

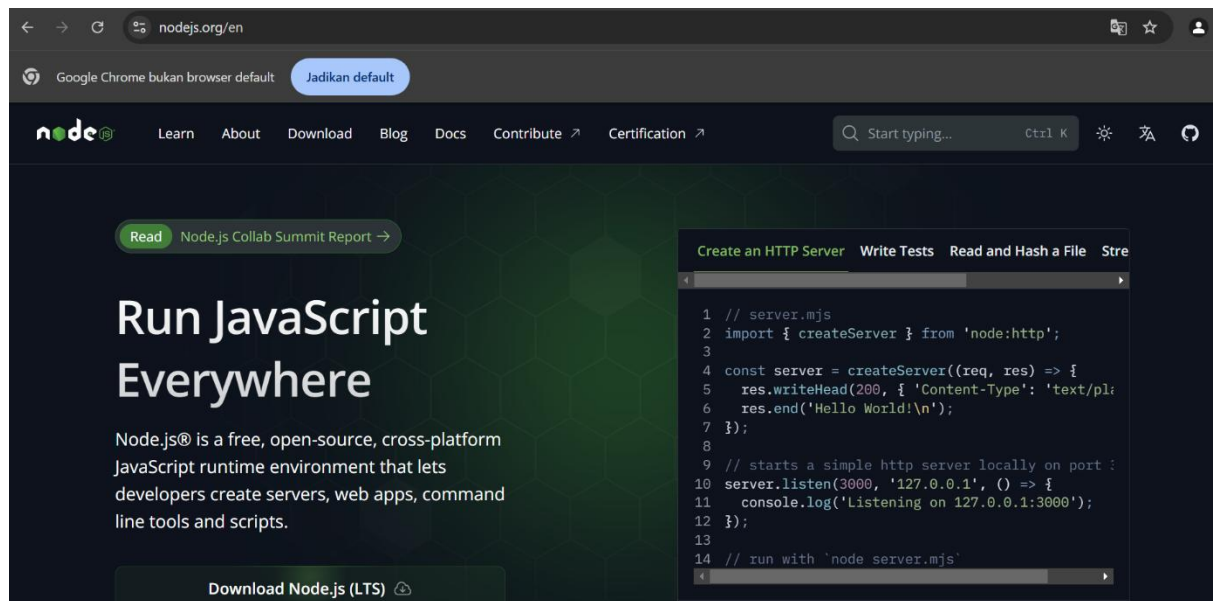
Nama : Mae Vanessa

Nim : 243026010

Prodi : S1 Sistem Informasi

LANGKAH – LANGKAH MEMBUAT API MOBILE LEGEND

1. Pada tahap pertama, perlu mengunduh aplikasi node.js dengan mengunjungi situs resmi Node.js (<https://nodejs.org/>)
2. Menjalankan installer dan mengikuti petunjuk yang ada di layar



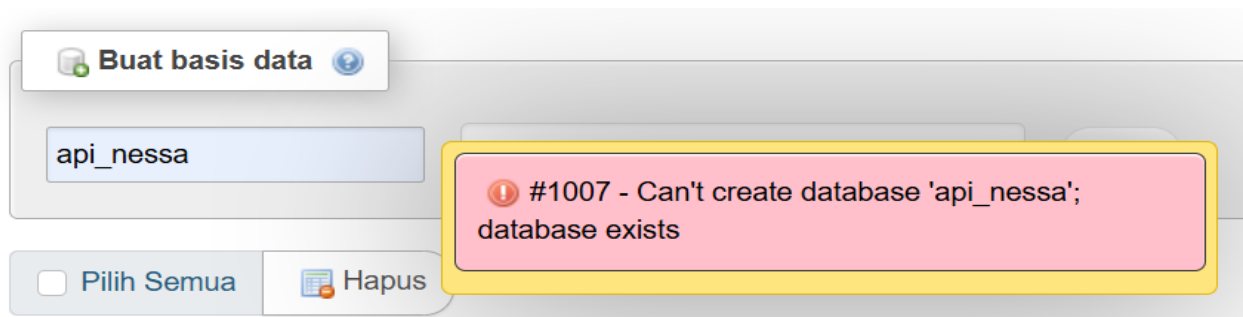
3. Setelah itu membuat folder API untuk proyek
4. Lalu mengecek apakah node & npm sudah terinstall dengan menggunakan `node -v` & `npm -v` di terminal

