Server Side Development

Pada project ini anda akan membuat sebuah aplikasi web menggunakan REST server sebagai backend dan front end library sebagai front end (react, angular, vue). Tema pada project ini adalah database pemain sepakbola.

Anda harus membuatkan REST server dan front end untuk sistem yang dapat menyimpan data pemain sepakbola, pada sistem ini terdapat user yang dapat mengisikan data ke tabel Player, untuk dapat mengisi data ke tabel player user harus melakukan login terlebih dahulu, login ini dilakukan untuk mendapatkan token JWT, dimana token jwt ini akan digunakan sebagai syarat authentikasi. Berikut ini syarat syarat pembuatan backend nya:

1. Menggunakan database mysql.
2. Pembuatan Server REST menggunakan bahasa PHP berorientasi objek atau framework PHP Laravel / Codeigniter atau menggunakan nodejs.
3. Berikut ini desain database yang akan dibuatkan REST server nya.
4. Isikan secara manual username dan password yang terenkripsi untuk mendapatkan token JWT.

A screenshot of a cell phone

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel | key | kolom | Keterangan |
| User |  | id | Primary key |
|  |  | username | Berisi username |
|  |  | password | Berisi password yang terenkripsi |
|  |  | createdDate | Berisi tanggal dibuatnya data user dalam format unix timestamp |
|  |  | modifiedDate | Berisi tanggal berubahnya data user dalam format unix timestamp |

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel | key | kolom | Keterangan |
| Player |  | id | Primary key |
|  |  | posisi | Berisi pilihan posisi pemain sepakbola |
|  |  | nama | Berisi nama pemain sepakbola |
|  |  | nomor\_punggung | Berisi nomor punggung pemain |
|  |  | createdBy | Berisi foreingn key dari tabel user pada saat data dibuat |
|  |  | modifiedBy | Berisi foreign key dari tabel user pada saat data diubah |
|  |  | createdDate | Berisi tanggal data dibuat dalam format unix timestamp |
|  |  | modifiedDate | Berisi tanggal data diubah dalam format unix timestamp |

1. Buatlah endpoint REST yang dibutuhkan untuk mengakomodasi fitur fitur berikut ini :

Endpoint Requirement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource | GET | POST | PUT | DELETE |
| /users/auth | NA | Login ke sistem melakukan pengecekan username dan password untuk mendapatkan token JWT | NA | NA |
| /users | Mengembalikan daftar seluruh user pada tabel User | Membuat data user baru pada tabel user | N/A | N/A |
| /users/{id} | Mengembalikan satu baris data pada tabel User berdasarkan parameter {id} | N/A | Mengubah satu baris data pada tabel User berdasarkan parameter {id} | Menghapus satu baris data pada tabel User berdasarkan parameter {id} |
| /players | Mengembalikan daftar seluruh player pada tabel Player | Membuat data pleyer baru pada tabel Player | N/A | N/A |
| /players/{id} | Mengembalikan satu baris data pada tabel Player berdasarkan parameter {id} | N/A | Mengubah satu baris data pada tabel User berdasarkan parameter {id} | Menghapus satu baris data pada tabel User berdasarkan parameter {id} |

Detail Spesifikasi

|  |  |
| --- | --- |
| Judul | Login Ke sistem dan mendapatkan token JWT |
| Url | /users/auth |
| Method | POST |
| URL Params | None |
| Data Params | {  "username":"superuser",  "password":"passwordraw",  } |
| Success Response | HTTP Response Code : 200  Response :  {  token : "sadkhu139871babsdm!@#sadkba"  } |
| Error Response | HTTP Response Code: 422  Response : {error:”Invalid Username”}  HTTP Response Code: 422  Response : {error:”Invalid Password”} |
| Sample Call | $.ajax({  url: "/users/auth",  dataType: "json",  data : {  u: {  "username":"superuser",  "password":" passwordraw",  }  },  type : "POST",  success : function(r) {  console.log(r);  }  }); |
| Notes | Pada proses POST ke url /users/auth dilakukan validasi input dengan rule sebagai berikut :   1. Username : required, minimal 4 huruf alphanumerik 2. Password : required, minimal 6 karakter maksimal 8 karakter alhpanumerik   Jika data yang dikirim lolos validasi maka backend melakukan   1. Enkripsi input password. 2. Membandingkan username dan password yang dikirim dengan data pada database. 3. Mengenerate dan return success response jwt token jika username dan password cocok. 4. Return error yang sesuai dengan error response jika data username dan password tidak cocok. |

