



Venkatesh Mogili
#WebGuru

తెలుగు లో

PART-1



MERN Stack Application

VM Training

Software Requirements

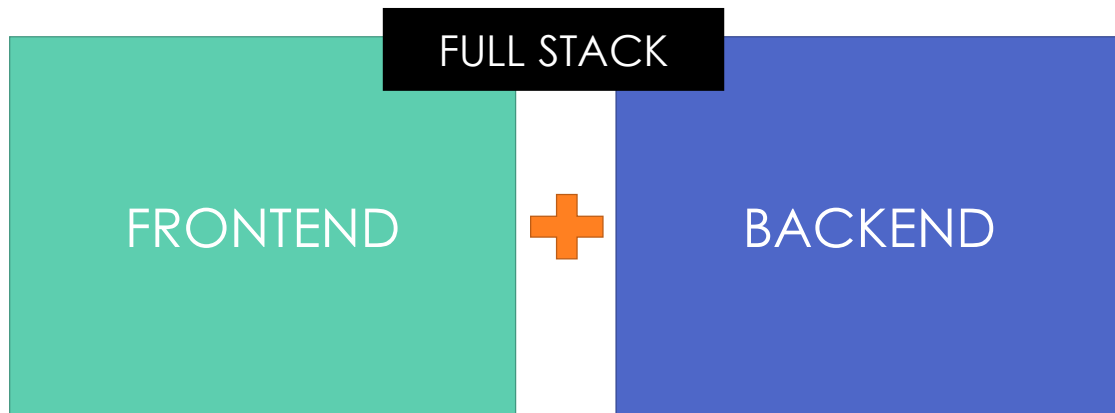
- **Visual Studio Code Editor**
➤ <https://code.visualstudio.com/download>
- **NodeJS 14+**
➤ <https://nodejs.org/en/download/>
- **XAMPP 7+**
➤ <https://www.apachefriends.org/download.html>
- OR
- **MySQL Installer 8+**
➤ <https://dev.mysql.com/downloads/installer/>



Venkatesh Mogili
#WebGuru

M E R N

Project Structure



Venkatesh Mogili
#WebGuru

M E R N

Project Features

- REGISTRATION
- LOGIN
- VIEW PROFILE
- EDIT PROFILE
- DELETE PROFILE
- CHANGE PASSWORD
- FORGET PASSWORD
- VERIFY EMAIL
- RESET PASSWORD
- LOGOUT
- ROLE MANAGEMENT, JWT AUTHENTICATION AND EMAIL SERVICE.
- CORS, SQL Injection Prevention.



Venkatesh Mogili
#WebGuru

M E R N

Project Setup Backend

- **Backend: (MySQL)**
- **Create a new nodejs project:**
- Create a folder (**mern-stack-app**) and **cd mern-stack-app** and **npm init -y**
- **npm i express, cors, ejs, dotenv, nodemon**
- **Create an express server**
- **Database setup and connect to nodejs**
- **Create REST APIs and Authentication with JWT**
- **Setup Email Service and Email Templates**



Venkatesh Mogili
#WebGuru

M E R N

MySQL Installation

There are several ways to install MySQL.

1. **MySQL Installer**
<https://dev.mysql.com/downloads/installer/>
(OR)
 2. **XAMPP has MySQL server.**
<https://www.apachefriends.org/download.html>
- ✓ Download any of the above software and install them and run. By default MySQL will run on port 3306.
 - ✓ Any one MySQL can be run at a time, more than one servers can't be run. If you update port, then you can use more than one servers, but it's not recommended.



XAMPP



Venkatesh Mogili
#WebGuru

M E R N

Database Setup

Database: **mern-stack-database**



Tables:

- 1) **users** – id, name, **email**, **password**, **profilepic**, **role**, **verified**, **refreshToken**, **expireToken**
- 2) **blacklist** – id, token, email
- 3) **courses** – id, **coursename**, **description**, **technologies**

Note: createdAt and updatedAt are common for any table.



Venkatesh Mogili
#WebGuru

M E R N

MySQL Connection in NodeJS

1. Install mysql using below command
➤ **npm install mysql**
2. Import mysql
3. Create Connection for mysql
4. Check Connection is established or not.
5. Once database is created, update the connection configuration.

