Automobile maintenance tracker (ATM)

Description:

In this application, I have covered 3 modules (user, vehicle and maintenance). Each module contains functionality like (list, add, edit, view and delete) using react JS with some required validations.

Without login user can’t access any application functionality. User need to be logged in first to access all functionality. Also an application has validations like add unique user, required all fields, oil change is not applied on electric vehicles

**Note**:

* Please find source code on git (<https://github.com/hardikpatel1644/amt/> ).
* Please find postman sample\_request files here : /amt/sample\_requests

Tools and Technology Used:

* HTML
* jQuery , React JS
* CSS – Bootstrap
* PHP, MySql - XAMPP
* phpunit

Installation:

* Unzip "amt.zip" file into your server root directory.
* Create database "amt" and import "amt.sql" sample database file from (/amt/db/) directory.
* Now try to execute project from server. or use this link (http://localhost/amt or http://127.0.0.1/amt) or <http://yourserverip/amt>

Project directory structure:

- API: contains Server side API logic

- includes: contains configurations files

- config.php - basic configuration for project

- db.php - database connection class

- lib: contains common library or functions files

- functions.php - common functions used in project

- maintenance: contains api calls for maintenance module

- create.php - add data into table

- delete.php - delete data from table

- index.php - list data from table

- update.php - update data for table

- view.php - view single entry from table

- models:

- auth.php

- base\_model.php

- maintenance\_model.php

- user\_model.php

- vehicle\_model.php

- user:

- create.php - add data into table

- delete.php - delete data from table

- index.php - list data from table

- login.php - api call for check login for user

- logpot.php - api call for logout

- update.php - update data for table

- view.php - view single entry from table

- vehicle:

- create.php - add data into table

- delete.php - delete data from table

- index.php - list data from table

- update.php - update data for table

- view.php - view single entry from table

- Tests: contains test cases for application

- API: contains test cases for API

- includes: contains configurations files

- config.php - basic configuration for Test of project

- lib:

- function\_test.php - Test cases for functions.php

- models:

- auth\_test.php - test cases for auth.php class

- maintenance\_model\_test.php - test cases for maintenance\_model.php class

- user\_model\_test.php - test cases for user\_model.php class

- vehicle\_model\_test.php - test cases for vechicle\_model.php class

- assets: contains css and javascript files

- css :

- style.css

- js : contains javascript files for react components and classes

- maintenance: contains react components for maintenance API

- create\_maintenance.component.js

- delete\_maintenance.component.js

- maintenance\_table\_row.component.js

- maintenances\_table.component.js

- update\_maintenance.component.js

- view\_maintenance.component.js

- user: contains react components for user API

- create\_user.component.js

- delete\_user.component.js

- main.component.js

- update\_user.component.js

- user\_table\_row.component.js

- users.component.js

- users\_table.component.js

- view\_user.component.js

- vehicle: contains react components for vehicle API

- create\_vehicle.component.js :

- delete\_vehicle.component.js

- vehicle\_table\_row.component.js

- vehicles\_table.component.js

- update\_vehicle.component.js

- view\_vehicle.component.js

- login.component.js

- top\_actios.component.php

- db: contains sample database

- amt.sql

- amt\_test.sql

- documents: contains documents for projects

- readme.txt

- unit\_test\_readme.txt

- sample\_request : contains sample postman requests for API calls

- maintenance.postman\_collection.json

- user.postman\_collection.json

- vehicle.postman\_collection.json

- vendor: contains library of testcases

- composer.json: file for install phpunit test case library / vendors

- composer.lock

- dashboard.html: file for dashboard of application

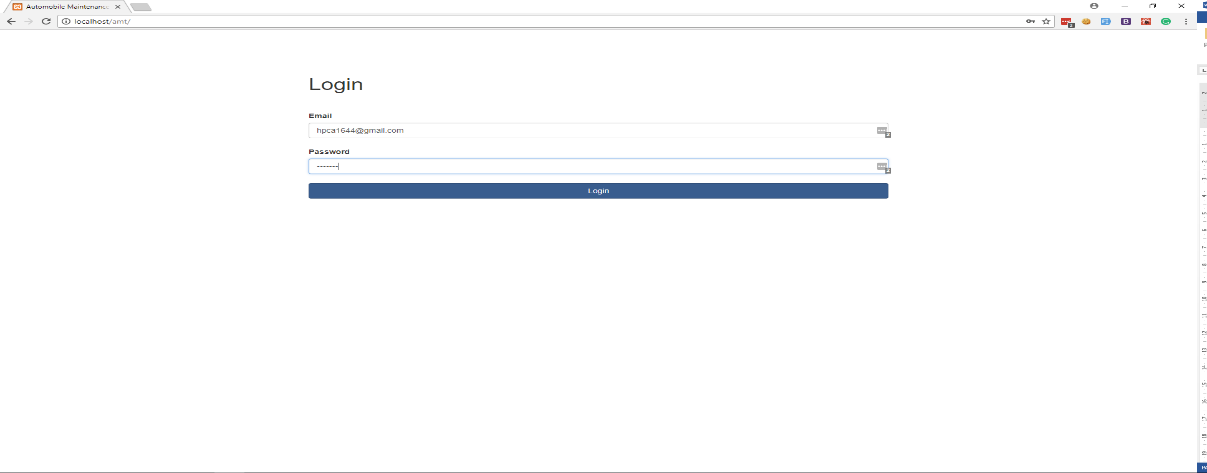
- index.html: file for main project execution file

