit allen Arten von Autos, nach denen Sie suchen.

# **Kontotypen**

Es gibt nur zwei Arten von Konten in der Webanwendung. Dies ist ein normaler Benutzer und Administrator.

* Benutzer - hat das Recht, Posts von Autos hinzuzufügen, ihre Autos zu bearbeiten, ihre Autos zu löschen, die Möglichkeit, Posts von Autos anzuzeigen und einen Kommentar unter jedem Post zu hinterlassen.
* Admin - hat die gleichen Rechte wie der Benutzer. Er hat immer noch das Recht, jeden Benutzer zu löschen, alle seine Autos zu sehen und kann jedes Fahrzeug löschen. Es kann nur einen Admin geben. Der Admin ist ein Benutzer mit username = "admin".

# **Hauptteil**

## **Datenbankstruktur**

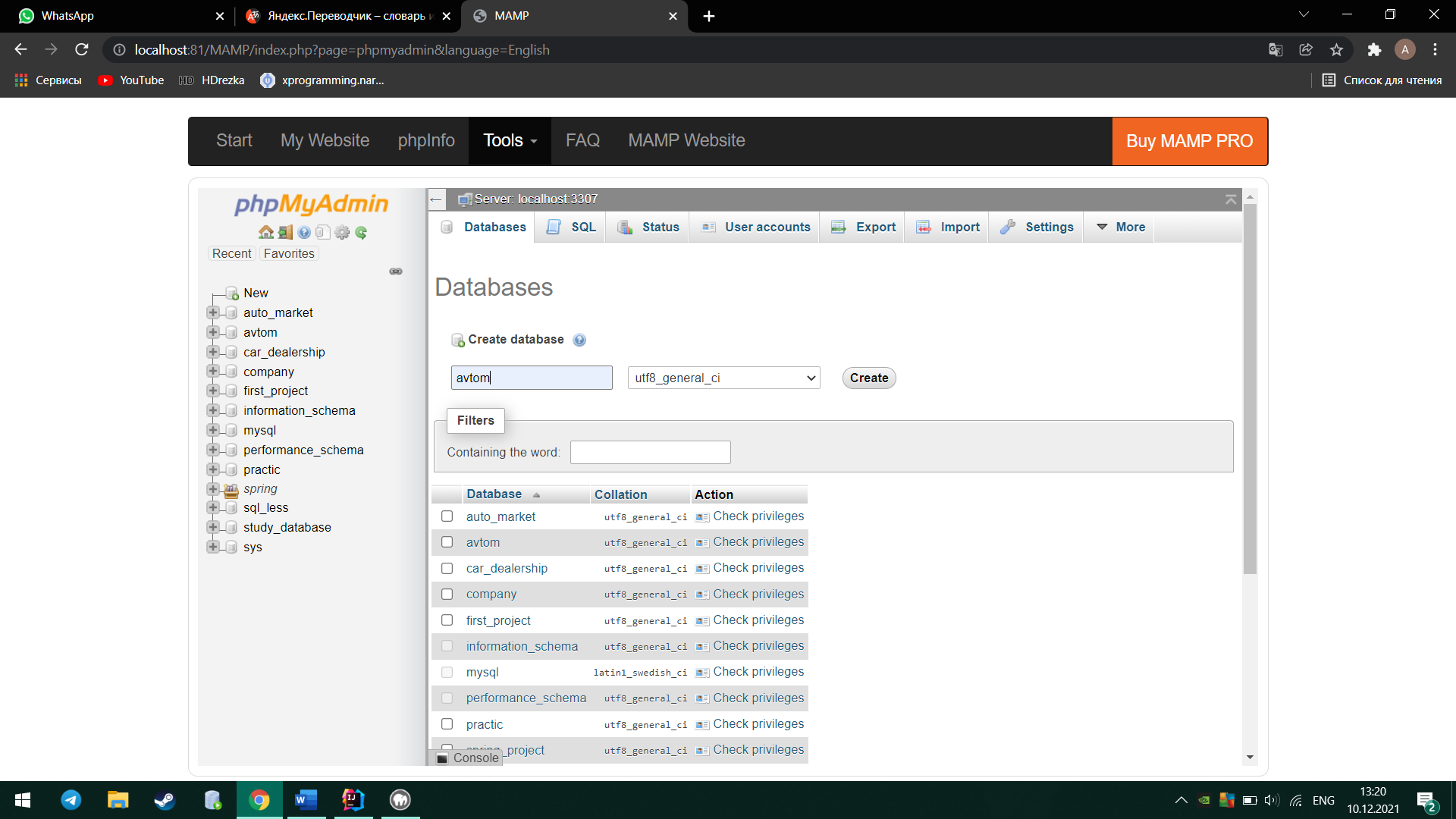
Die Datenbank der Webanwendung wird über MySQL ausgeführt. Wir haben die MAMP-Anwendung verwendet, um die Datenbank zu erstellen.

Abbildung 2. Erstellen einer Datenbank für eine Website mit MAMP

Dann haben wir die Abhängigkeiten für die Verwendung der MySQL-Datenbank in der Datei pom.xml verbunden.

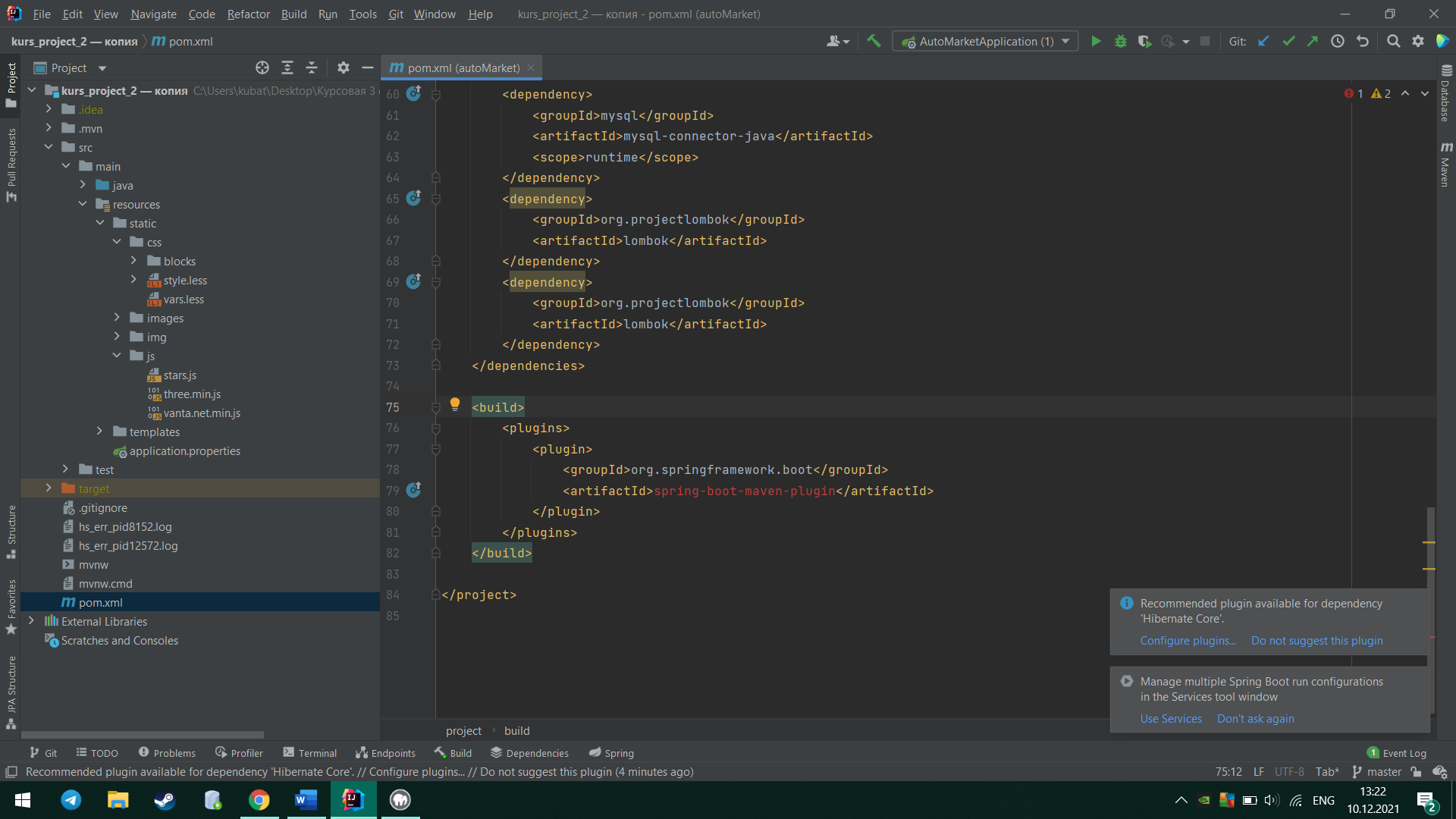


Abbildung 3. Fügen Sie Abhängigkeiten hinzu, um MySQL zu verwenden.

Schließlich wurden die notwendigen Daten für die Nutzung der avtom-Datenbank in die Datei application.properties geschrieben.

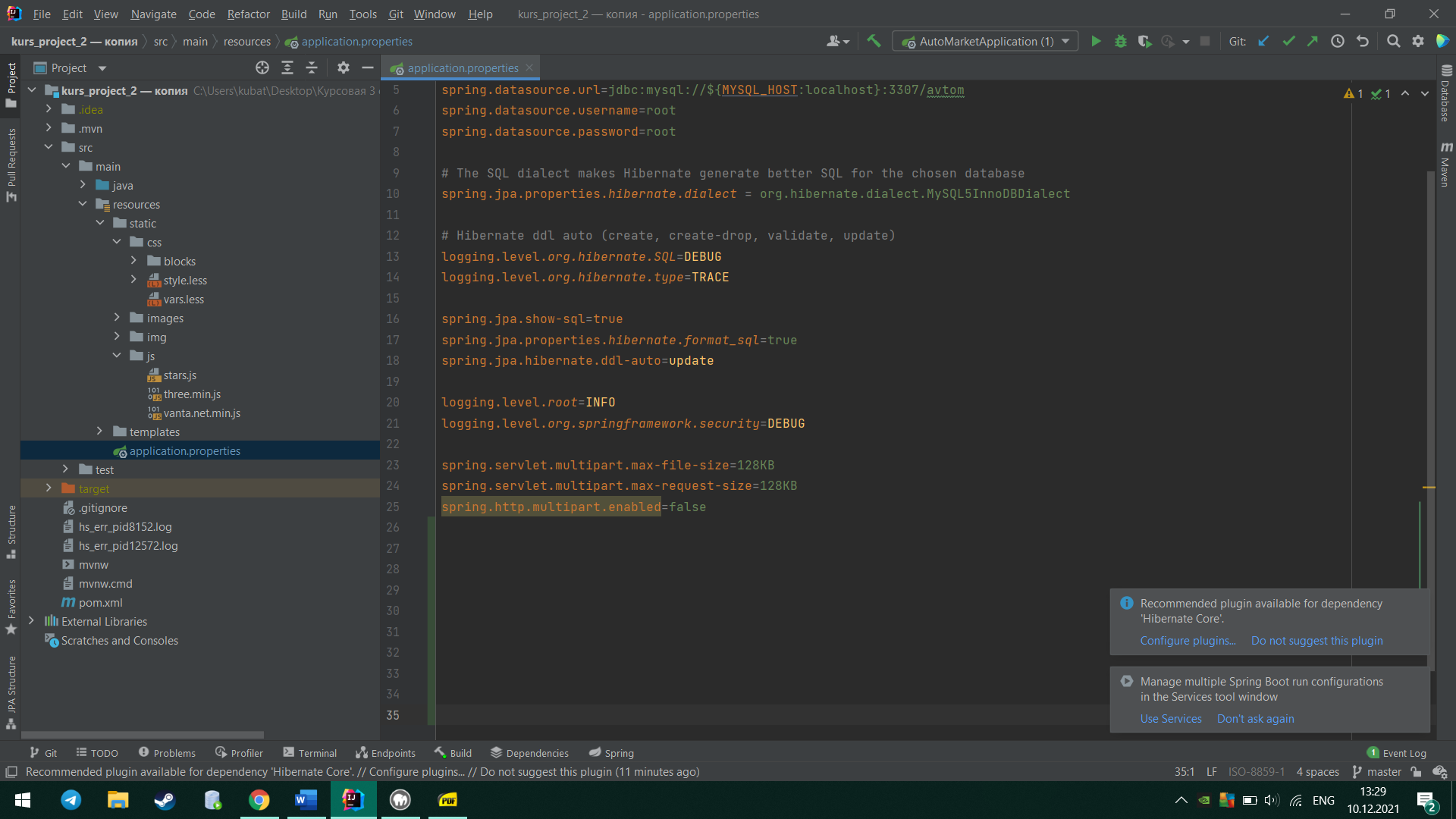


Abbildung 4. Gibt die Datenbank an, die wir verwenden werden.

Für jede Tabelle in der Datenbank wurden eigene Modelle und Repositories zum Anzeigen, Eingeben, Aktualisieren oder Löschen von Daten aus Tabellen geschrieben.