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMITS

- PS C:\Users\Dell Latitude 7390\Documents\api_vanessa_p1> `node -v`
v23.3.0
- PS C:\Users\Dell Latitude 7390\Documents\api_vanessa_p1> `npm -v`

5. Jika sudah, membuat database dan table di phpMYAdmin

Disini saya membuat database api_nessa



Lalu membuat table & kolom

Create new table

Nama tabel

posts

Jumlah kolom

25

Buat

Mengisi Nama, Jenis, Panjang, Bawaan, Tak Ternilai, dan A_I pada setiap kolom

Server: 127.0.0.1 > Database: api_nessa > Tabel: posts

Jelajahi

Struktur

SQL

Cari

Tambahkan

Ekspor

Impor

Hak Akses

Operasi

Trigger

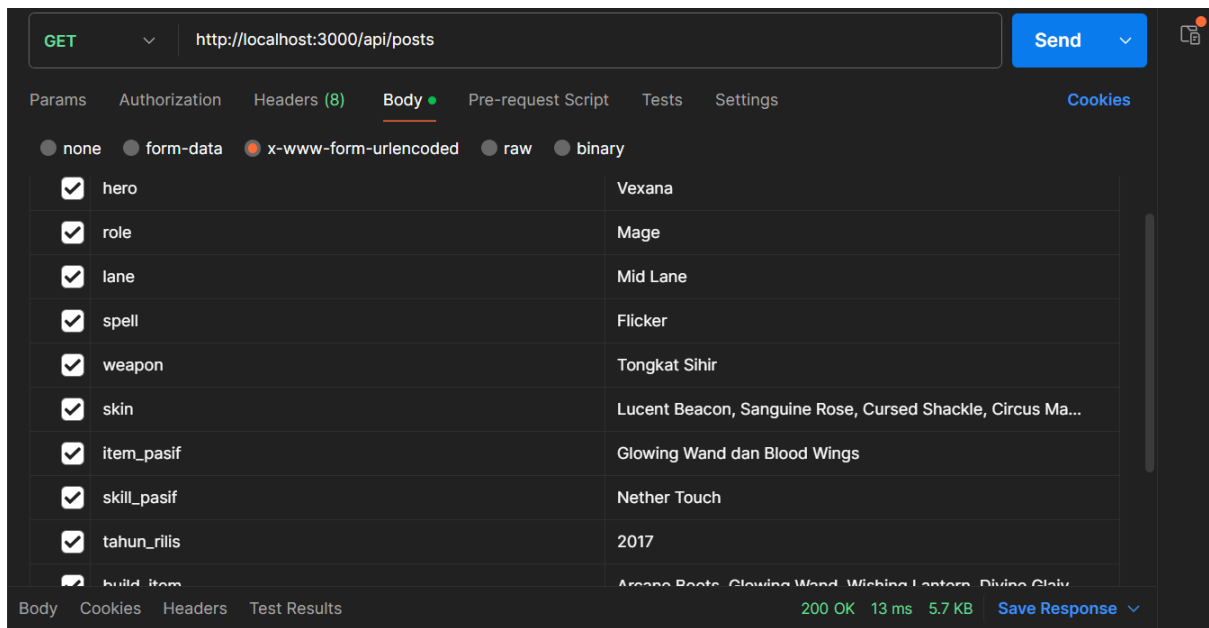
Struktur

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	A_I
id	INT	11	Tidak ada			<input type="checkbox"/>	<input checked="" type="checkbox"/>
hero	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
role	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
lane	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
spell	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
weapon	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
skin	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
item_pasif	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
skill_pasif	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Konsol	VARCHAR	255	NULL	utf8mb4_general_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>