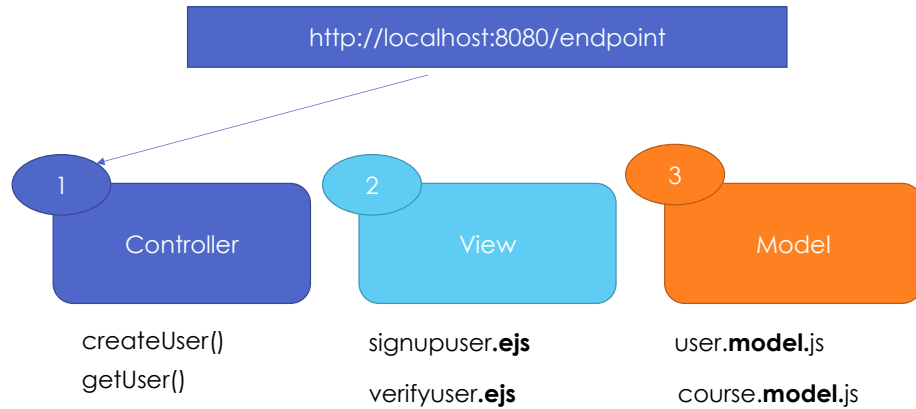


Venkatesh Mogili
#WebGuru

M E R N

MVC Structure

1. Controllers
2. Models
3. Views
4. Routes



Venkatesh Mogili
#WebGuru

M E R N

Routing

➤ User CRUD Operations:

- `/signup (POST)`
- `/getUser/:id (GET)`
- `/editUser/:id (PUT)`
- `/deleteUser/:id (PUT)`

➤ Authentication Flow:

- `/verify/:token (POST)`
- `/signin (POST)`
- `/changepassword (PUT)`
- `/forgetpassword (POST)`
- `/resetpassword (PUT)`
- `/logout (POST)`

1

2



Venkatesh Mogili
#WebGuru

M E R N

Middleware

- `commonResponse()`
- `mailservice()`
- Authentication middleware with jwt verification.



Venkatesh Mogili
#WebGuru

M E R N

Email Service

- Create [SendGrid](#) Account and GET API Key.
- You can also use [nodemailer-smtp-transport](#) for free email service.

```
var nodemailer = require('nodemailer');
var smtpTransport = require('nodemailer-smtp-transport');

var transporter = nodemailer.createTransport(
  smtpTransport({
    service: 'gmail',
    host: 'smtp.gmail.com',
    auth: {
      user: 'youremailaddress@gmail.com',
      pass: 'password',
    },
  })
);
```



Venkatesh Mogili
#WebGuru

M E R N

Setup Email Service and Templates

```
var nodemailer = require("nodemailer");
const sendgridTransport = require("nodemailer-sendgrid-transport");
const transporter = nodemailer.createTransport(
  sendgridTransport({
    auth: {
      api_key: `YOUR_SENDGRID_API_KEY`,
    },
  })
);
const path = require("path");
var ejs = require("ejs");

var pathreset = path.resolve(__dirname, "../views/templates/signupuser.ejs");
```



Venkatesh Mogili
#WebGuru

M E R N

Setup Email Service and Templates

```
var mailOptions = {
  from: "mogiliv3@gmail.com",
  to: req.body.email,
  subject: "Account Created Successfully!",
  html: data,
};
ejs.renderFile(pathreset, {user, token}, function(err, data) {
  if (err) {
    console.log(err);
  }
});
```

```
else {
  transporter.sendMail(mailOptions, function(
    error, info) {
    if (error) {
      console.log(error);
      res.send({success: false, data: null, message: error ? error : "Something went wrong"});
    } else {
      res.send({success: true, data: null, message: "Mail sent successfully!"});
    }
  });
}
```



Venkatesh Mogili
#WebGuru

M E R N

Add Authentication With JWT

- -----Basic Logout (Can be hacked)
 - Create a authentication middleware function to verify whether JWT Token is valid or not.
- -----Strict Logout (Can't be hacked)
 - Since, the JWT Token can't be destroyed until it expires, We should maintain a separate list of users who are logged in.
 - On logout, we should update the loggedIn user's token as Empty and need to check in authentication middleware function. If It is the same as JWT token then it should be valid, otherwise invalid.