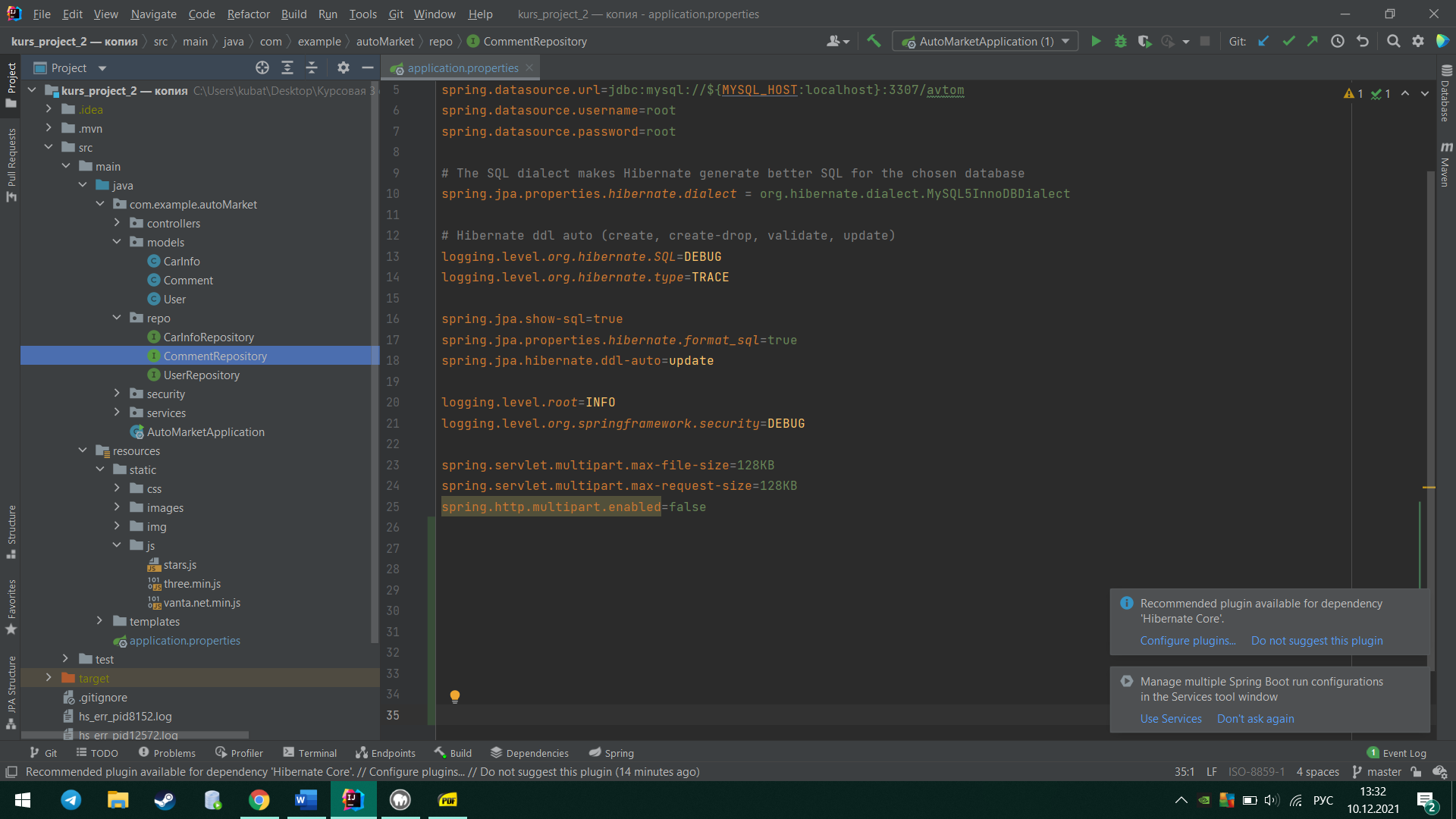


Abbildung 5. Modelle zum Erstellen einer Tabelle und ihres Repositorys.

Der Aufbau der Tabellen in der Datenbank sieht wie folgt aus:

Abbildung 6. Die Struktur der Tabellen in der Datenbank.

Die Datenbank enthält Tabellen wie:

* Car\_info - speichert Informationen über Autos
* id – Fahrzeugidentifikationsnummer
* brand – Automarke
* carcase – Autorahmen
* color – Autofarbe
* date – Datum der Veröffentlichung des zum Verkauf stehenden Fahrzeugs
* drive – Triebwerk
* engine\_volume – Hubraum
* fuel\_type – Treibstoffart
* horse\_power – Motorleistung
* image – Autobild
* mileage – Autokilometer
* model – Automodell
* price – Autopreis in Som
* steering\_wheel – Lenkradposition
* transmission – Übertragung
* year – Baujahr des Autos
* user\_id – die Identifikationsnummer des Nutzers, der das Fahrzeug veröffentlicht hat
* dop\_info – zusätzliche Fahrzeuginformationen
* vin – VIN-Code Auto
* User – speichert Informationen über Benutzer
* id – Benutzeridentifikationsnummer
* password – Benutzer-Passwort
* phone – Benutzertelefonnummer
* username – das Pseudonym, das der Nutzer auf unserer Website verwendet
* Comment – speichert Informationen über Kommentare
* id – Kommentar-ID
* text – Kommentartext
* car\_id – Identifikationsnummer des Fahrzeugs, unter dem der Kommentar hinterlassen wurde
* user\_id – die ID des Benutzers, der den Kommentar hinterlassen hat

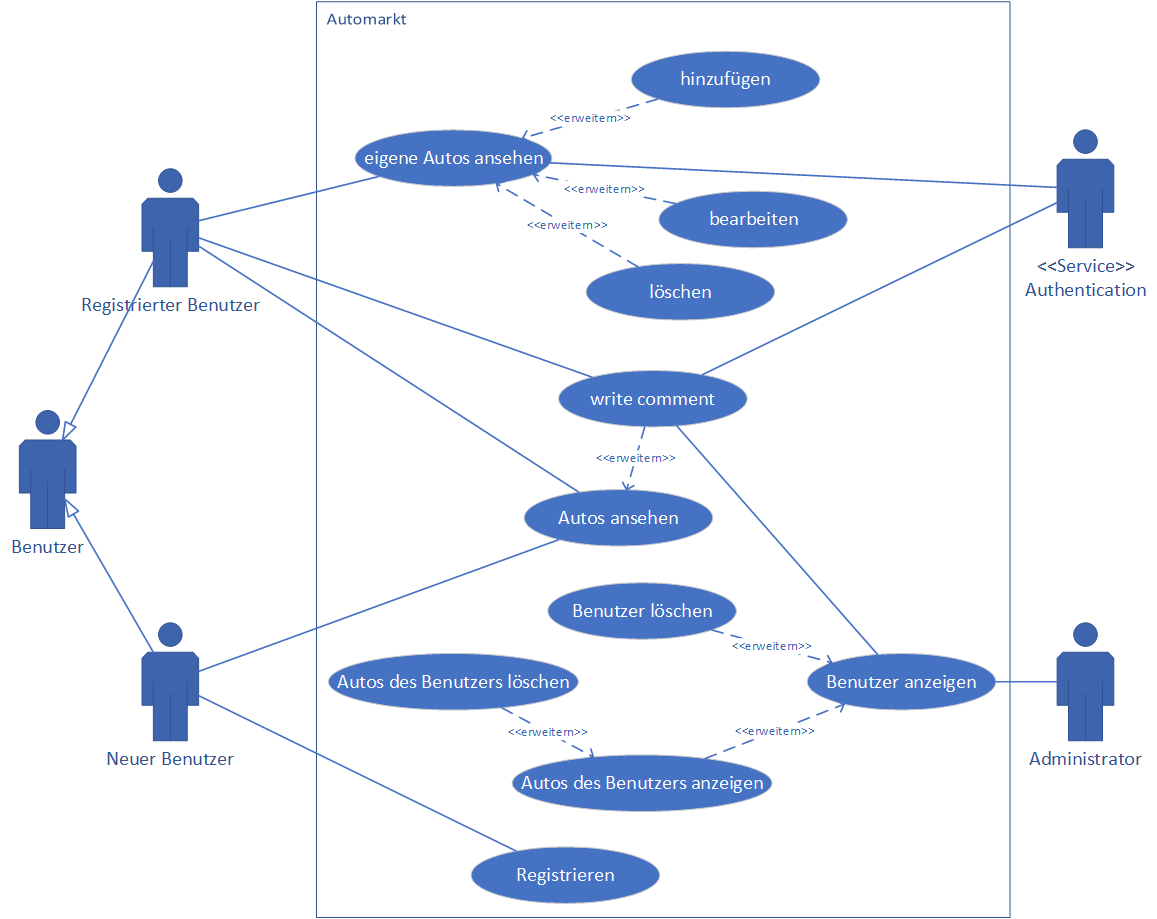
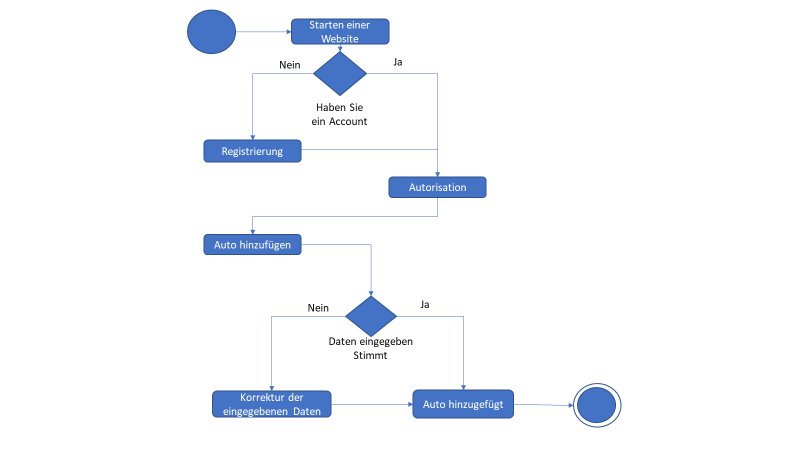
****Abbildung 7. Use case Diagramm.

Abbildung 8. Aktivitätsdiagramm.

## **Frontend**

Über JavaScript wird ein animierter Hintergrund angebunden. Die Dateien „three.min.js“ und „vanta.net.min.js“ enthalten den wichtigsten Hintergrundcode.



Abbildung 9. JavaScript-Dateien.

Die Datei "stars.js" implementiert die Verbindung zum Haupt-DIV-Tag in der HTML-Datei ("backg") und veränderbare Parameter, damit Sie das Erscheinungsbild des Hintergrunds ändern können.



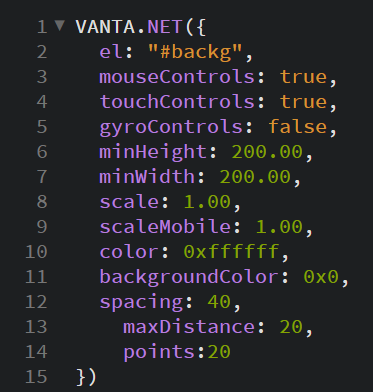


Abbildung 10. Hinzufügen von Animationen mit JS.

Darüber hinaus werden SCRIPT-Tags in die HTML-Datei geschrieben, damit die JavaScript-Datei korrekt funktioniert.

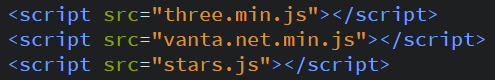
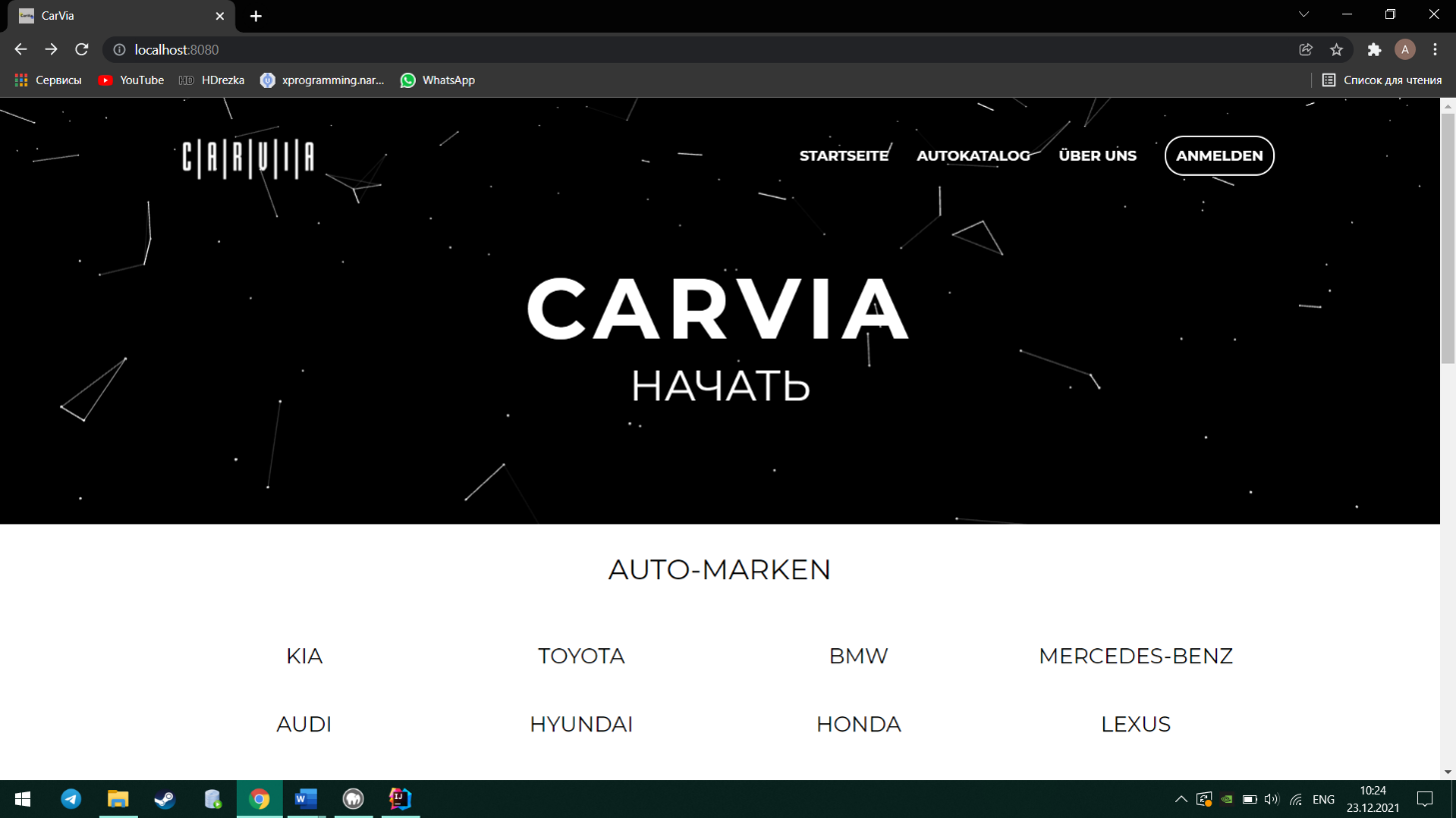


Abbildung 11. Hinzufügen von JS zu einer HTML-Seite.