- users.html: file for list all project users

- vehicles.html: file for list all vehicles and their maintenances

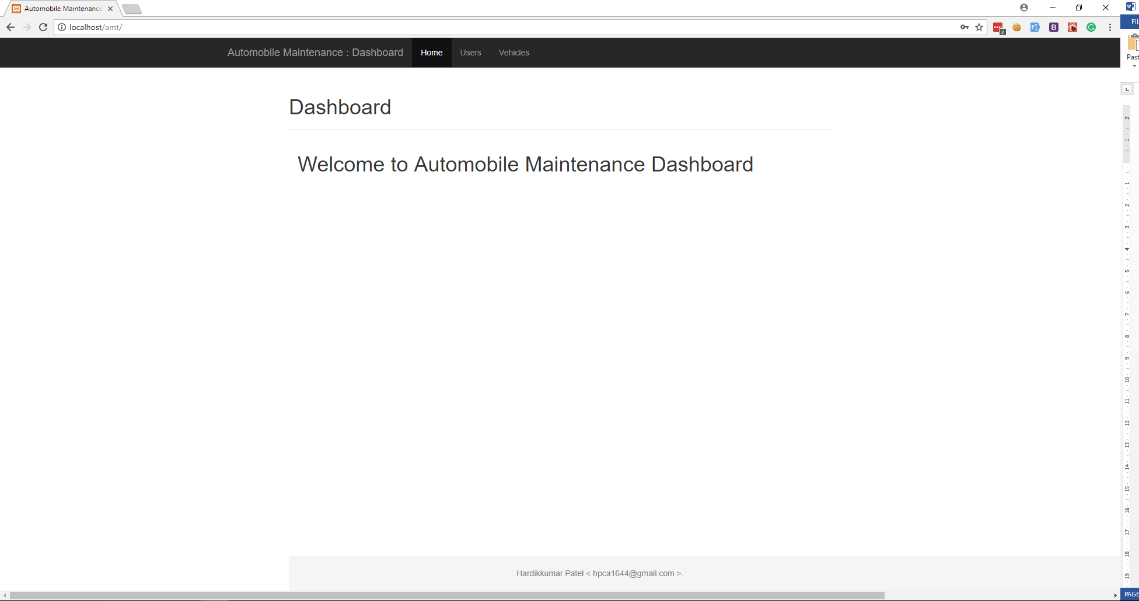
How to use Application:

1. Open application base url in browser (<http://localhost/amt/>).



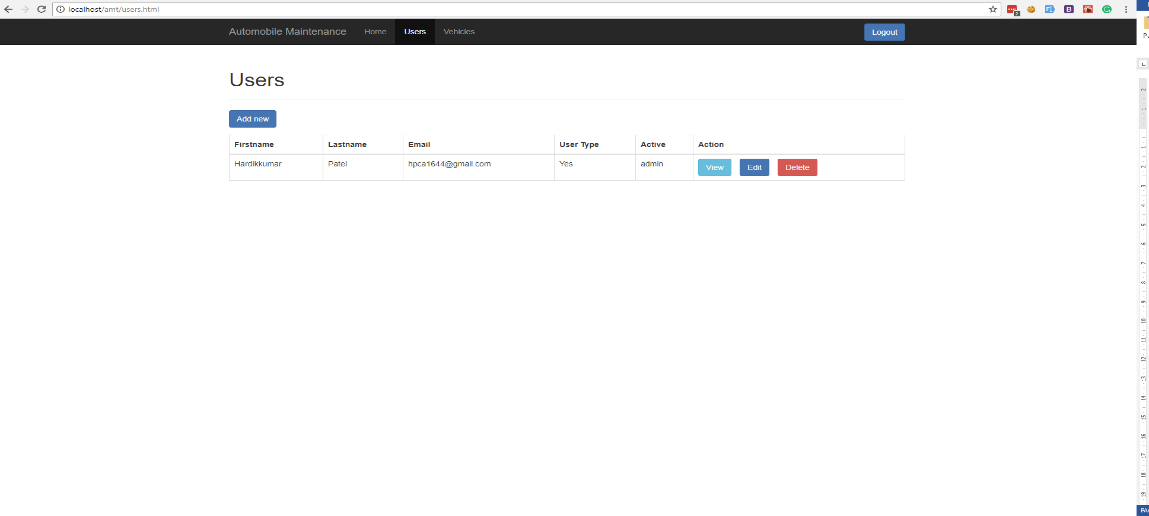
1. Login with valid username / password ( [hpca1644@gmail.com](mailto:hpca1644@gmail.com) / test123) which is already created in database.
2. Now End user can view dashboard of application. End user can view 3 links on top bar.

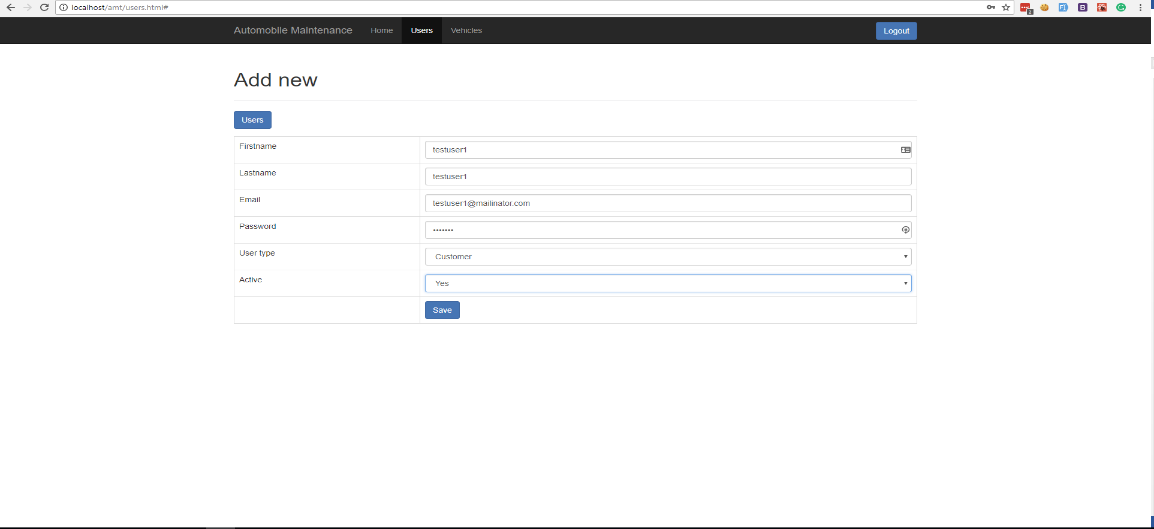
* Home: this link for main page.
* User: this link for user module (list, add, edit, view, delete).
* Vehicle: This link for vehicle module (list, add, edit, view, delete vehicles and maintenances).