|  |  |
| --- | --- |
| Judul | Mengembalikan daftar seluruh user pada tabel User |
| Url | /users |
| Method | GET |
| URL Params | None |
| Data Params | JWT Token |
| Success Response | HTTP Response Code : 200  Response :  {  "data" : [  {  "id": 1,  "username": "superuser",  "createdDate": "1576204776" ,  "modifiedDate": "1576204776"  },  {  "id": 2,  "username":"mediumuser" ,  "createdDate": "1576204776" ,  "modifiedDate": "1576204776"  }  ],  } |
| Error Response | HTTP Response Code : 404  Response : {error: "no user found"} |
| Sample Call | $.ajax({  url: "/users",  dataType: "json",  type : "GET",  success : function(r) {  console.log(r);  }  }); |
| Notes | Perhatikan pada proses get /users field password tidak ikut dikembalikan. |

|  |  |
| --- | --- |
| Judul | Membuat data user baru pada tabel user |
| Url | /users |
| Method | POST |
| URL Params | None |
| Data Params | JWT Token  {  "username":"superuser",  "password":"passwordraw",  } |
| Success Response | HTTP Response Code : 201  Response :  {  "id": "1",  "username":"superuser",  "createdDate":"1576204776"  "modifiedDate": "",  } |
| Error Response | Code : 422  Content : {error:”Invalid Username”}  Code : 422  Content : {error:”Password is to short”} |
| Sample Call | $.ajax({  url: "/users",  dataType: "json",  data : {  u: {  "username":"superuser",  "password":" passwordraw",  }  },  type : "POST",  success : function(r) {  console.log(r);  }  }); |
| Notes | Pada proses POST ke url /users dilakukan validasi input dengan rule sebagai berikut :   1. Username : minimal 4 huruf alphanumerik 2. Password : minimal 6 karakter maksimal 8 karakter alhpanumerik   Jika data yang dikirim lolos validasi maka backend melakukan   1. Enkripsi input password. 2. Membuat created\_date sesuai waktu pembuatan data. 3. Menyimpan data ke database. 4. Mengembalikan data berformat json sesuai pada Success Response. 5. Pada saat data pertama kali di create modified\_date isinya kosong. |

|  |  |
| --- | --- |
| Judul | Mengembalikan satu baris data pada tabel User berdasarkan parameter {id} |
| Url | /users/:id |
| Method | GET |
| URL Params | id = [integer]  Example  id = 10  urls = /users/10 |
| Data Params | JWT Token |
| Success Response | HTTP Response Code: 200  Response :  {  "id": "1",  "username":"superuser",  "createdDate": "1576204776" ,  "modifiedDate": "1576204776"  } |
| Error Response | HTTP Response Code: 422  Response : {error:”user not found”} |
| Sample Call | $.ajax({  url: "/users/1",  dataType: "json",  type : "GET",  success : function(r) {  console.log(r);  }  }); |
| Notes | Pada proses get ke url /users/:id dikembalikan data user berupa id, username, created\_date dan modified\_date tanpa field password. |

|  |  |
| --- | --- |
| Judul | Mengubah satu baris data pada tabel User berdasarkan parameter {id} |
| Url | /users/:id |
| Method | PUT |
| URL Params | id = [integer]  Example  id = 10  urls = /users/10 |
| Data Params | JWT Token  {  "username":"superuser",  "password":"passwordraw",  } |
| Success Response | HTTP Response Code: 200  Response :  {  "id": "1",  "username":"superuser",  "createdDate":"1576204776"  "modifiedDate": "1576204776",  } |
| Error Response | HTTP Response Code: 422  Response : {error:”Invalid Username”}  HTTP Response Code: 422  Response : {error:”Password is to short”} |
| Sample Call | $.ajax({  url: "/users/10",  dataType: "json",  data : {  u: {  "username":"superuser",  "password":" passwordraw",  }  },  type : "PUT",  success : function(r) {  console.log(r);  }  }); |
| Notes | Pada proses PUT ke url /users dilakukan validasi input dengan rule sebagai berikut :   1. Username : required, minimal 4 huruf alphanumerik 2. Password : required, minimal 6 karakter maksimal 8 karakter alhpanumerik   Jika data yang dikirim lolos validasi maka backend melakukan   1. Enkripsi input password. 2. Membuat modified\_date sesuai waktu pengubahan data. 3. Mengupdate data pada database sesuai id data yang di ubah. 4. Mengembalikan data berformat json sesuai pada Success Response. 5. Created\_date pada saat edit data tidak diubah. |

|  |  |
| --- | --- |
| Judul | Menghapus satu baris data pada tabel User berdasarkan parameter {id} |
| Url | /users/:id |
| Method | DELETE |
| URL Params | id = [integer]  Example  id = 10  urls = /users/10 |
| Data Params | JWT Token |
| Success Response | HTTP Response Code: 200  Response :  {  "success":"Data Deleted Succesfully"  } |
| Error Response | HTTP Response Code: 404  Response : {error: "no user found"} |
| Sample Call | $.ajax({  url: "/users/10",  dataType: "json",  type : "DELETE",  success : function(r) {  console.log(r);  }  }); |
| Notes | Proses delete dilakukan dengan melakukan pengecekan input id pada url /users/:id ex /users/10. Jika terdapat data dengan id tersebut maka dilanjutkan dengan proses delet jika tidak proses dihentikan dan memberikan response sesuai Error Response |