Um HEADER im Vordergrund und einen animierten Hintergrund im Hintergrund zu erstellen, wird die POSITION-Eigenschaft mit dem RELATIVE-Wert in der CSS-Datei für den CONTAINER-Selektor angegeben.

 Abbildung 12. Site Header mit JS gemacht.

Um ein gutes Aussehen zu schaffen, wurde eine CSS-Datei beigefügt, in der verschiedene Eigenschaften für die Selektoren geschrieben sind.

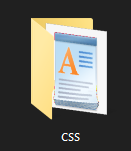
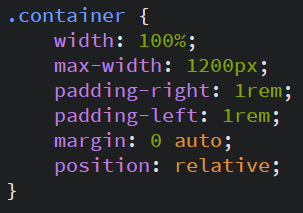
 

Abbildung 13. Stil für das Haupt div.

Die gesamte „index.html“ ist auf zentrierte Ausrichtung, eine Monserrat Sans Serif-Schriftart, eine Vordergrundfarbe von #2B2A2C und eine Schriftgröße von 15 Pixel (1 rem) eingestellt.

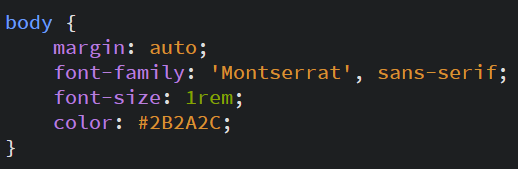


Abbildung 14. Stil für body.

Für Container wird das DIV-Tag verwendet, für Links das A-Tag, für den Seitenanfang das HEADER-Tag, für den Text das SPAN-Tag, für die Fotos das IMG-Tag, für den Seitenende FOOTER.







Abbildung 15. Tags und Klassen zum Erstellen eines Auto-Posts.

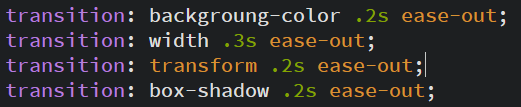






Abbildung 16. Tags und Klassen zum Erstellen von Website-Caps und -Kellern.

Um Text zu unterstreichen und zu vergrößern, Schatten für Fotos zu erzeugen, wird die HOVER-Pseudoklasse verwendet, die ausgelöst wird, wenn Sie den Mauszeiger über ein Element bewegen. Für eine flüssige Animation wird die Eigenschaft TRANSITION verwendet, bei der der Wert durch die Eigenschaft festgelegt wird, die auftreten soll, wenn der Mauszeiger schwebt: HINTERGRUND-FARBE - ändert den Hintergrund, WIDTH - unterstreicht den Text, TRANSFORM - erhöht das Element um Prozent , BOX-SHADOW - verdunkelt das Element. Der Zeitpunkt, ab dem die Änderung erfolgt, wird ebenfalls festgelegt.



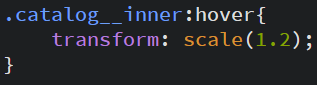
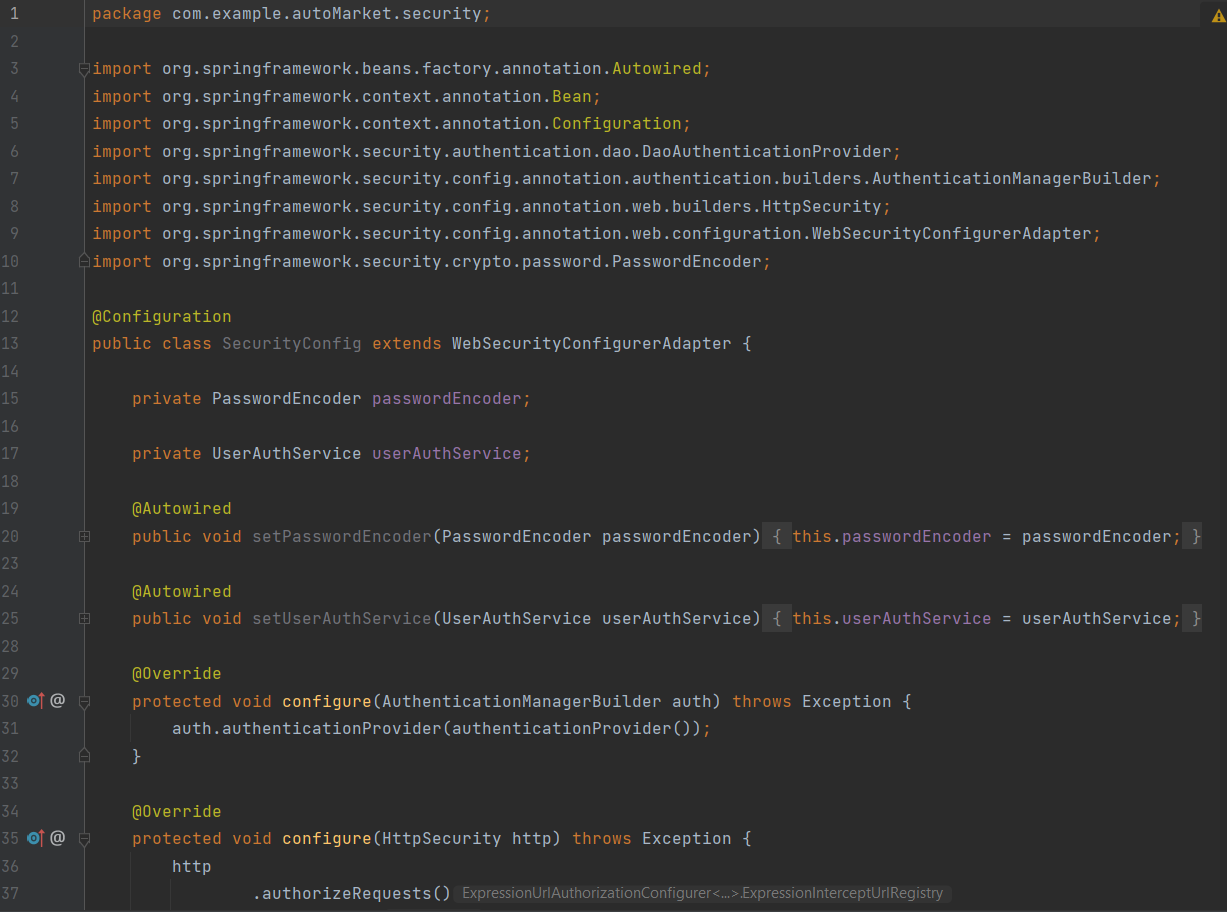
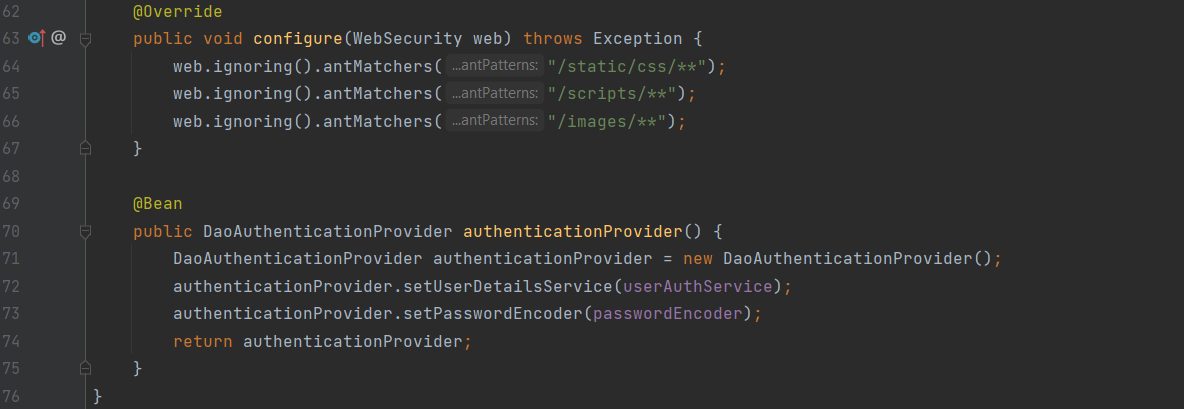
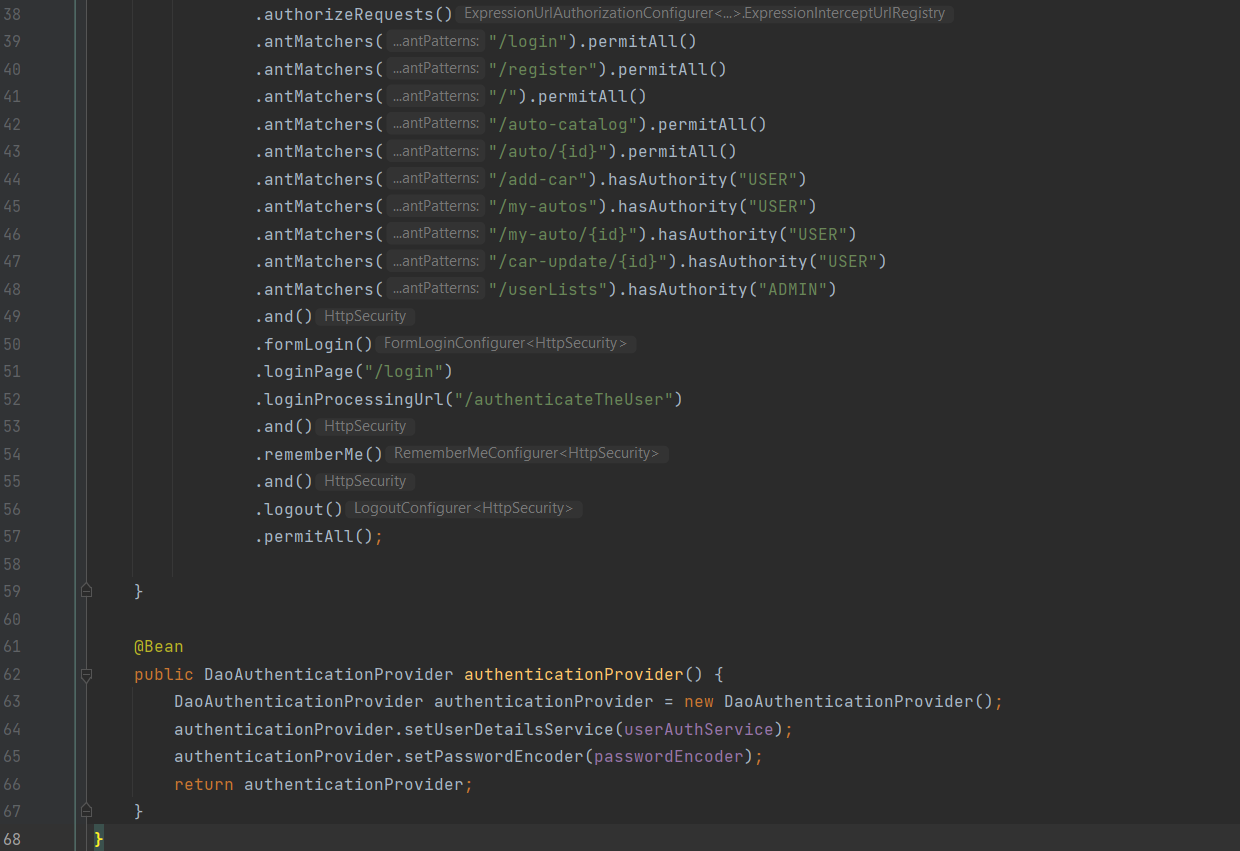


Abbildung 17. Animation für die Homepage.

## **Spring Security**

 Abbildung 18. Konfigurationen für Spring Security.

Paketklassen werden zuerst importiert

Dann kommt die Klasse mit dem Namen “SecurityConfig”, die von “WebSecurityConfigurerAdapter“ erbt.

Klasseninhalt:

* Zeile 15 – Klassenfeld “PasswordEncoder passwordEncoder”, dass um Passwörter zu  
  verschlüsseln verwendet
* Zeile 17 – Klassenfeld “UserAuthService“, dass um alle Autorisierungsanfragen zu bearbeiten verwendet
* Zeile 20-25 – Klassenfeld einstellen
* Zeile 30-32 – Methode, die Autorisierung bereitstellt
* Zeile 35-59 – Methode, die Seiten verwaltet, Seitenanfragen stellt, den Zugriff auf Seiten einer Website gewährt oder verweigert
* Zeile 63-67 – Methode, Methode, die Stile, Bilder, Skripte enthält
* Zeile 70-75 – Methode zur Autorisierung des Benutzers (Login mit Nutzername und Passwort)

## **Authentifizierungssystem**

Abbildung 19. Der Autorisierungscode der Website.