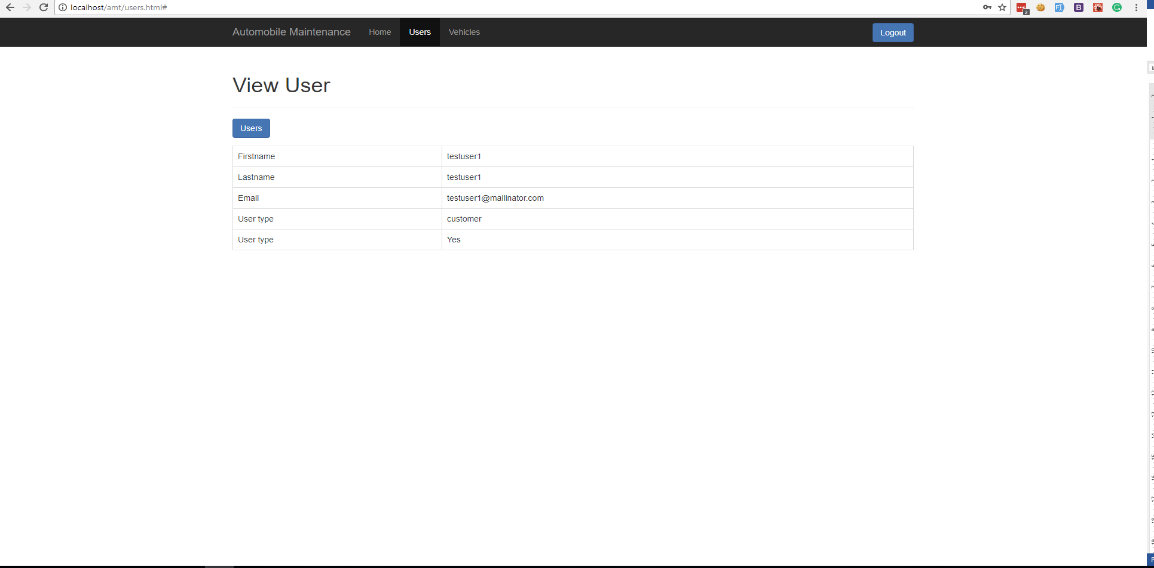


1. User management.

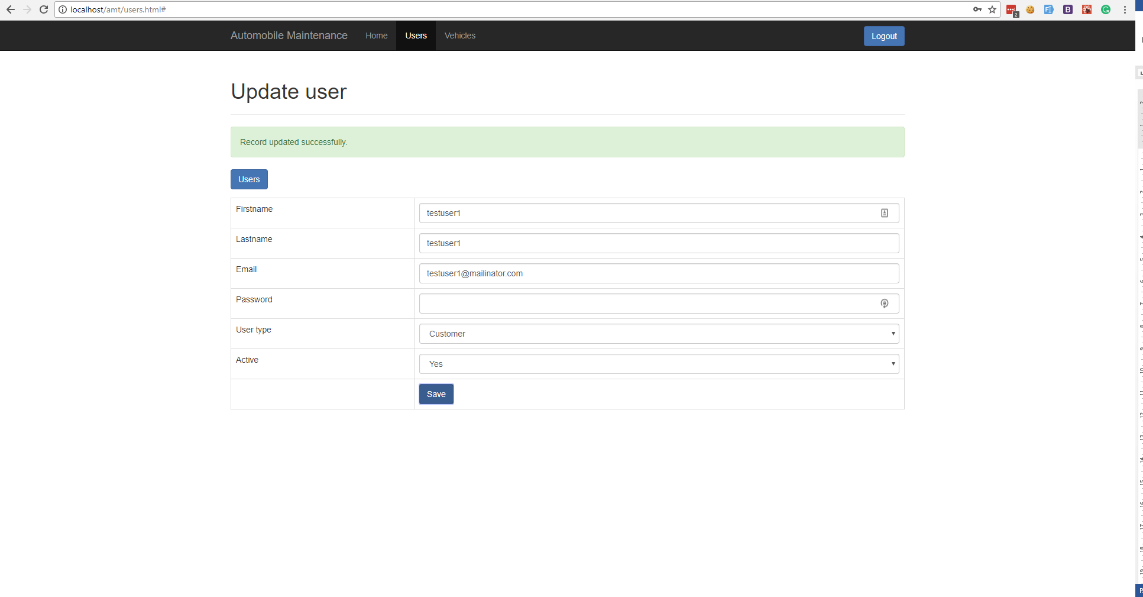
* List all users.



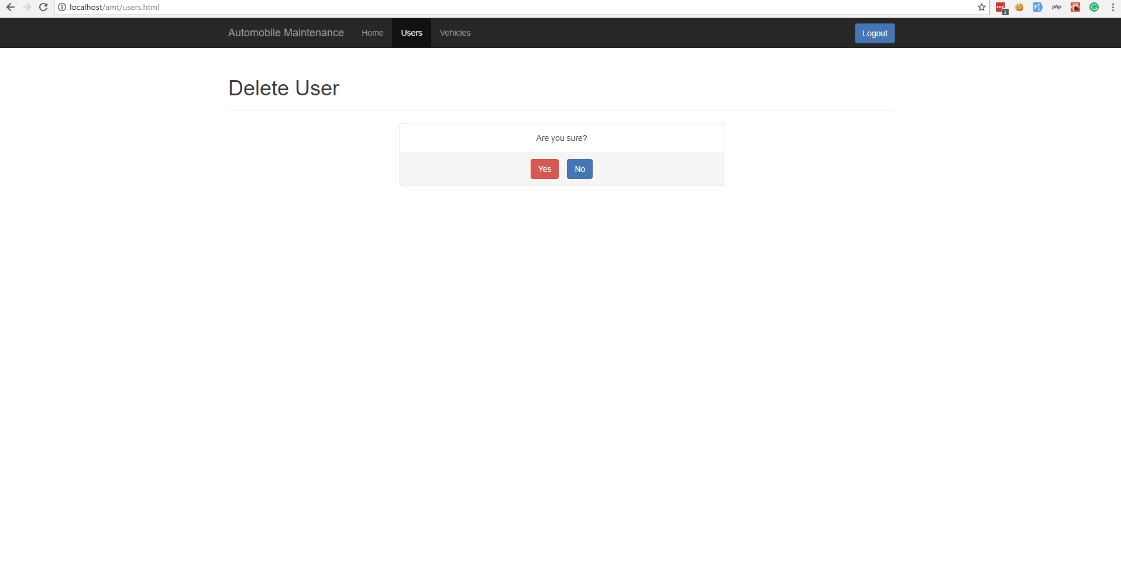
* Add new user
* .
* View user.



