



Kopienas mājas lapa

# MERN aplikācijas izstrāde

Pasākumu finansē: Eiropas Jūrlietu un zivsaimniecības fonds, projekts "Viedo ciemu attīstība piekrastes teritorijās". Nr. 19-00-F043.0443-000001.  
VRG "Partnerība laukiem un jūrai", sadarbībā ar vadošo partneri VRG "Liepājas rajona partnerība"



NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



EIROPAS SAVIENĪBA  
Eiropas Jūrlietu un  
zivsaimniecības fonds



Atbalsta Zemkopības ministrija un Lauku atbalsta dienests

## Mēs zinām

- kas ir WEB lapa (HTML, CSS, JS)
- kā glabā un publicē versionētu WEB lapu
- funkcionālas/nefunkcionālās prasības
- API tehnoloģiju izmantošana mājas lapas veidošanā
- WEB lapas testēšanas iespējas
- Ir iestrādes ziņojuma dēļa dizainā un paraugs izstrādātai sagatavei

Ziņojuma dēļa protoipa piemērs ir

## Šodien uzzināsim

- MERN aplikācija - kas tas ir
- Visual studio Code un izmantotās bibliotēkas
- Klients, Serveris un bibliotēku sagatave
- Pirmā mūsu klienta aplikācija

Pirmā Full Stack aplikācija, ko radam paši

# Vides sagatavošana

Programmēšanas vide -

<https://code.visualstudio.com/>

JS programmēšanas framework

<https://nodejs.org/en/>

# MERN aplikācija

MERN nozīmē - MongoDB, Express, React, Node - tās ir 4 atslēgas tehnoloģijas, kas nodrošina pilnu izstrādes vidi:

- MongoDB - dokumentu datubāze
- Express(.js) - Node.js WEB framework
- React(.js) - klienta puses JavaScript framework
- Node(.js) - pamata framework JavaScript web serverim



# Ko mēs tehniski gribam uzprogrammēt?

**CRUD** Nozīme:

**CRUD** ir abreviatūra no programmēšanas pasaules, kas apzīmē 4 funkcijas. Tās nepieciešamas lai izveidot pamata lietas jebkurai aplikācijai.

C - create - izveidot ierakstu

R - read - nolasīt ierakstu no datu glabātuves

U - update - atjaunot/labot ierakstu datu glabātvē,

D - delete - dzēst ierakstu no datu glabātuves.

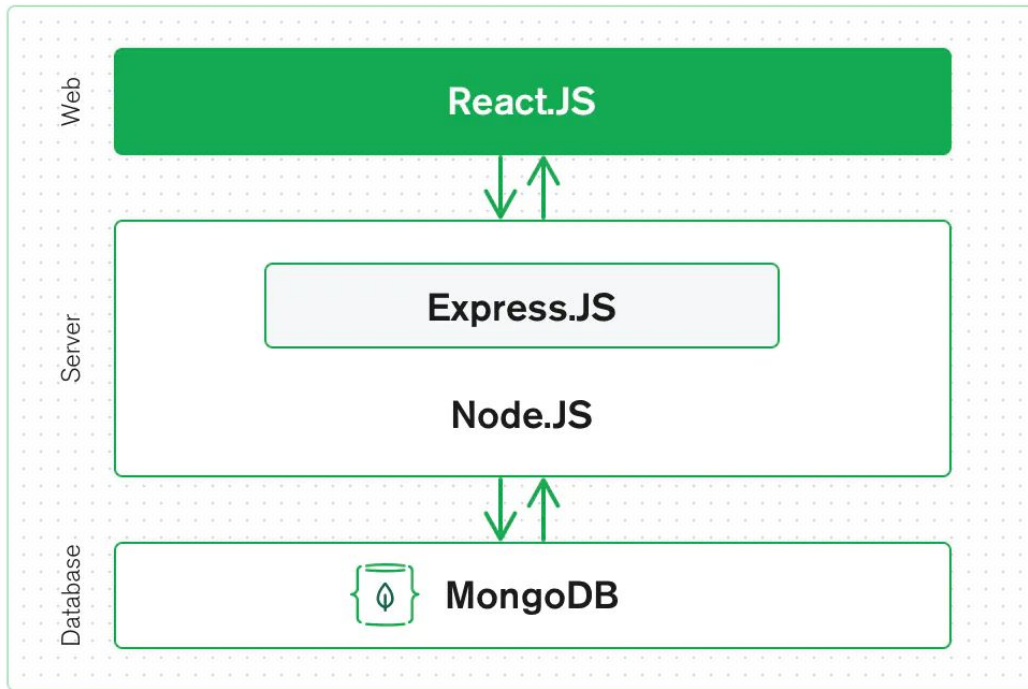


# Kā tas izskatīsies?

The screenshot displays a web browser window with the following elements:

- Browser Address Bar:** Shows the URL `localhost:3000`.
- Page Header:** A white bar with the title "Ziņojumu dēlis" and a small profile icon.
- Message Cards:**
  - Card 1:** Author "Daina", posted "7 days ago". Content includes "#vietējais #svaigs #jauks" and the number "26565812". It has "3 LIKE" and a "DELETE" button.
  - Card 2:** Author "Jolanta", posted "7 days ago". Content includes "#zemeses #vasara #2021" and the number "3425322". It has "1 LIKE" and a "DELETE" button.
- Form "Jauns ziņojums" (New Message):**
  - Input field for "Autors" (Author) containing "Daina".
  - Input field for "Kontakti" (Contacts).
  - Input field for "Ziņojums" (Message).
  - Input field for "Hashtagi" (Hashtags).
  - File upload section: "Choose File" button, "No file chosen" text.
  - Buttons: "SAGLABĀT" (Save) in blue and "NOTĪRĪT" (Post) in pink.
- Taskbar:** Shows the Windows Start button, search bar with "Rakstiet šeit, lai meklētu", and various application icons.
- System Tray:** Shows the date and time as "4:06 PM 7/7/2021" and temperature as "30°C".

# Arhitektūra?



# Ar ko sākt?

1. Atveram Visual Studio Code logu
2. Izveidojam tukšu direktoriju uz Desktop, kur glabāsies mūsu kods - SLUDINAJUMI\_PROJEKTS
3. Drag un Drop iekš Visual Studio Code
4. View -> Terminal
5. Sadalam Terminal daļu divās pusēs - viena būs priekš Client un otra Server
6. komanda "clear" - notīra Terminal logu
7. Izvedojam 2 direktorijas - "client" un "server"
8. Pirmajā terminal logā - cd client
9. Otrajā terminal loga - cd server



# Ar ko sākt?

## client

1. Inicializējam React aplikāciju  
`npx create-react-app ./`
  - instalējas klienta puse

## server

2. Izvedojam failu `index.js`
3. izveidojam tukšu `package.json` ar komandu  
`npm init -y`
4. Inicializējam servera puses izmantotās bibliotēkas  
`npm install body-parser cors express mongoose nodemon`
5. `index.js` visas izmantotās pakotnes importējam

```
import express from 'express';  
import bodyParser from 'body-parser';  
import mongoose from 'mongoose';  
import cors from 'cors';
```

# Ar ko sākt?

client

8. Gaidam kad klienta puse ir uzinstalējusies

9. Instalējam izmantotas bibliotēkas

```
npm install axios moment react-file-base64  
redux redux-thunk
```

10. Izdzēšam src direktoriju un uztaisam jaunu  
“src”

11. Izveidojam failu iekš src -> index.js

server

6. package.json rindiņa zem “main”:”index.js”  
“type”:”module”,

7. tūpat nodzēšam test scriptu un ieliekam start skriptu

```
■ "start": "nodemon index.js"
```

# Ar ko sākt?

client

server

12. index.js rakstam bibliotēku importu

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import App from './App';
```

```
ReactDOM.render(<App />, document.getElementById('root'));
```

13. Izvedojam src -> App.js failu

```
import React from 'react';  
const App = () => {  
  return (  
    <div><h1>Hello World!</h1></div>;  
  )  
}
```

# Ar ko sākt?

client

server

14. App.js failam turpinājums

```
export default App;
```

15. saglabājam abus client puses failus

16. palaižam aplikāciju

```
npm start
```

# Pirmā mūsu aplikācija

**Hello World!**



**PALDIES**

Apmācības nodrošina NVO "Piekrastes Konvents". Vairāk info - [info@piekrasteskonvents.lv](mailto:info@piekrasteskonvents.lv)