Paketklassen werden zuerst importiert

Dann kommt eine Klasse namens "UserAuthService", die die Schnittstelle "WebSecurityConfigurerAdapter" implementiert. Klasseninhalt:

* Zeile 17 – Klassenfeld “UserRepository repository”, dass Feld - Benutzer-Repository ist
* Zeile 20 – Das ist Konstrukteur, und in diesem Konstruktor setzen wir den Wert für das Klassenfeld “repository“
* Zeile 25-45 – Methode, die die Daten des autorisierten Benutzers zurückgibt

## **Beschreibung der Controller**

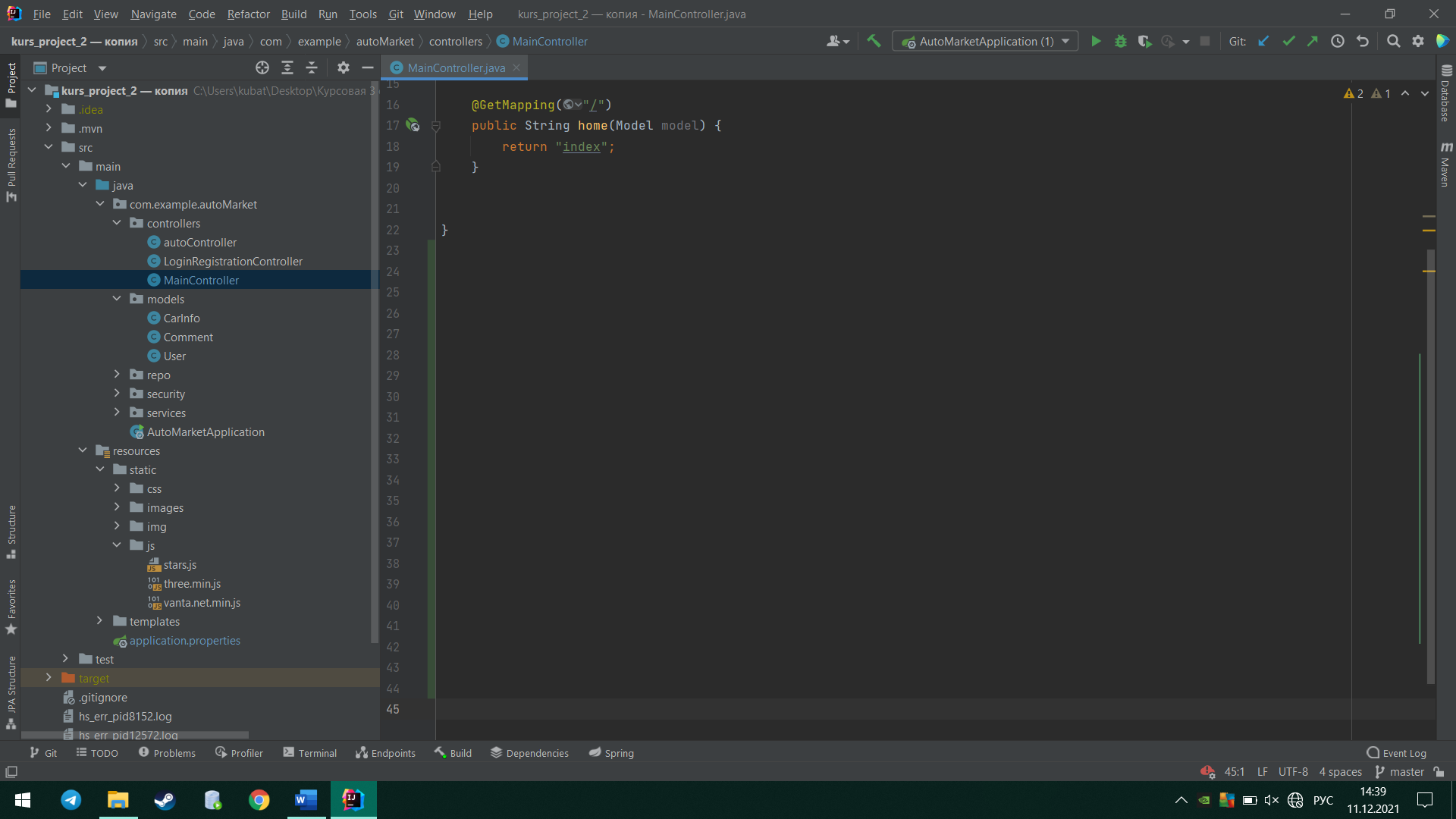


Abbildung 20. Code zum Anzeigen der Startseite.

Im Hauptcontroller haben wir eine Methode hinzugefügt, um nur eine URL-Adresse (/) zu erkennen - die Hauptseite. Diese Methode gibt HTML-Code von der Hauptseite (index.html) zurück.

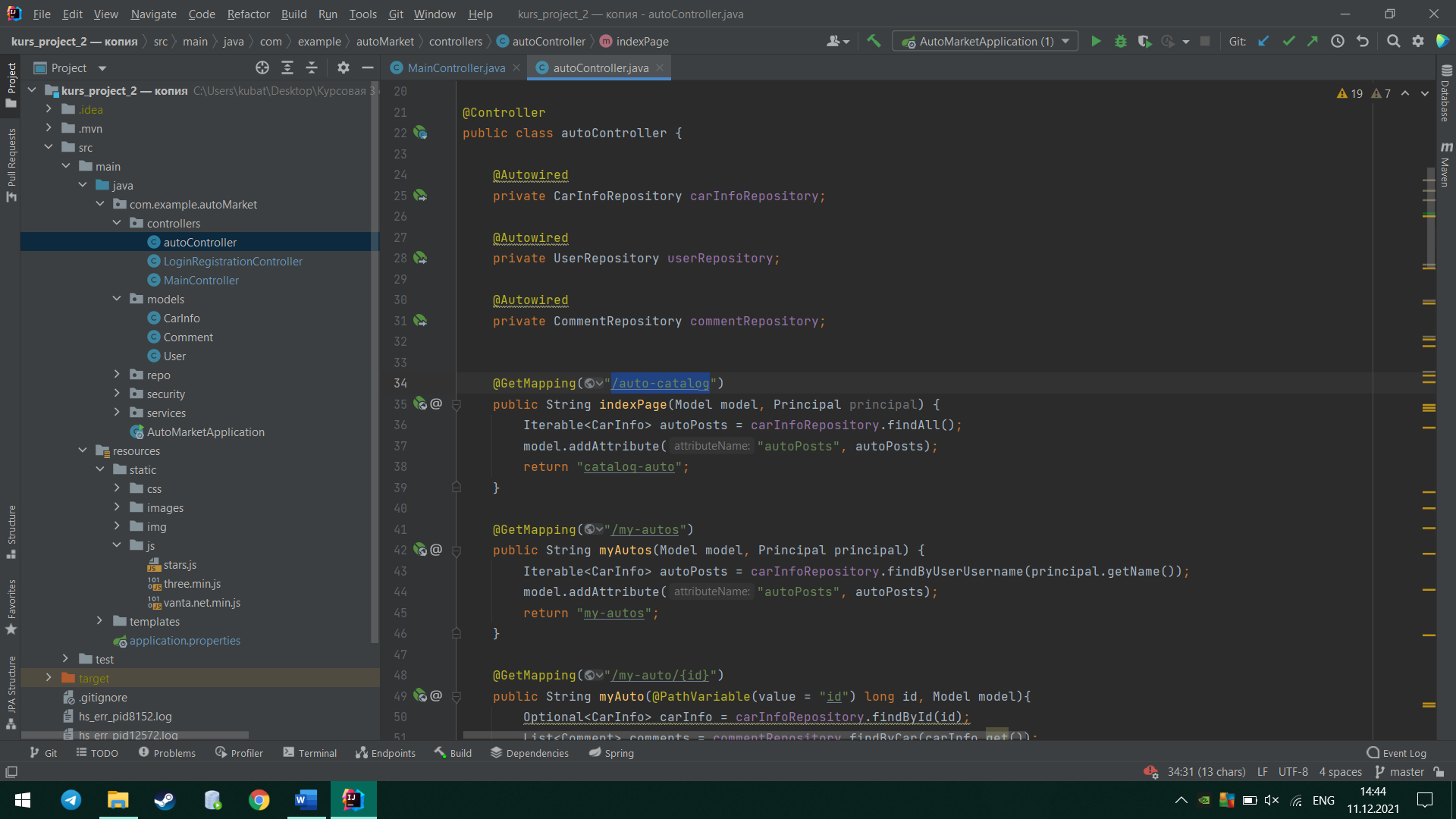


Abbildung 21. Repositories, die im autoController verwendet werden.

In autoController verwenden wir alle Tabellen aus der Datenbank, also erstellen wir Variablen für alle Repositorys, um die Tabellen zu verwenden.

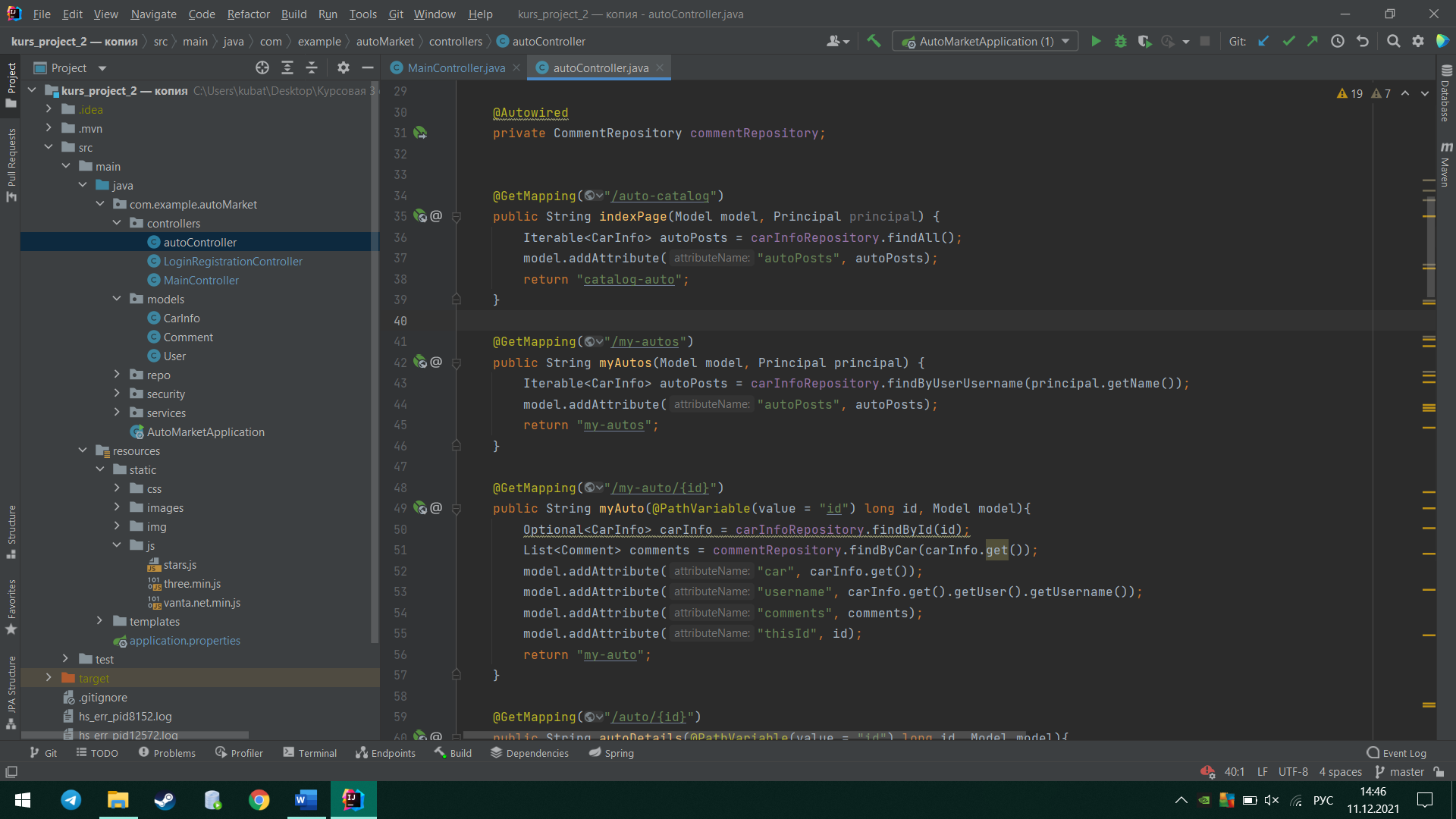
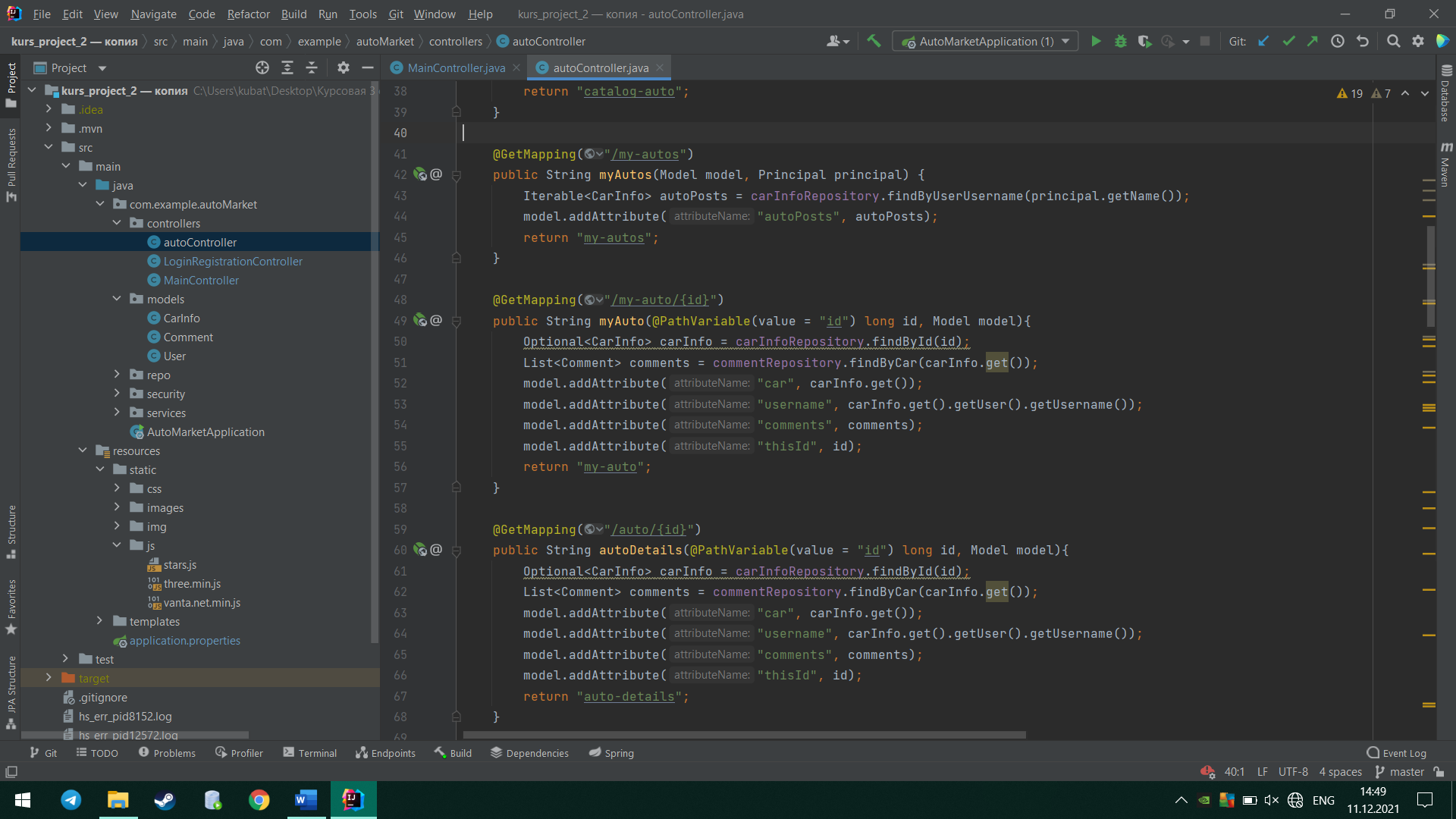


Abbildung 22. Code zum Anzeigen der Auto-Katalogseite.