|  |  |
| --- | --- |
| Judul | Mengembalikan daftar seluruh player pada tabel Player |
| Url | /players |
| Method | GET |
| URL Params | None |
| Data Params | JWT Token |
| Success Response | HTTP Response Code: 200  Response :  {  data: [  {  "id": 1,  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  "created\_by": 1,  "modified\_by": ,  "created\_at": 1576204776,  "modified\_at": 1576204776,  },  {  "id": 1,  "posisi":"CF" ,  "nama": "Cristiano Ronaldo ",  "nomor\_punggung": 7,  "createdBy ": 1,  "modifiedBy": ,  "createdDate": 1576204776,  "modifiedDate": 1576204776,  },  ]  } |
| Error Response | HTTP Response Code : 404  Response :  {error : “Player Not Found”} |
| Sample Call | $.ajax({  url: "/players",  dataType: "json",  type : "GET",  success : function(r) {  console.log(r);  }  }); |
| Notes | Proses Get Player dapat dilakukan public tanpa menyertakan JWT |

|  |  |
| --- | --- |
| Judul | Membuat data player baru pada tabel player |
| Url | /player |
| Method | POST |
| URL Params | None |
| Data Params | JWT Token  {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  "createdBy":1  } |
| Success Response | HTTP Response Code:: 200  Response :  {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  } |
| Error Response | HTTP Response Code:: 422  Response : {error:”invalid number”} |
| Sample Call | $.ajax({  url: "/players",  dataType: "json",  data : {  u: {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  }  },  type : "POST",  success : function(r) {  console.log(r);  }  }); |
| Notes | - |

|  |  |
| --- | --- |
| Judul | Mengembalikan satu baris data pada tabel Player berdasarkan parameter {id} |
| Url | /players/:id |
| Method | GET |
| URL Params | id = [integer]  Example  id = 10  urls = /users/10 |
| Data Params | JWT Token |
| Success Response | HTTP Response Code:: 200  Response :  {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  "createdBy": 1,  "modifiedBy": null,  "createdDate": 1576204776,  "modifiedDate": 1576204776,  } |
| Error Response | HTTP Response Code:: 422  Response : {error:”player not found”} |
| Sample Call | $.ajax({  url: "/players/1",  dataType: "json",  type : "GET",  success : function(r) {  console.log(r);  }  }); |
| Notes | - |

|  |  |
| --- | --- |
| Judul | Mengubah satu baris data pada tabel Player berdasarkan parameter {id} |
| Url | /players/:id |
| Method | PUT |
| URL Params | id = [integer]  Example  id = 10  urls = /players/10 |
| Data Params | JWT Token  {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  "modified\_by" : 1,  } |
| Success Response | HTTP Response Code:: 200  Response :  {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  "createdBy": 1,  "modifiedBy": 1,  "createdDate": 1576204776,  "modifiedDate": 1576204776,  } |
| Error Response | HTTP Response Code:: 422  Response : {error:”Invalid number”} |
| Sample Call | $.ajax({  url: "/players/10",  dataType: "json",  data : {  u: {  "posisi":"CF" ,  "nama": "Lionel Messi",  "nomor\_punggung": 10,  }  },  type : "PUT",  success : function(r) {  console.log(r);  }  }); |
| Notes | - |

|  |  |
| --- | --- |
| Judul | Menghapus satu baris data pada tabel Player berdasarkan parameter {id} |
| Url | /players/:id |
| Method | DELETE |
| URL Params | id = [integer]  Example  id = 10  urls = /players/10 |
| Data Params | JWT Token |
| Success Response | HTTP Response Code:: 200  Response :  {  "success":"Data Deleted Succesfully"  } |
| Error Response | HTTP Response Code:: 404  Response : {error: "no player found"} |
| Sample Call | $.ajax({  url: "/players/10",  dataType: "json",  type : "DELETE",  success : function(r) {  console.log(r);  }  }); |
| Notes | - |

1. Ujilah backend yang anda buat menggunakan rest client postman atau insomnia.
2. Buatlah sebuah tampilan frontend menggunakan library frontend yang anda kuasai (react, angular, vue, jquery atau vanilla javascript) dengan fitur sebagai berikut :
   1. Login
   2. Lihat profil user
   3. Lihat Daftar Pemain
   4. Tambah Daftar Pemain
   5. Edit Pemain
   6. Hapus Data Pemain