Venkatesh Mogili
#WebGuru

M E R N

Project Setup Frontend

- Frontend: (React)
 - Create a react app using create react app:
 - **npm create-react-app frontend**
 - Setup Bootstrap, Font Awesome, Simple React Validator, React Router DOM, Redux
 - **npm i bootstrap@5.0.2 font-awesome@4.7.0 simple-react-validator@1.6.0 react-router-dom@5.2.0 redux react-redux redux-thunk**



Venkatesh Mogili
#WebGuru

M E R N

Bootstrap and Common Functionality

- Import bootstrap and font-awesome.
- Create a common utility for validation.
- Create a common utility for API Calls with Redux.
- Authentication
 - isAuthenticated
 - getAuthToken
 - clearLocalStorage



Venkatesh Mogili
#WebGuru

M E R N

Routing

- Create Public Route and Private Route Components.
- Add routing for Homepage, Sign Up, Sign In, Forget Password, Reset Password, Change Password, Dashboard, 404 Not Found.
- / - home
- /signup – Sign up
- /signin – Sign in
- /forgetpassword – Forget password
- /resetpassword/:token – Reset Password
- /changepassword – Change Password
- /dashboard – Dashboard
- /viewprofile – View/Edit Profile
- * - 404 Not Found.

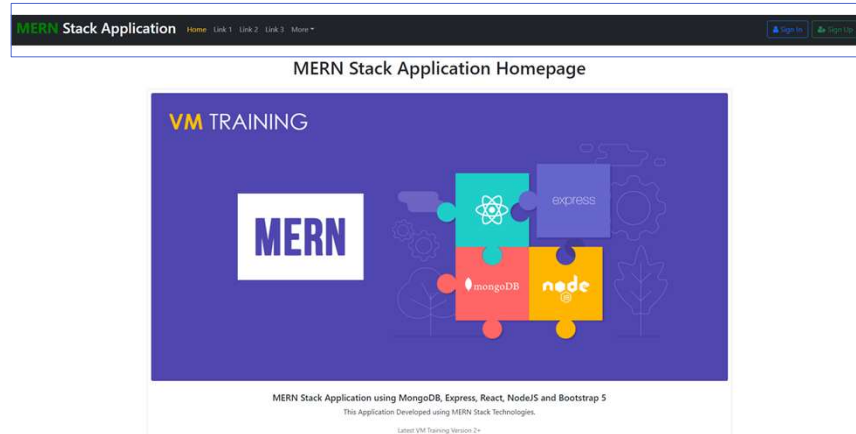


Venkatesh Mogili
#WebGuru

M E R N

Common Components

- Navbar
- Footer



Venkatesh Mogili
#WebGuru

M E R N

Registration

- Component Name – Register
- Required Fields:
 - Username
 - Password
 - Confirm Password
- Validation:
 - Valid Email and Password
 - Password and Confirm Password should be matched.
- API: (POST Method)
- /signup
- Request Body:
 - {username:"test@gmail.com", password:"test@123"}
- On Successful sign up Redirect to /signin

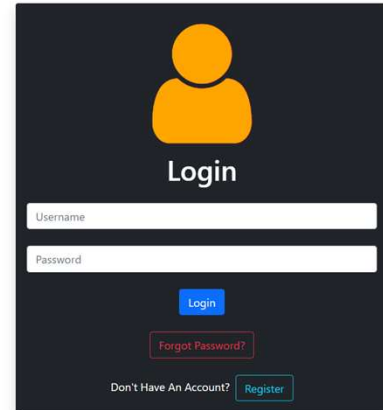


Venkatesh Mogili
#WebGuru

M E R N

Login

- Component Name – Login
- Required Fields:
- Username
- Password
- Validation:
- Valid Email and Password
- API: (POST Method)
- /signin
- Request Body:
- {username:"test@gmail.com", password:"Test@123"}
- On Successful login Redirect to /dashboard