Die URL-Adresse (/auto-catalog) gehört zur Seite „Auto-Katalog“ und wird mit der Methode indexPage verarbeitet und übergibt die Liste an die HTML-Seite catalog-auto.

 Abbildung 23. Der Code zum Anzeigen der Seite "Meine Autos".

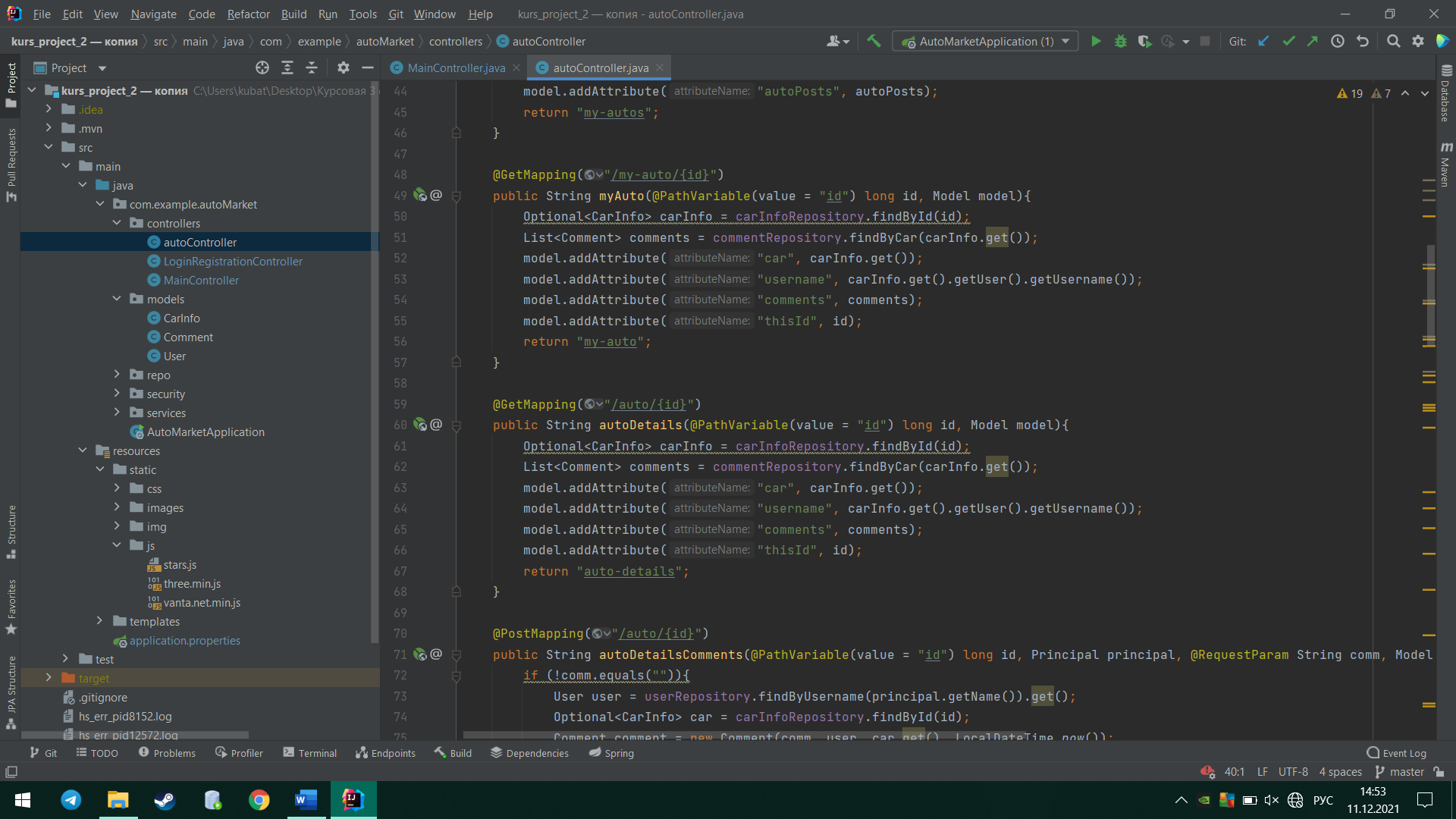
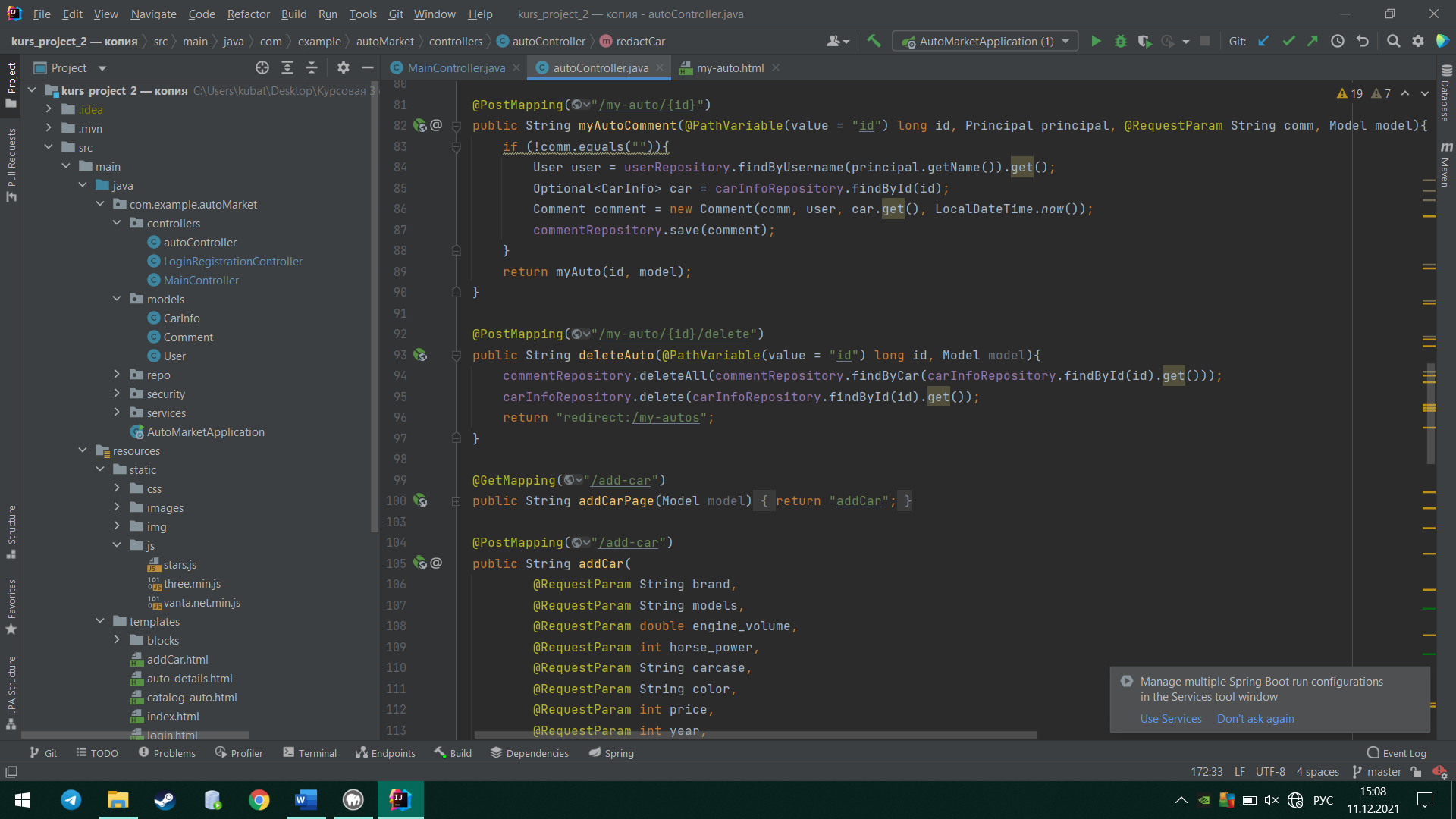
Die URL-Adresse (/my-autos) gehört zur Seite „Meine Autos“ und wird mit der Methode myAutos verarbeitet und funktioniert genauso wie die Methode indexPage, außer dass sie anstelle einer Liste aller Autos die Liste der Autos von autorisierten Benutzern. Diese Seite funktioniert nur nach Benutzerautorisierung.

Abbildung 24. Code zum Anzeigen der persönlichen Seite des Autors.

Die URL-Adresse (/auto/{id}) gehört zur Auto-Post-Seite, wird mit der Methode autoDetails verarbeitet und übermittelt Daten zum Auto, zum Nutzer, der das Auto veröffentlicht hat, Auto-ID und eine Liste mit Kommentaren, die unter dem Post hinterlassen wurden.

Die URL-Adresse (/ my-auto /{id}) der Beitragsseite meines Autos wird mit der myAuto-Methode verarbeitet und funktioniert genauso wie die vorherige Methode. Der einzige Unterschied besteht darin, dass die gesendete HTML-Seite Schaltflächen zum Löschen und Bearbeiten enthält.

Abbildung 25. HTML-Code für die Schaltflächen "Bearbeiten" und "Löschen".

 Abbildung 26. Methode zum Entfernen von Auto.