* Edit user

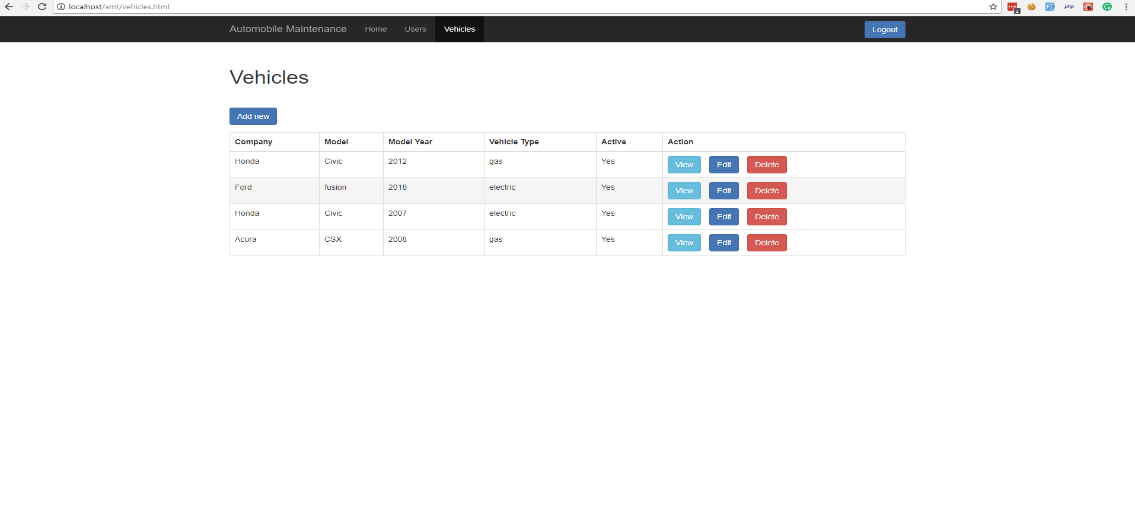


* Delete user

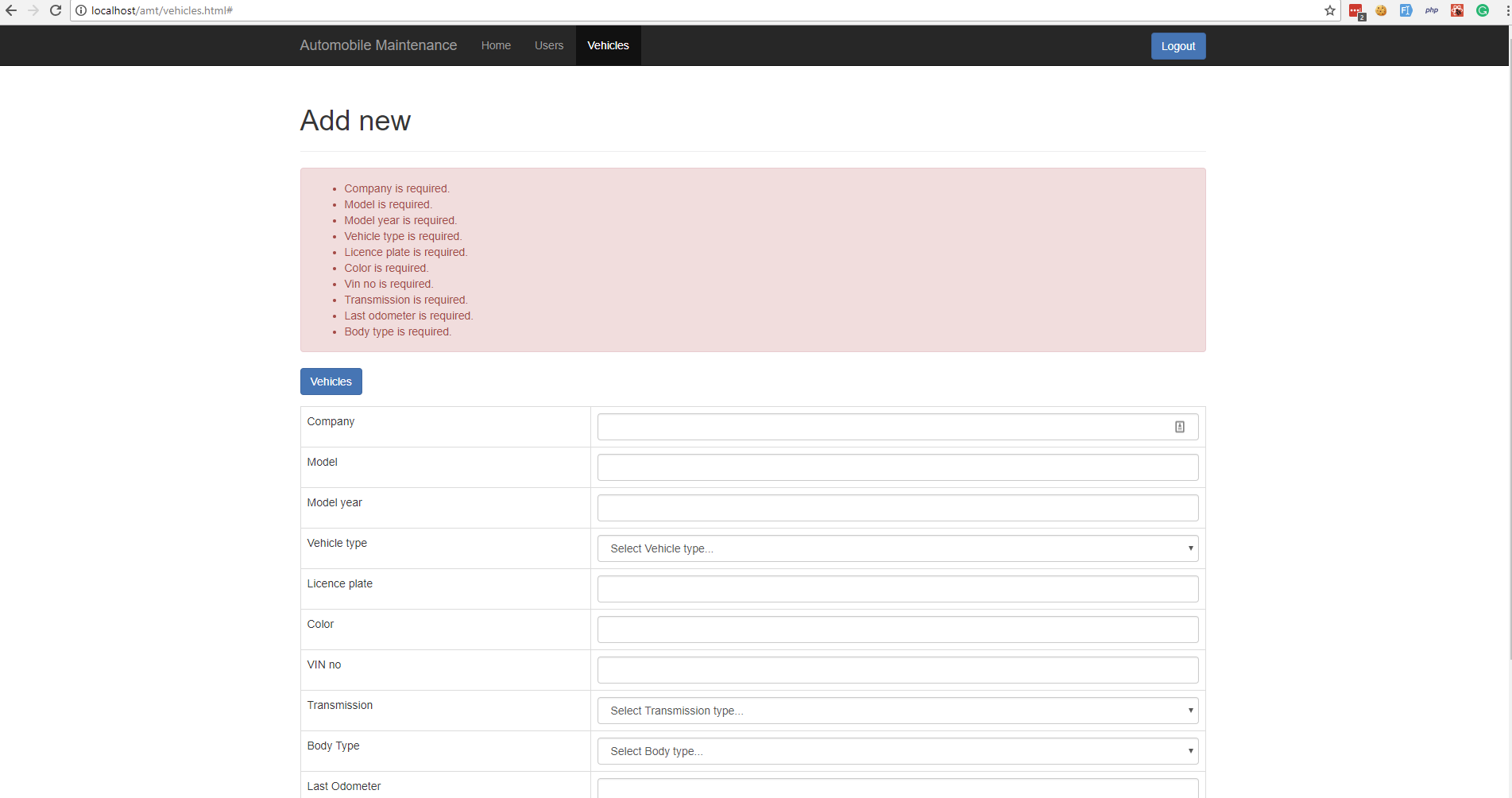


1. Vehicle and maintenance management.