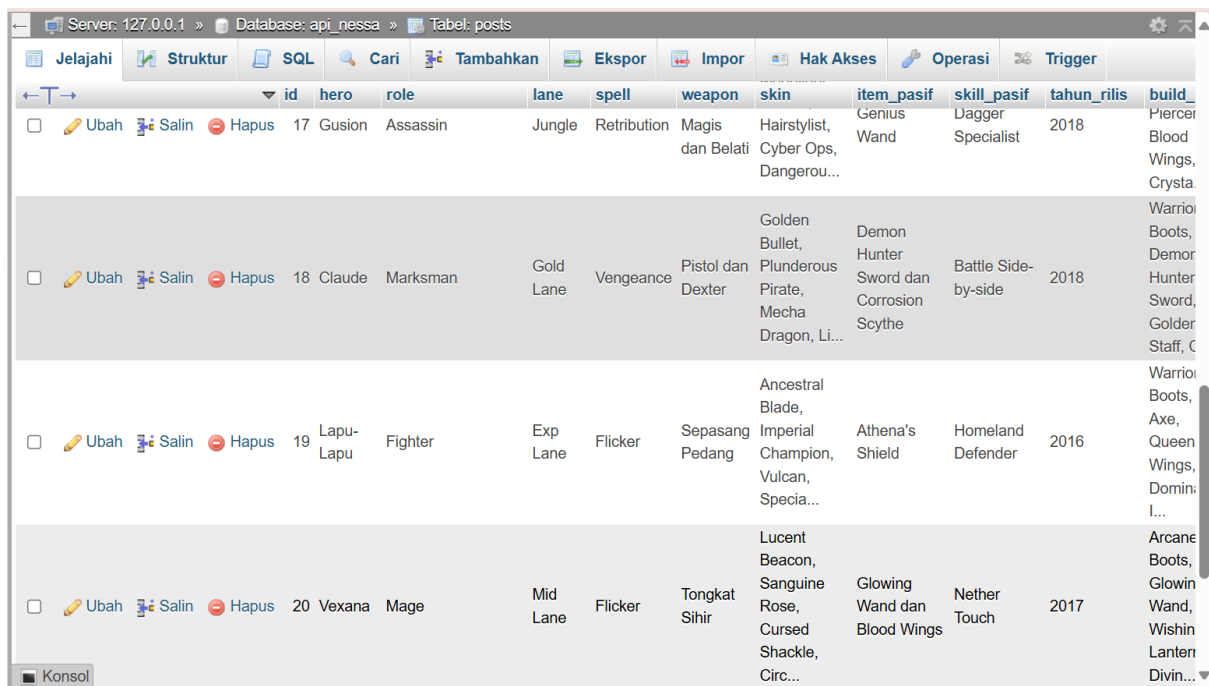
Jika sudah, maka tampilannya akan seperti ini :

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/> 2	hero	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 3	role	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 4	lane	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 5	spell	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 6	weapon	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 7	skin	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 8	item_pasif	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 9	skill_pasif	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 10	tahun_rilis	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 11	build_item	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 12	hero_counter	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 13	combo_skill	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya

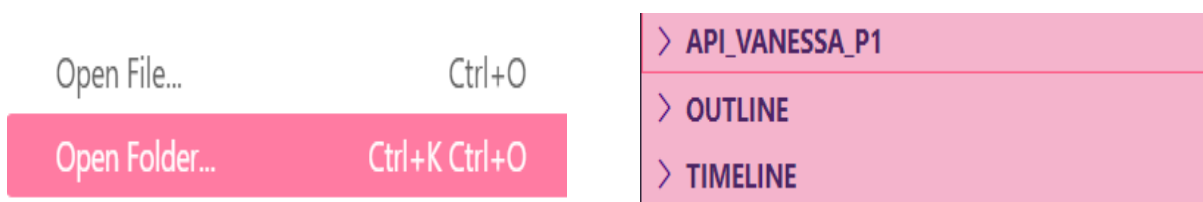
Lalu menginput data lewat Postman one by one




Setelah terinput, maka akan muncul tampilan seperti ini di phpMyAdmin



6. Setelah itu membuka folder API yang sudah dibuat sebelumnya ke VSCode



7. Jika sudah, perlu mengunduh package.json yang berisi body-parser, express, express-validator, mysql, nodemon dari terminal vscode agar codingan tidak terjadi error