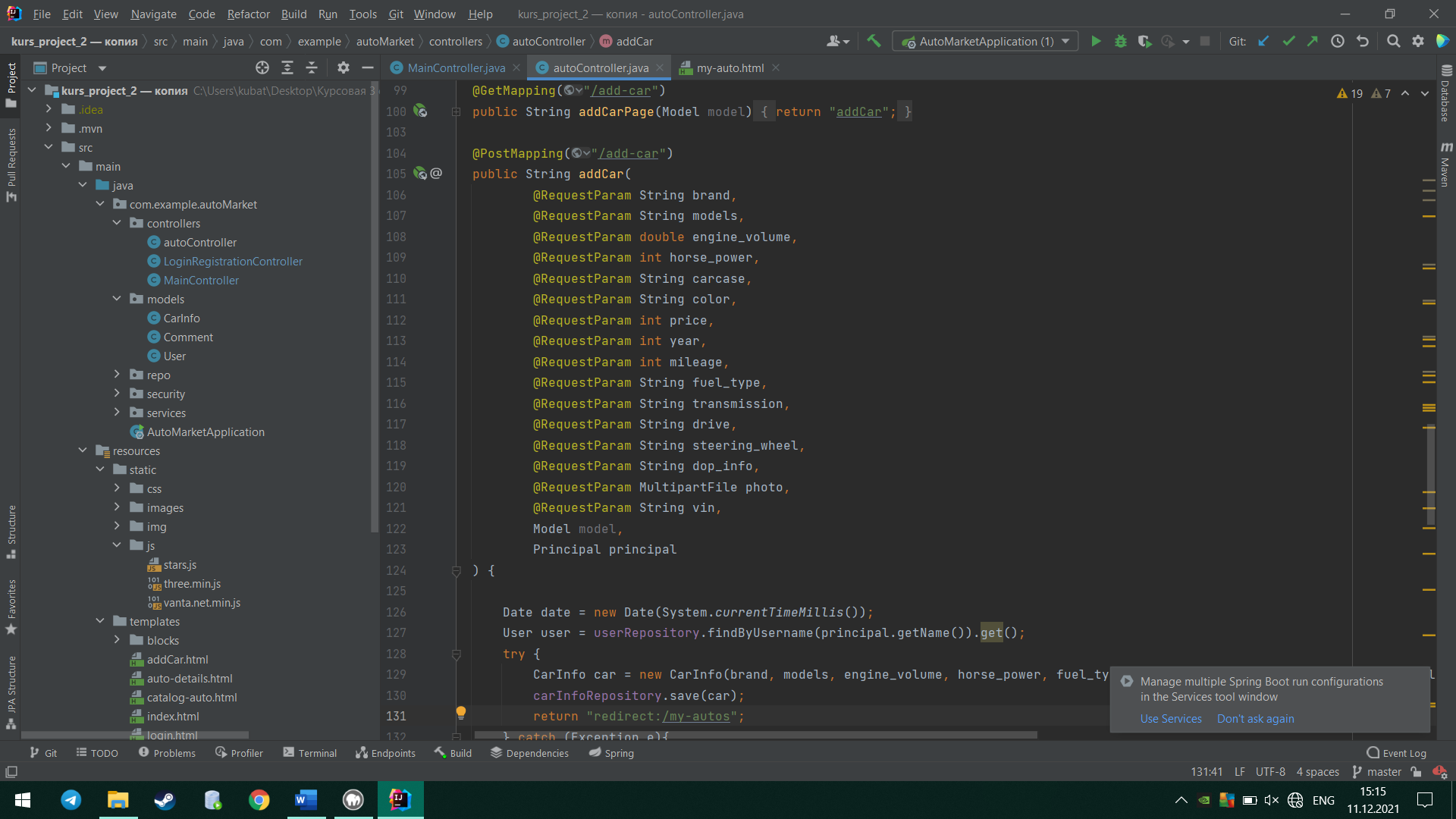
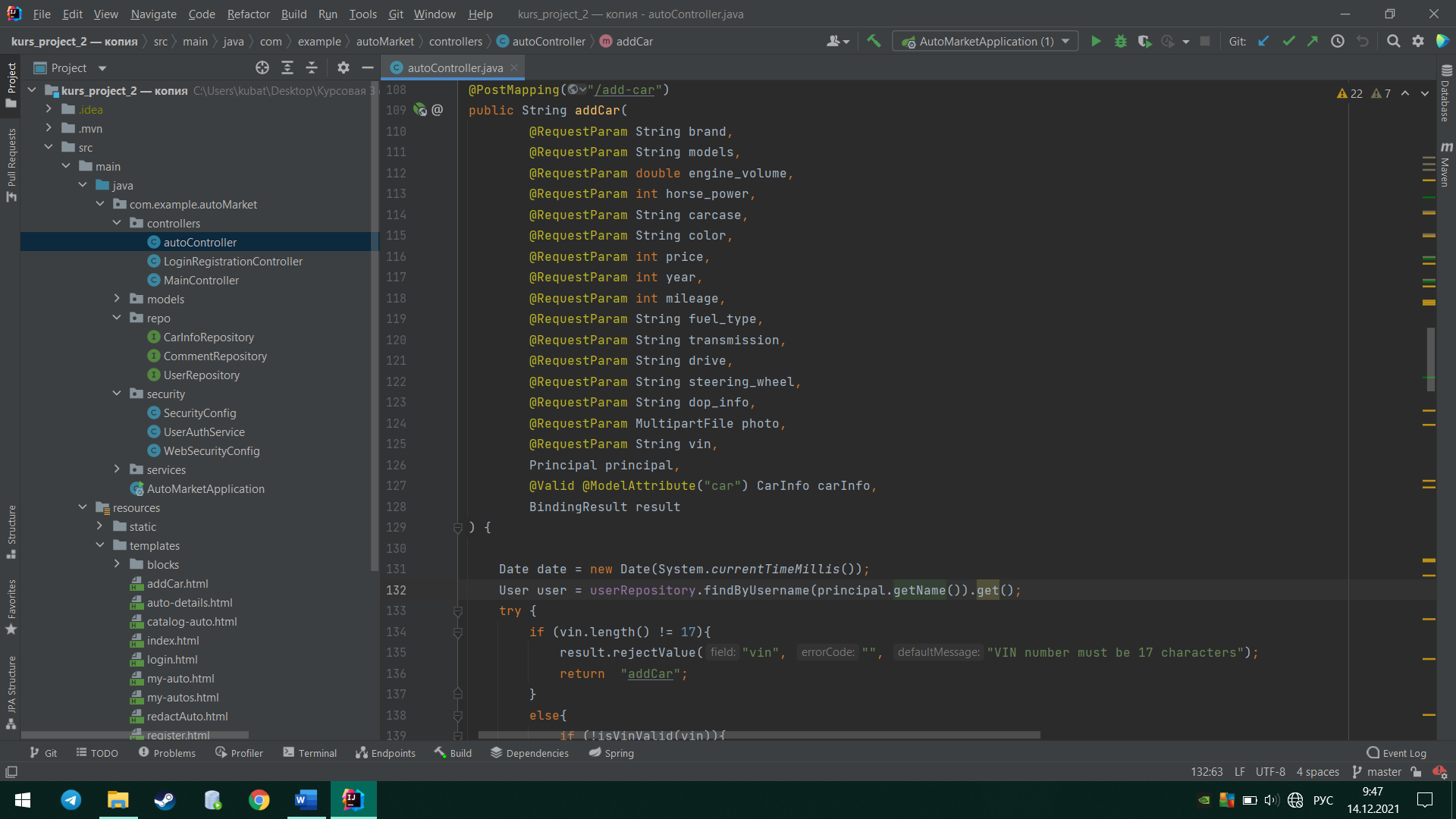
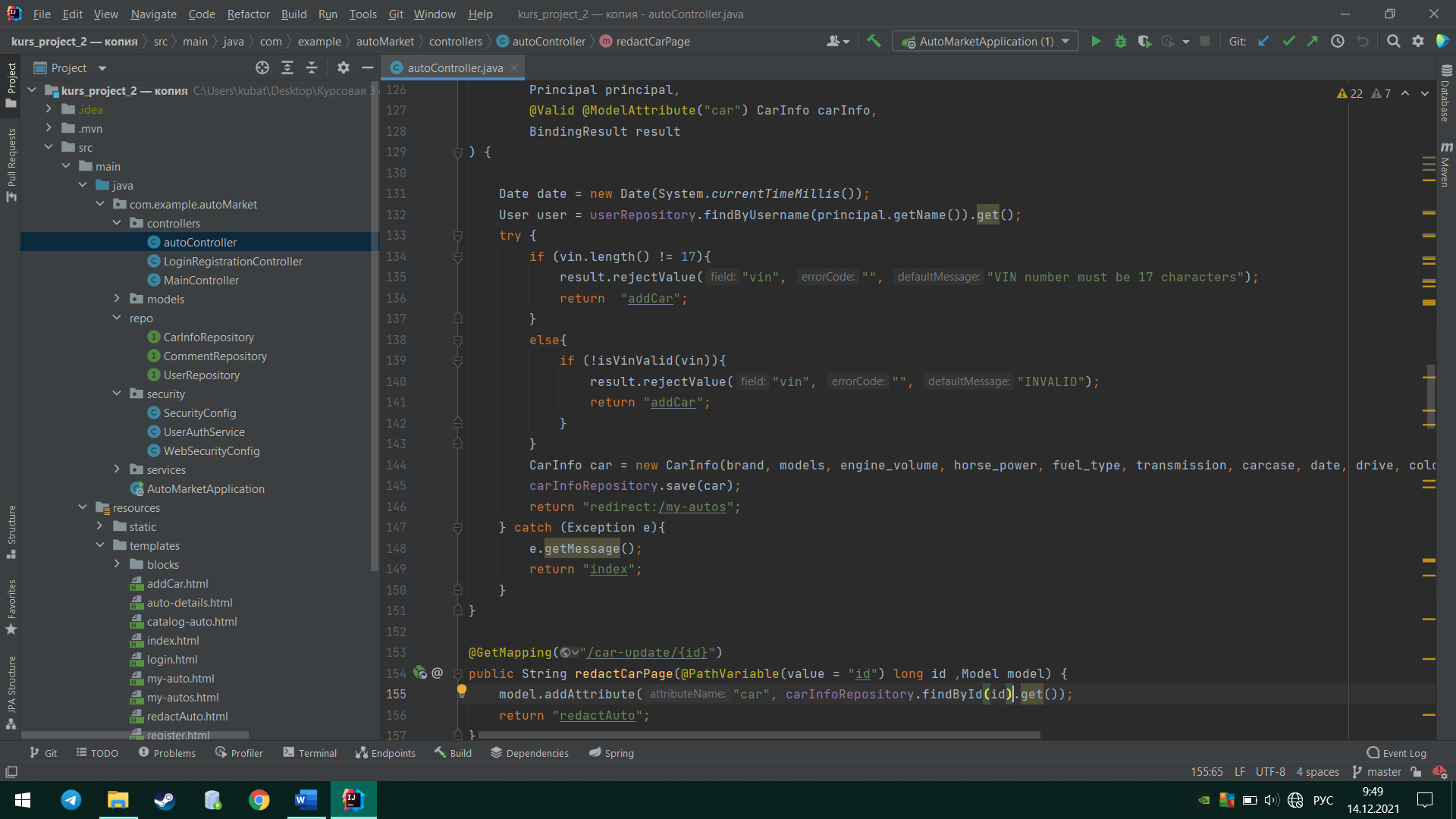
Das Löschen eines Autos erfolgt, wenn der Benutzer beim Drücken der Löschtaste an die URL-Adresse (/my-auto/{id}/delete) übergeben wird, die von der Methode deleteAuto verarbeitet wird. Diese Methode findet das Auto anhand der in der URL angegebenen ID und entfernt es. Dann leitet es den Benutzer an die URL-Adresse (/my-autos) weiter.

Abbildung 27. Methode zum Anzeigen der Seite Auto hinzufügen.

Die URL-Adresse (/add-car) gehört zur Seite „Auto hinzufügen“ und behandelt die Methode addCarPage, die die HTML-Seite an addCar.html übergibt.

 Abbildung 28. Methode zum Hinzufügen von Auto.

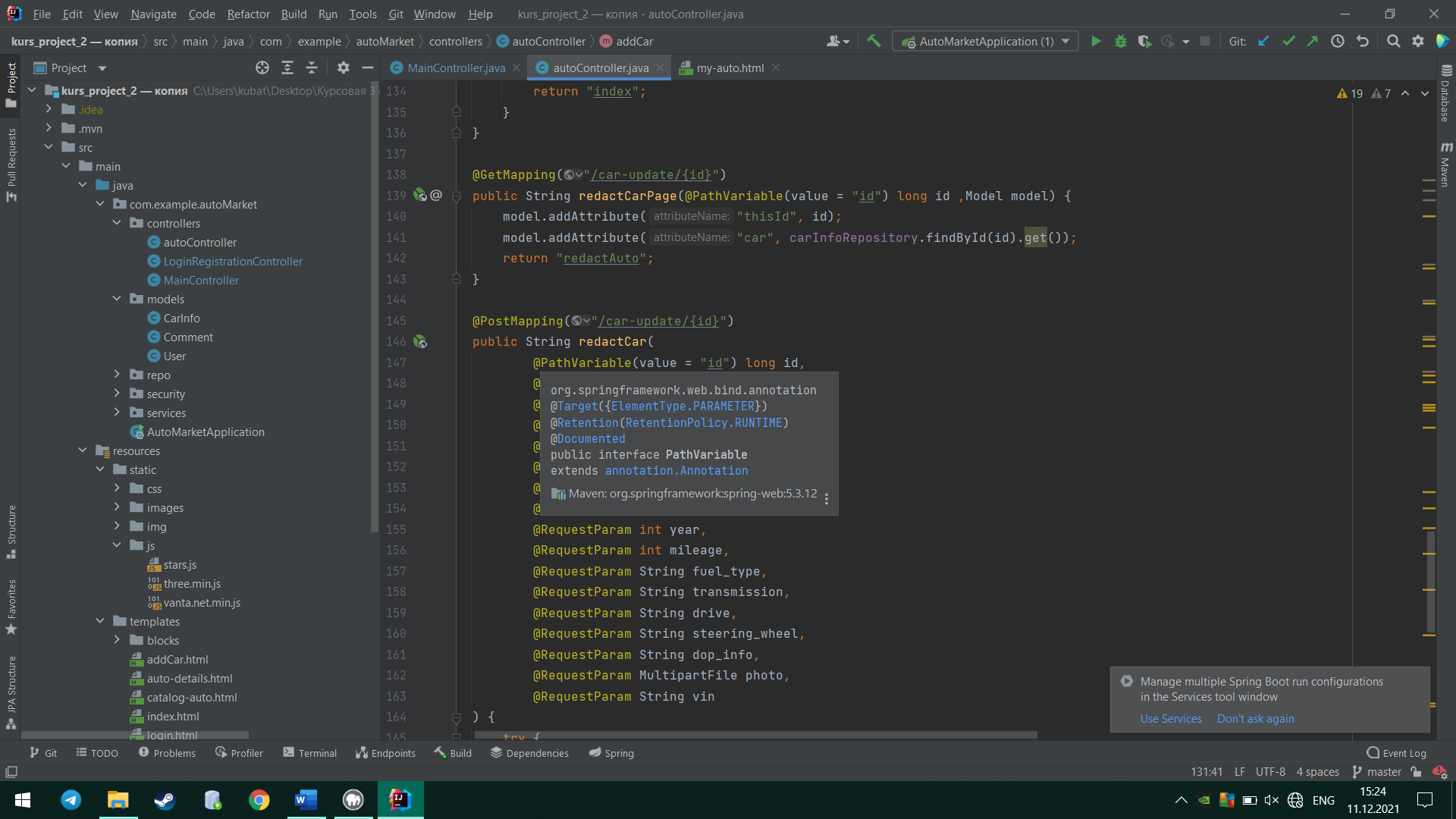
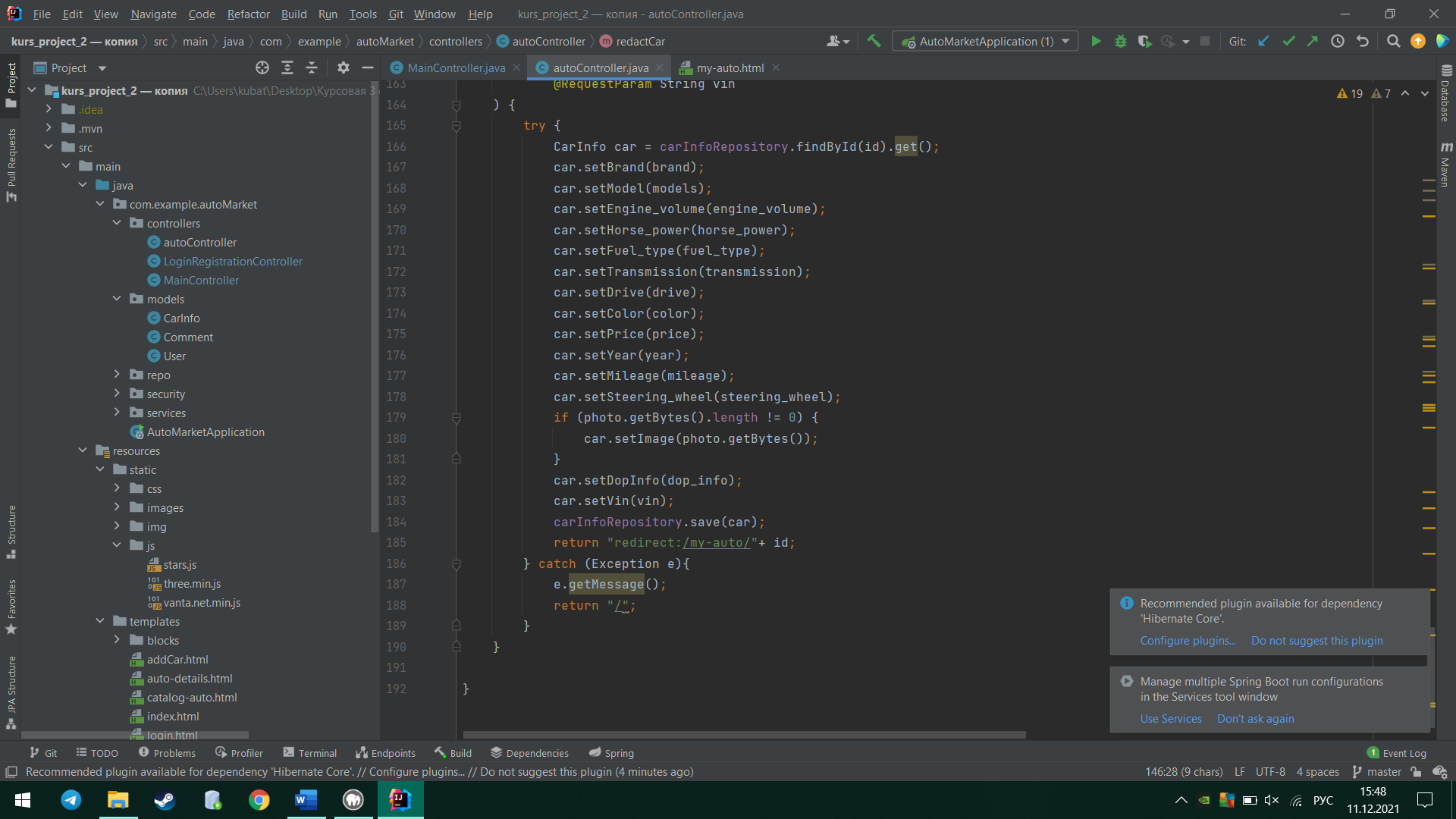
Alle von der URL-Adresse (/add-car) eingegebenen Daten werden von der addCar-Methode verarbeitet. Es akzeptiert eingegebene Werte mit dem Parameter @RequestParam. Erstellt eine CarInfo-Klasse mit den eingegebenen Daten und speichert sie in der Tabelle cars. Auch in der Methode gibt es eine Überprüfung des Weincodes des Autos. Nach Abschluss leitet die Methode den Benutzer zur URL (/my-autos) weiter.