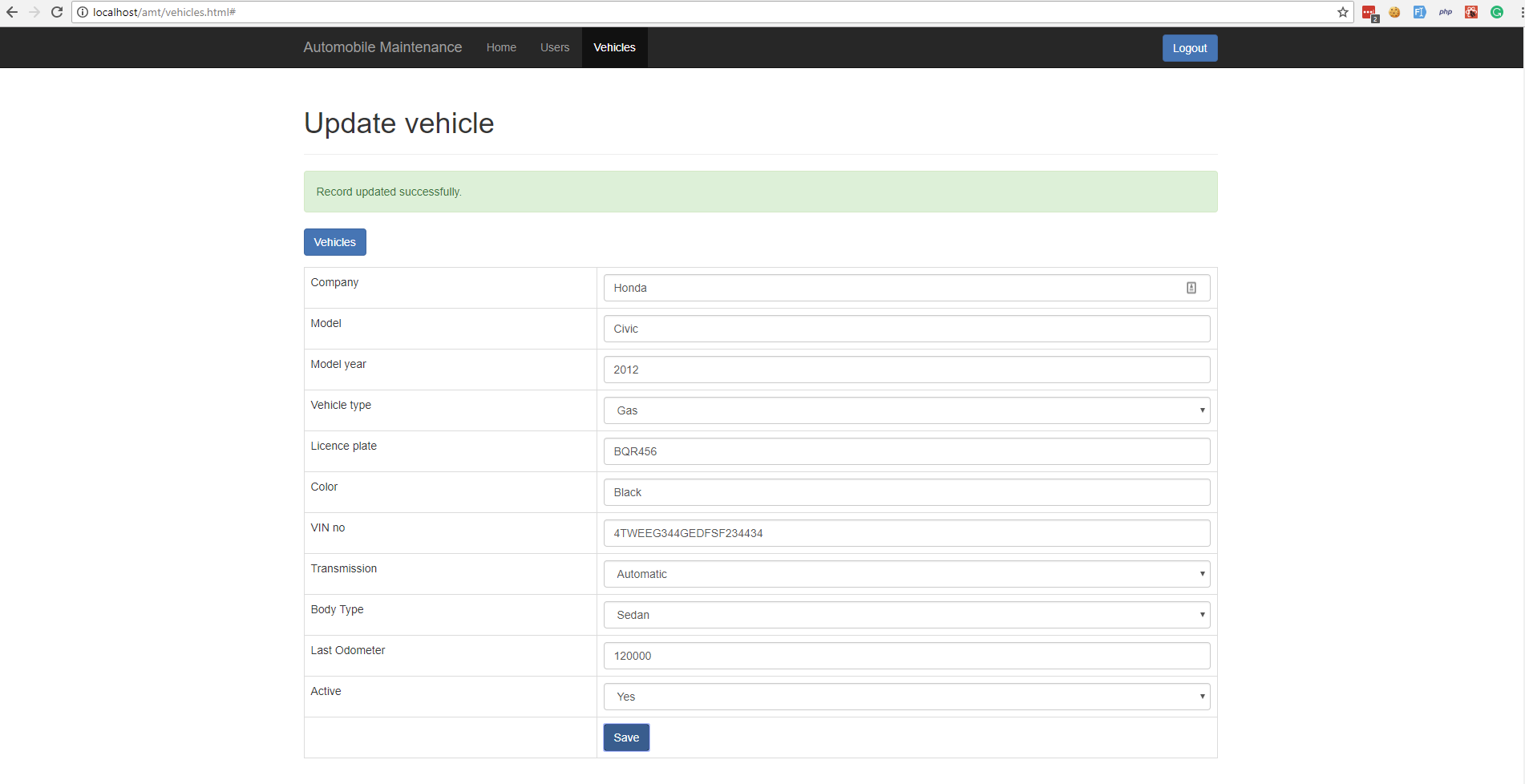
* List vehicles



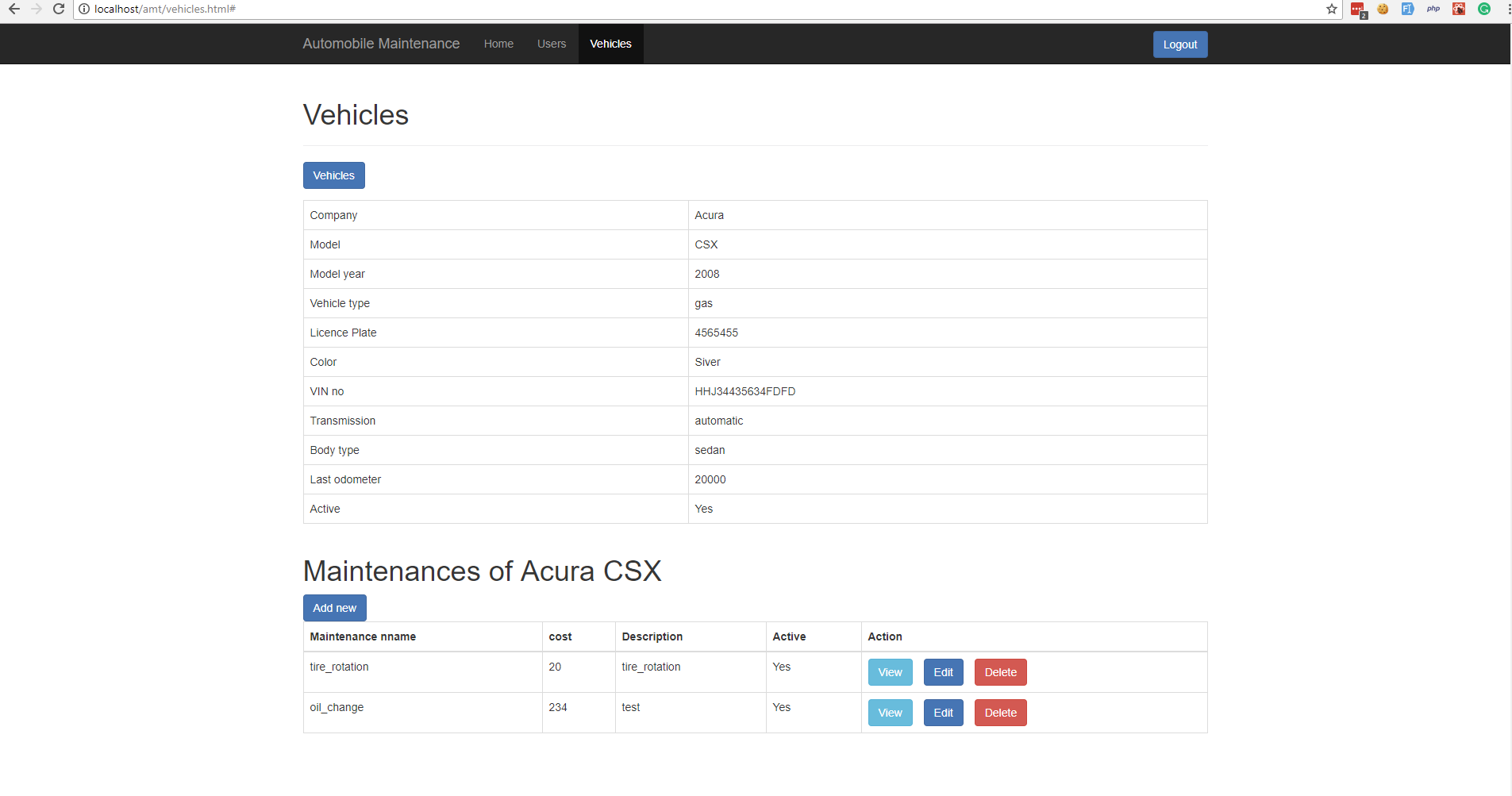
* add vehicle