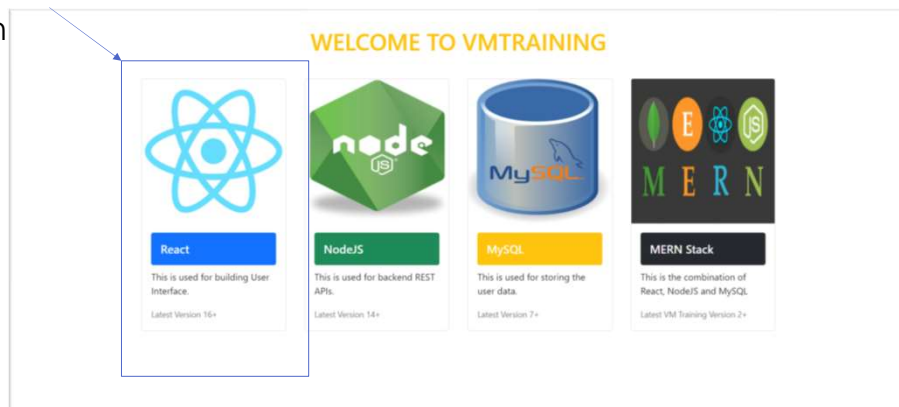


Venkatesh Mogili
#WebGuru

M E R N

Dashboard

- Card Item can be made as common component

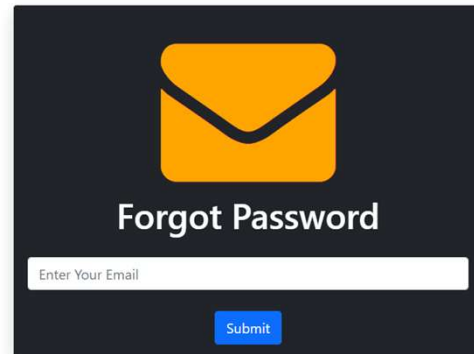


Venkatesh Mogili
#WebGuru

M E R N

Forget Password

- Component Name – ForgetPassword
- Required Fields:
- Email
- Validation:
- Valid Email Address
- API: (POST Method)
- /forgetpassword
- Request Body:
- {username:"test@gmail.com"}
- On Successful update Redirect to /signin

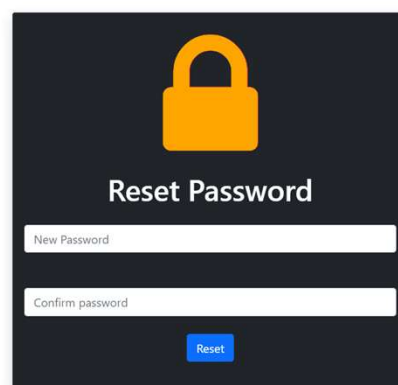
A dark-themed mockup of a 'Forgot Password' form. At the top is a large orange envelope icon. Below it, the text 'Forgot Password' is centered. Underneath is a white input field with the placeholder text 'Enter Your Email'. At the bottom right is a blue 'Submit' button.

Venkatesh Mogili
#WebGuru

M E R N

Reset Password

- Component Name – ResetPassword
- Required Fields:
- New Password
- Confirm Password
- Validation:
- Valid Password
- New password and confirm password should be matched.
- API: (PUT Method)
- /resetpassword
- Request Body:
- {password:"Test@111"}
- On Successful reset Redirect to /signin

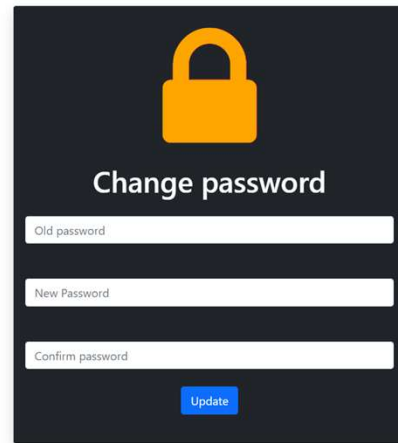
A dark-themed mockup of a 'Reset Password' form. At the top is a large orange padlock icon. Below it, the text 'Reset Password' is centered. Underneath are two white input fields: the first with the placeholder 'New Password' and the second with 'Confirm password'. At the bottom right is a blue 'Reset' button.

Venkatesh Mogili
#WebGuru

M E R N

Change Password

- Component Name – ChangePassword
- Required Fields:
- Old Password
- New Password
- Confirm Password
- Validation:
- Valid Password
- New password and confirm password should be matched.
- API: (PUT Method)
- /changepassword
- Request Body:
- {oldpassword:"Test@123",newpassword:"Test@222"}
- On Successful update Redirect to /signin

A dark-themed UI for changing a password. It features a yellow padlock icon at the top. Below the icon, the text "Change password" is displayed. There are three input fields: "Old password", "New Password", and "Confirm password". At the bottom right, there is a blue "Update" button.

Venkatesh Mogili
#WebGuru

M E R N

View Profile, Edit Profile, Logout

- Make a dropdown for logged in user.
- Make a ViewProfile Component to Display the logged in user data and to edit the user data.
- Once we click on Logout, call the /logout api and clear the localStorage.

Venkatesh Mogili ▾

View Profile

Edit Profile

Logout



Venkatesh Mogili
#WebGuru

M E R N

Thank You



Check out other courses



<https://vmtraining.netlify.app>



VWatch



IMplement



Try Experiments

VM Online Training