Abbildung 29. Zeigt die Seite zum Bearbeiten von Auto an.

Die URL-Adresse (/car-update/{id}) wird mit der redactCarPage-Methode verarbeitet. Das überträgt die ID des Autos und die Daten über das Auto auf die HTML-Seite. Die Methode gibt die HTML-Seite redactAuto.html zurück.

Die redactCar-Methode zum Verarbeiten von Daten an der URL-Adresse (/car-update/{id}) akzeptiert dieselben Daten wie die addCar-Methode. Aber sie haben einen anderen Methodenkörper. Die Methode addCar hat eine neue Klasse erstellt und in der Datenbank gespeichert, und diese Methode holt die bereits erstellte Klasse aus der Datenbank, aktualisiert die Daten mit Settern und speichert diese Klasse. Somit werden die Daten aktualisiert. Nach Abschluss des Verfahrens wird der Nutzer auf die Seite des bereits bearbeiteten Autos weitergeleitet.



## 

Abbildung 30. Methode zum Bearbeiten von Auto.

## **Admin**

Der Admin-Controller enthält alle Modell-Repositories, um mit allen Tabellen zu interagieren.

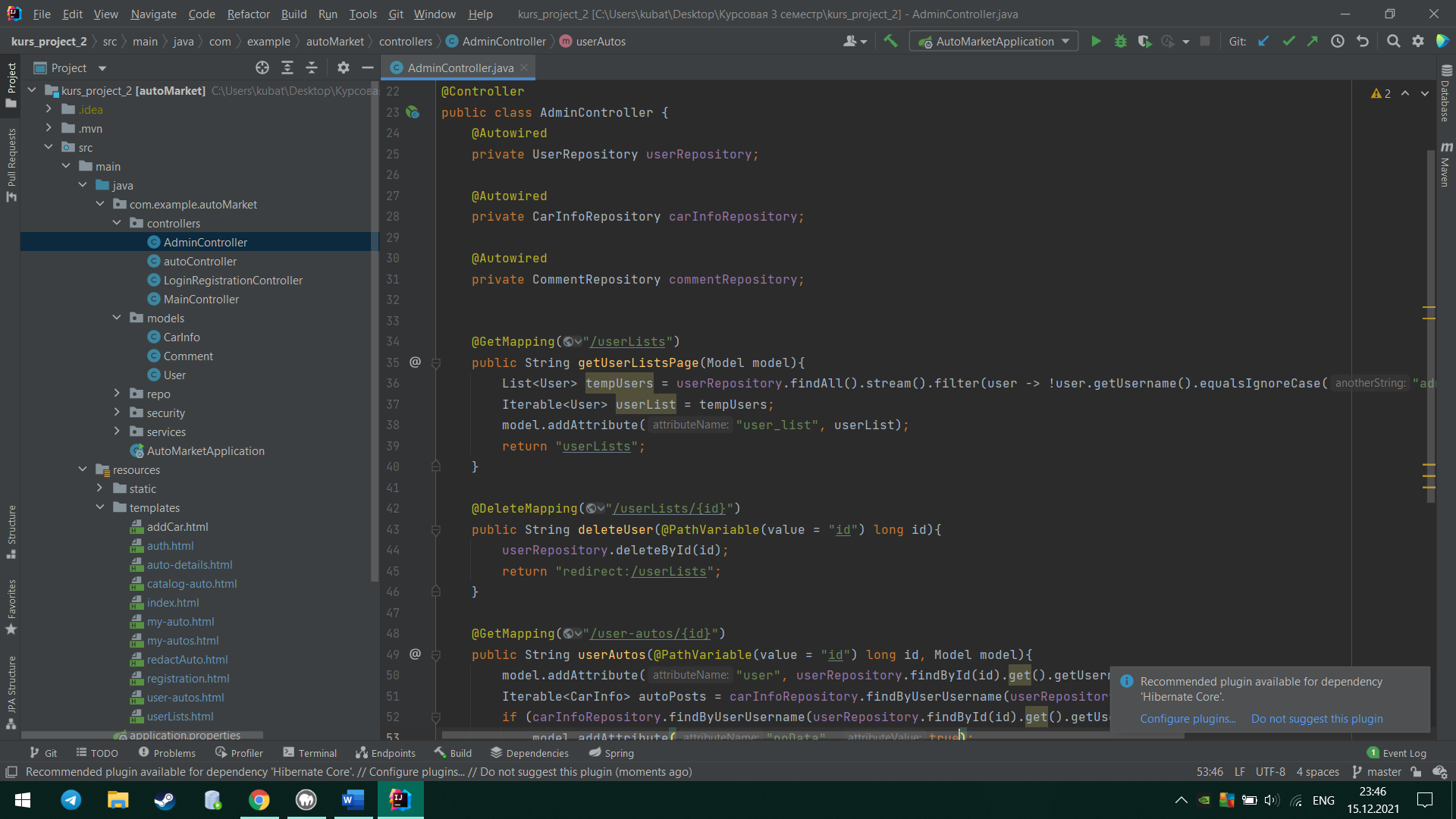


Abbildung 31. Repositories, die in AdminController verwendet werden.

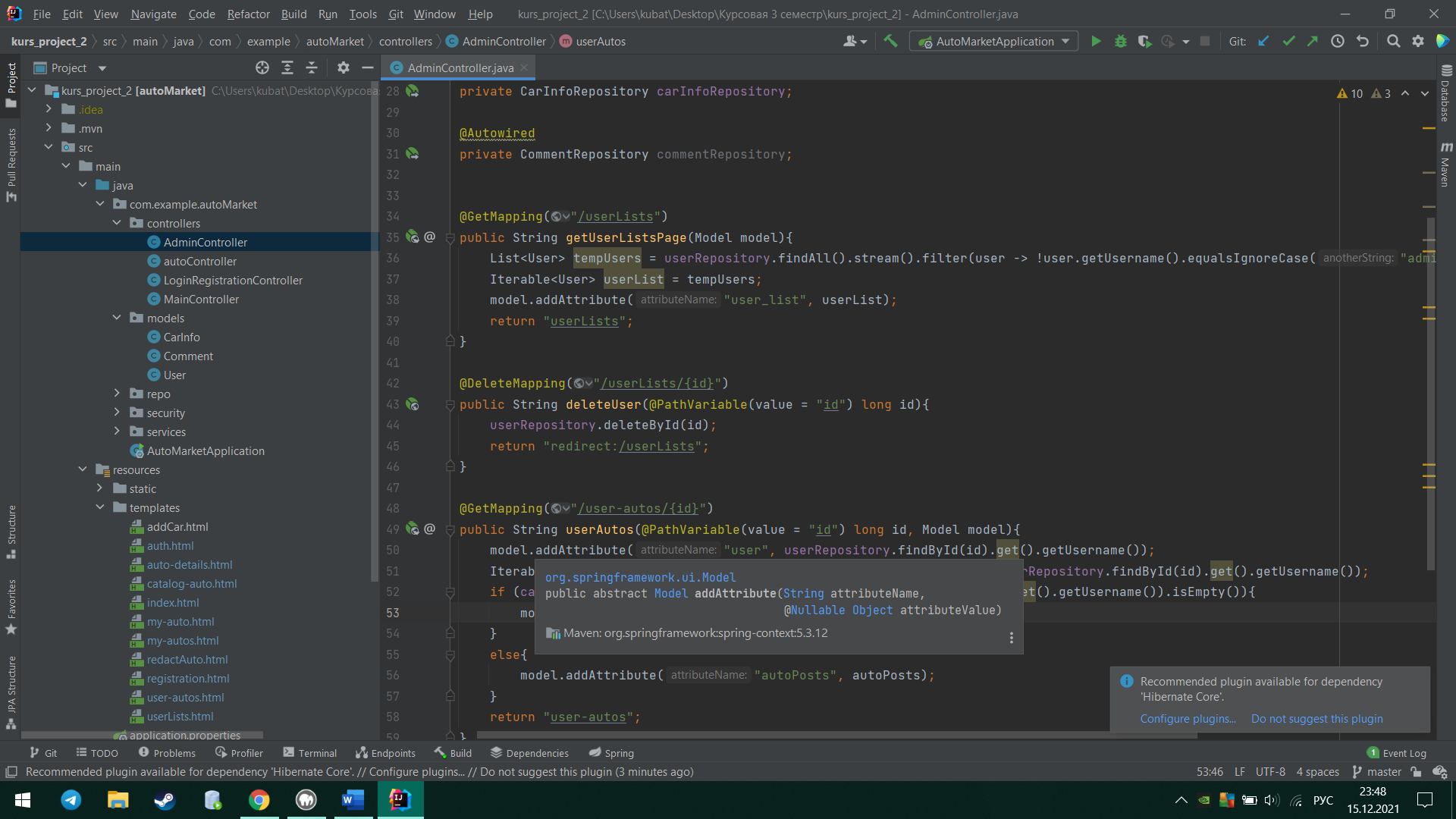
Url адрес (/userLists) обрабатывается методом getUserListsPage. Метод передает список всех пользователей и возвращает HTML страницу userLists.

Abbildung 32. Zeigt eine Liste von Benutzern an.

Die URL (/userLists) wird von der getUserListsPage-Methode verarbeitet. Die Methode übergibt eine Liste aller Benutzer und gibt die HTML-Seite userLists zurück.