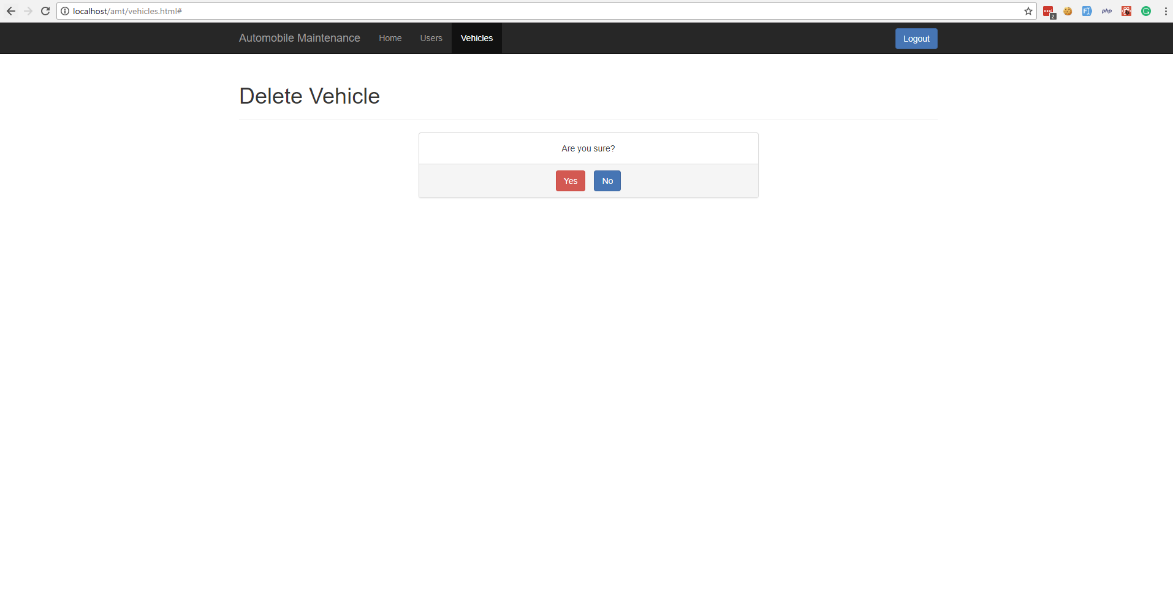
* edit vehicle



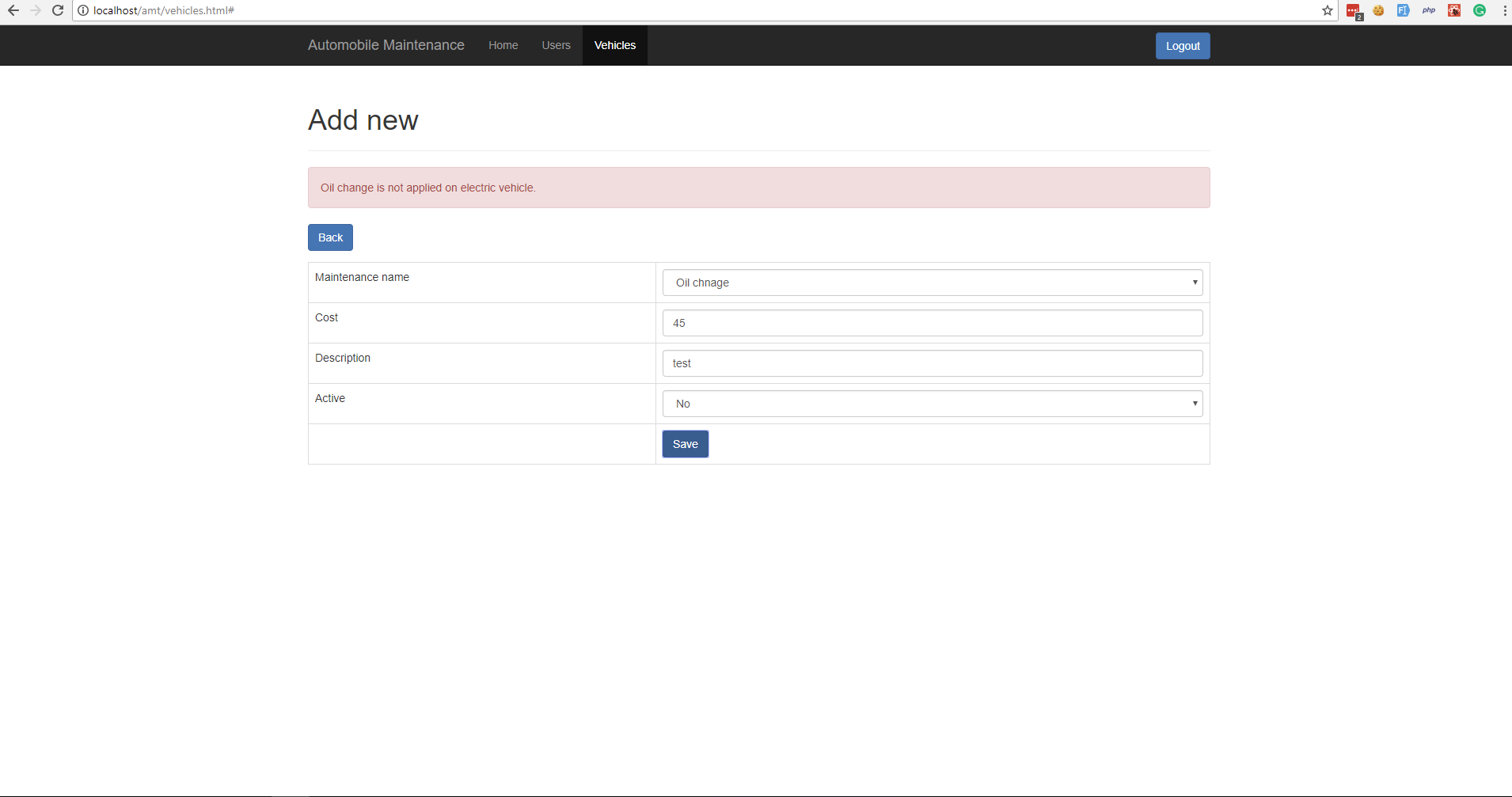
* View vehicle and list related maintenances