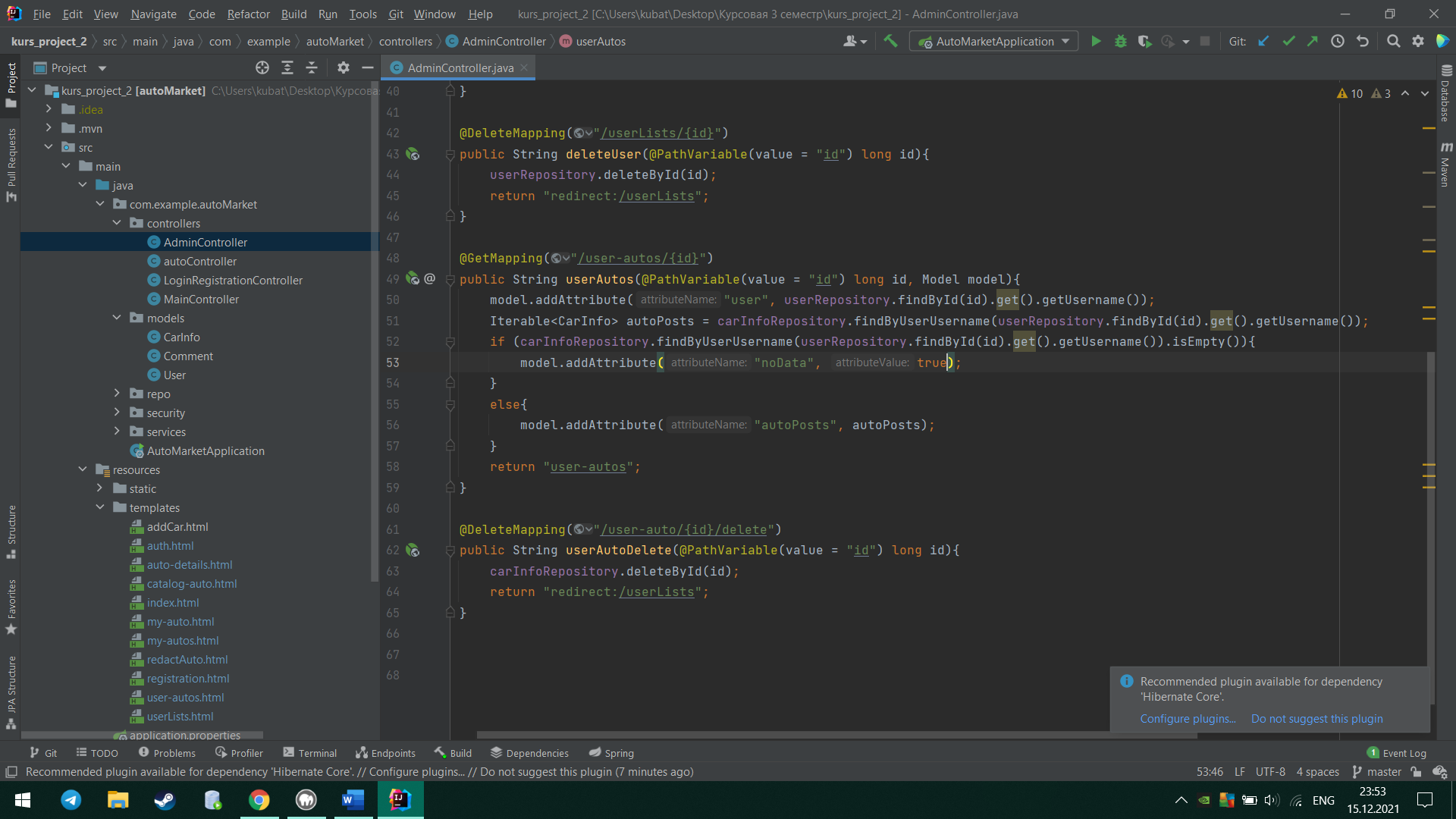
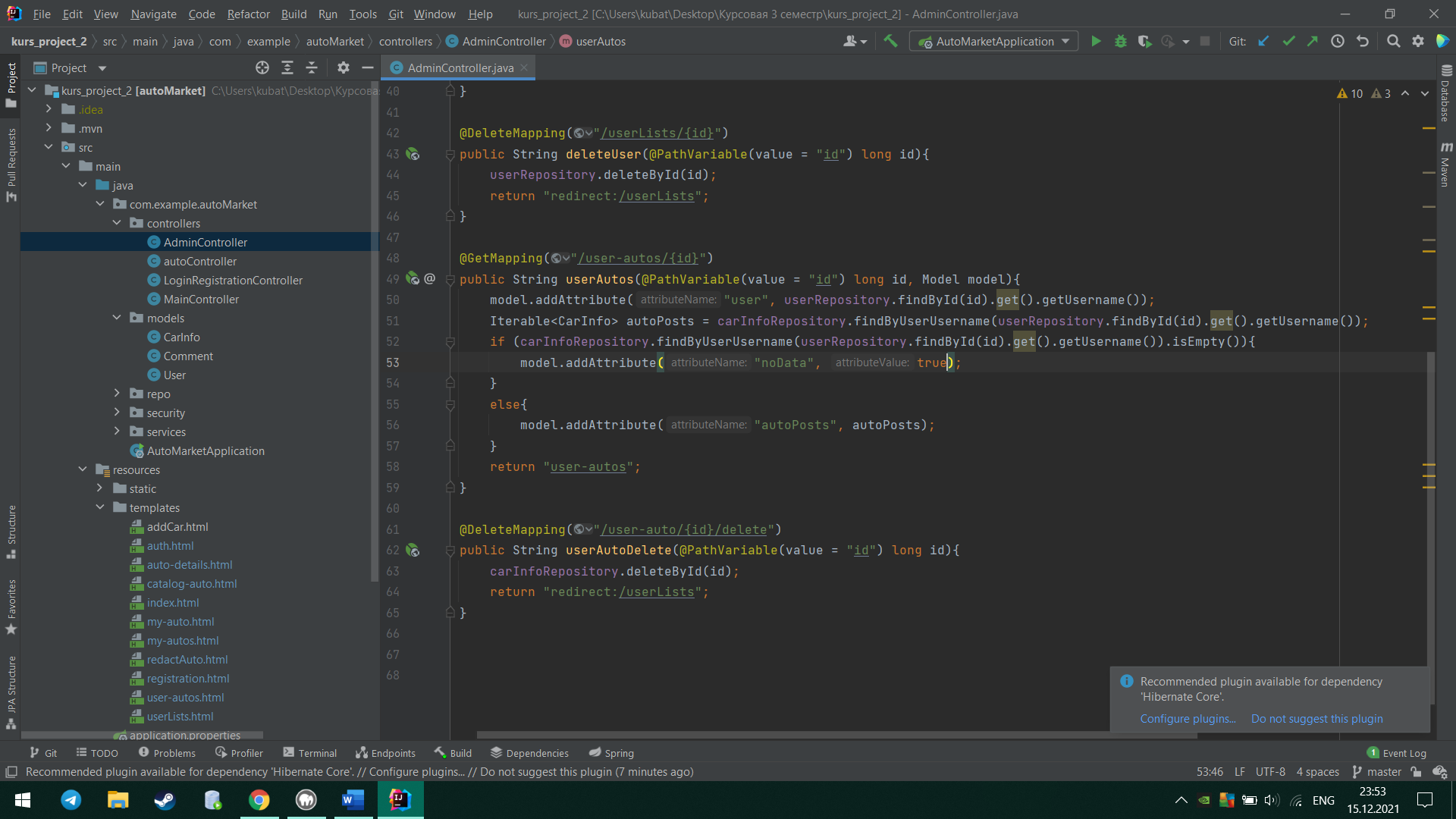


Abbildung 33. Methode zum Löschen des Benutzers.

Die Liste der Fahrzeuge des Benutzers wird unter der URL (/user-auto/{id}) angezeigt und von der userAutos-Methode verarbeitet. Die Methode überträgt den Benutzernamen sowie eine Liste seiner Fahrzeuge. Wenn keine Autos vorhanden sind, wird auf dem Bildschirm angezeigt, dass keine Autos vorhanden sind.

 Abbildung 34. Anzeige der Fahrzeuge des Benutzers.

Das Löschen von Fahrzeugen über den Admin erfolgt über die URL (/user-auto/{id}/delete) und wird von der userAutoDelete-Methode verarbeitet. Die Methode entfernt das Fahrzeug nach ID aus der URL der Adresse.

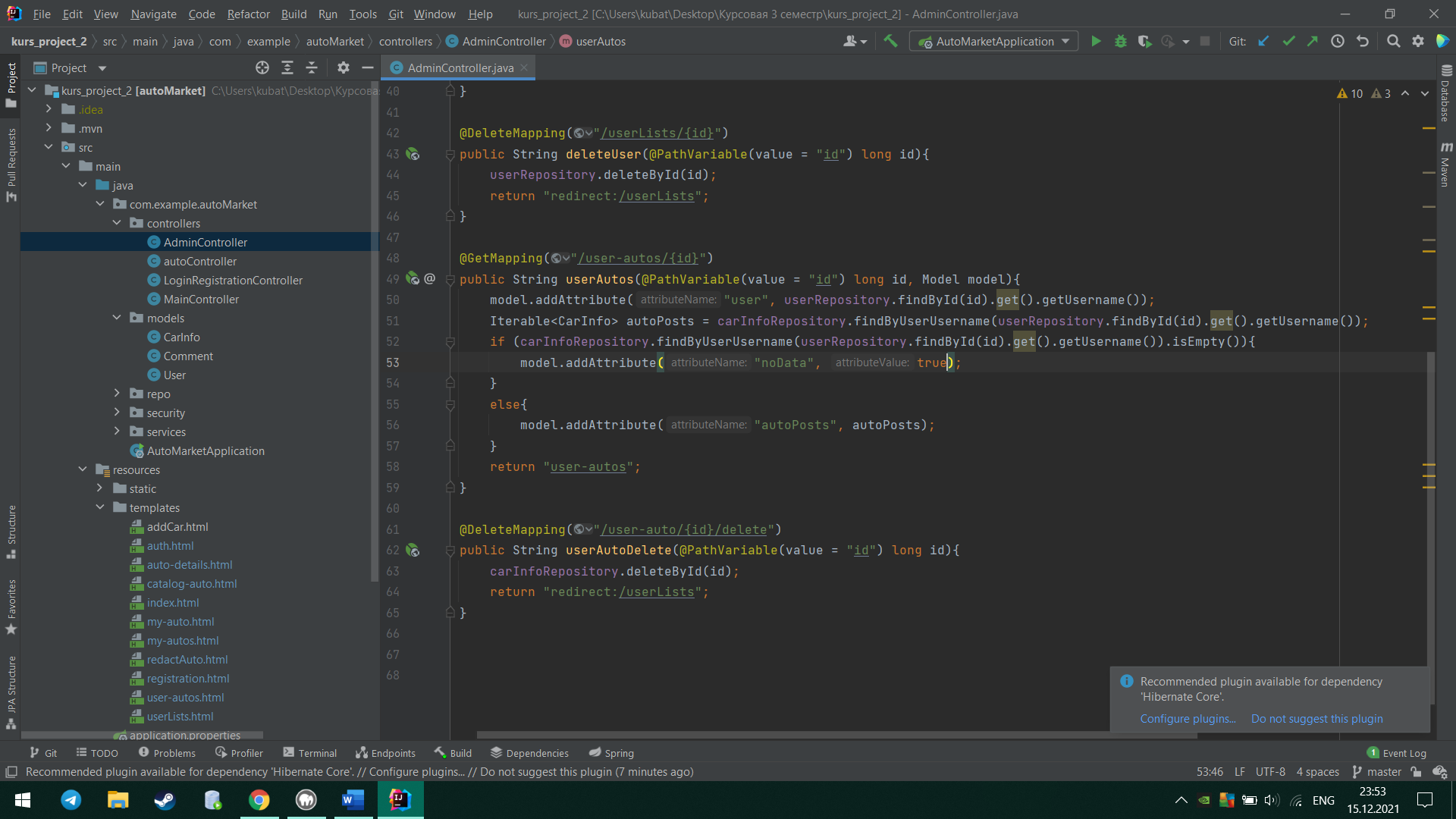


Abbildung 35. Entfernen Sie das Fahrzeug des Benutzers.

# **Fazit**

## **Ergebnisse**

Zusammenfassend möchte ich sagen, dass es uns gelungen ist, die gestellten Aufgaben zu lösen. Das Ergebnis unserer Kursarbeit zum Thema „AIS Automatisierung der Kfz-Verkaufsabrechnung“ sind folgende Ergebnisse:

1. Für unser Produkt wurde eine Marktanalyse durchgeführt
2. Der Aufbau der Datenbank für die Webanwendung wurde durchdacht
3. Das Layout für die Seite wurde erstellt
4. Code wurde für die Registrierung und Autorisierung mit Spring Security geschrieben
5. Der Code der Webanwendung wurde für das Layout geschrieben
6. Admin-Konto wurde hinzugefügt

Außerdem hat unser Team während des Schreibens der Hausarbeit seine Kenntnisse in der Java-Programmierung und der Erstellung von HTML-Seiten mit CSS-Stilen verbessert. Jeder Teilnehmer hat dazu beigetragen. Wir haben als Team gearbeitet. Dabei haben wir uns auch mit GitHub sowie dem SpringBoot-Webframework beschäftigt. Wir haben gute Erfahrungen in der Teamarbeit gesammelt. Es war manchmal schwierig, aber wir konnten es rechtzeitig bewältigen.

## **Spezielle Anforderungen**

Diese Webanwendung wurde mit den folgenden Tools erstellt:

1. Programmiersprache: Java
2. Framework: Spring Boot
3. IDE: Jetbrains intellij IDEA 2021
4. Datenbank: MySQL

# **Literatur**

* Thymeleaf
  + <https://www.baeldung.com/tag/thymeleaf/>
  + <https://stackoverflow.com/questions/42906027/spring-security-expressions-in-thymeleaf>
* Spring Boot
  + <https://youtu.be/FyZFK4LBjj0>
  + <https://youtu.be/5s50Zj8YIE8>
  + <https://youtu.be/S1oOMJorZ2w>
  + <https://youtu.be/oGK2KufvxM0>
  + <https://youtu.be/sPH58lbzYvo>
  + <https://youtu.be/sPH58lbzYvo>
  + <https://youtu.be/R3YNPm7ZmYY>
  + <https://youtu.be/nCfqKi0NjXM>
  + <https://spring.io/projects/spring-boot>
  + <https://habr.com/ru/post/435144/>
* Spring Security
  + <https://youtu.be/7uxROJ1nduk>
  + <https://youtu.be/HvovW6Uh1yU>
  + <https://youtu.be/rBKxqQJFeHQ>
  + <https://spring.io/projects/spring-security>
  + <https://habr.com/ru/post/482552/>