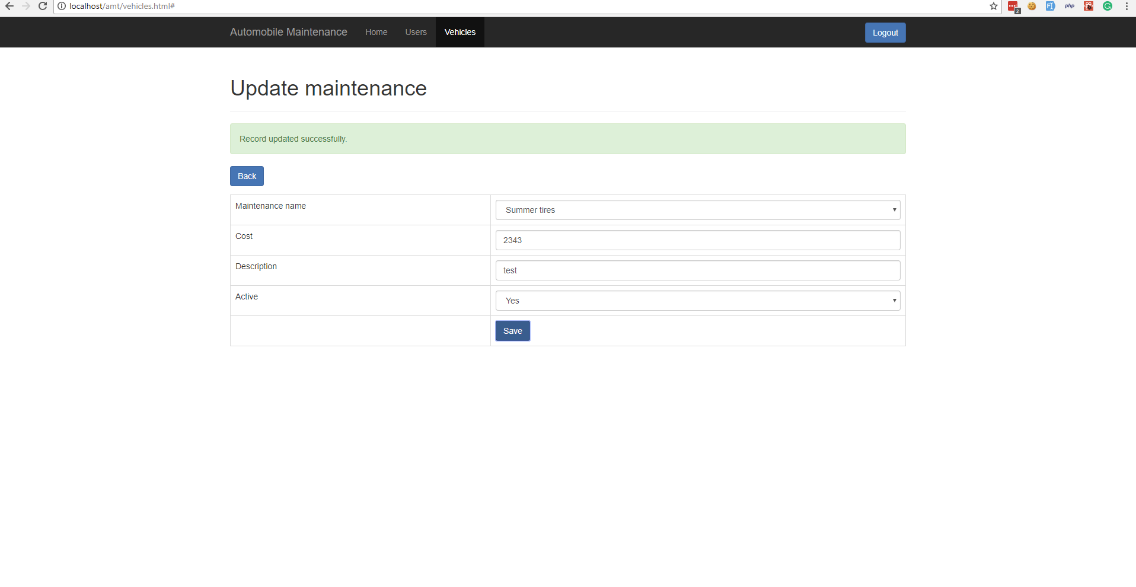
* Delete vehicle



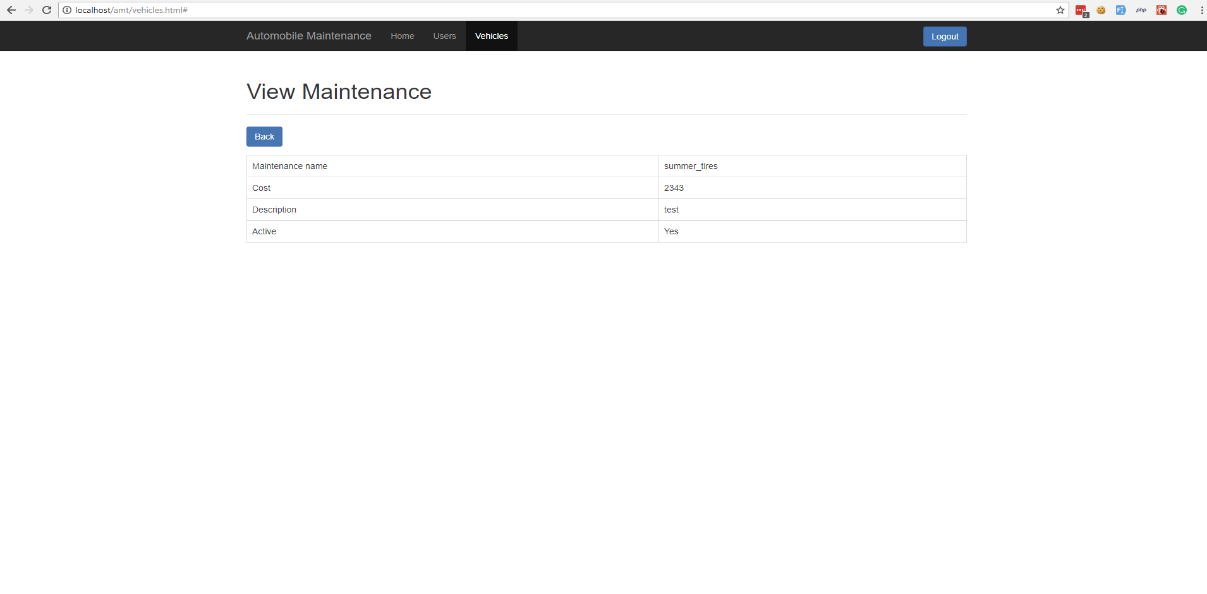
* Add new maintenance



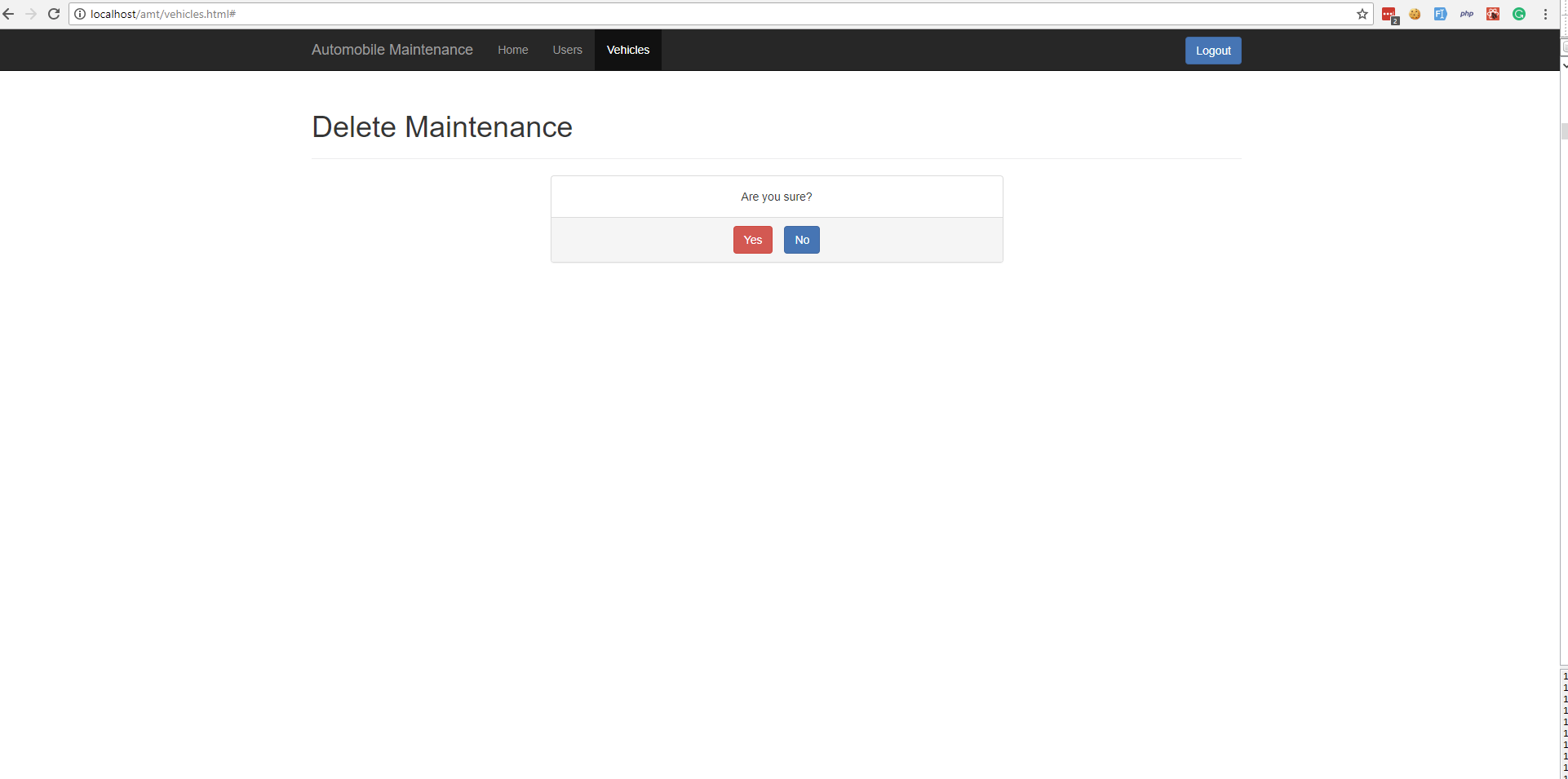
* Edit maintenance



* View maintenance



* Delete maintenance



Php unit test:

* Installation steps :

1. Create composer.json in to root directory.
2. Add following script into composer.json

{

"require": {

},

"require-dev": {

"phpunit/phpunit": "3.7.14"

},

"autoload": {

"psr-0": {

"API": ""

}

}

}

1. Open command prompt or windows power shell and go to project root directory (eg. c://xampp/htdocs/amt/).
2. now run command "composer install".

* Run php unit test cases :

1. Go to project root directory : /amt
2. Run command like : phpunit Test\API\lib\function\_test.php
3. lib
   * function\_test.php : phpunit \amt\Tests\API\lib\function\_test.php
4. models :
   * auth\_test.php : phpunit \amt\Tests\API\models\auth\_test.php
   * - maintainance\_model\_test.php : phpunit \amt\Tests\API\models\maintainance\_model\_test.php
   * - user\_model\_test.php : phpunit \amt\Tests\API\models\user\_model\_test.php
   * - vehicle\_model\_test.php : phpunit \amt\Tests\API\models\vehicle\_model\_test.php