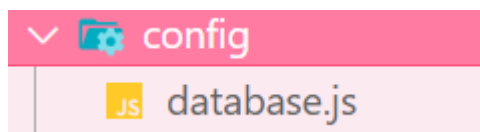
```
8 },
9 "author": "vanessa",
10 "license": "ISC",
11 "dependencies": {
12   "api_nessa": "file:",
13   "api_vanessa_p1": "file:",
14   "body-parser": "^1.20.3",
15   "express": "^4.21.2",
16   "express-validator": "^7.2.0",
17   "mysql": "2.18.1"
18 },
19 "devDependencies": {
20   "nodemon": "^3.1.7"
```

8. Selanjutnya membuat file index.js di VSCode



```
index.js
```

9. Pada tahap selanjutnya, membuat folder yang bernama config yang berisi file database.js



```
config
  database.js
```

10. Lalu membuat folder yang bernama routes yang berisi file posts.js



```
routes
  posts.js
```

11. Mengetikkan code file index.js untuk menampilkan halaman aw



```
1 const express = require('express');
2 const app = express();
3 const port = 3000;
4 const bodyparser = require('body-parser');
5
6 app.get('/', (req, res) => {
7   res.send('Welcome to Mobile Legends');
8 });
9
10 app.use(bodyparser.urlencoded({ extended: false }));
11 app.use(bodyparser.json());
12
13 // import route posts
14 const postsRouter = require('./routes/posts.js');
15 app.use('/api/posts', postsRouter);
16
17 // akses halaman website
18 app.listen(port, () => {
19   console.log('this application running in http://localhost:${port}');
20 });
```

12. Mengetikkan code file database.js untuk menyambungkan dengan mysql

```
database.js ×
config > database.js > connect
You, 5 days ago | 1 author (You)
1 let mysql = require('mysql') 272k (gzipped: 88.8k)
2
3 let connect = mysql.createConnection({
4   host: 'localhost',
5   user: 'root', You, 5 days ago • first commit
6   password: '',
7   database: 'api_nessa'
8 });
9
10 connect.connect(function(error) {
11   if (!error) {
12     console.log( error);
13   } else {
14     console.log('Connection Success');
15   }
16 });
17
18 module.exports = connect;
```

13. Mengetikkan code file posts.js untuk menampilkan data, menambahkan data, mengupdate data, dan menghapus data

Query untuk Menambahkan Data :

```
//Query untuk menambahkan data baru
connect.query('INSERT INTO posts SET?', formData, function(err,rows){
  //jika data tidak dapat disimpan, maka akan menampilkan status 500 error
  if(err){
    return res.status(500).json({
      status:false,
      message:'Error, Try again later'
    })
  }else{ You, 5 days ago • first commit
    //jika data berhasil disimpan, akan menampilkan 201 dan pesan sukses
    return res.status(201).json({
      status:true,
      message:'Data input Successfully'
    })
  }
})
})
})
```

Query untuk Menampilkan Data :

```
//Query untuk menampilkan data
router.get('/:id', function(req,res){
  let id = req.params.id

  connect.query(`SELECT * FROM posts WHERE ID=${id}`,
    function(error,rows){
      if(error){
        // jika terjadi error, maka akan menampilkan status 500 error
        return res.status(500).json({
          status:false,
          message: 'Server Error!!'
        })
      }
      // jika data tidak ditemukan, akan menampilkan status 404 error
      if(rows.length <= 0){
        return res.status(404).json({
          status:false,
          message: 'Data is missing'
        })
      } else {
        // jika data berhasil diambil, akan menampilkan status 200
        return res.status(200).json({
          status: true,
          message: 'Display data according to id',
          data: rows[0]
        })
      }
    }
  )
})
```

Query untuk Update Data :

```
//Query untuk update data
connect.query(`UPDATE posts set ? WHERE id=${id}`, formData,
  function(err,rows){
    if(err){
      //jika data tidak dapat di update, akan menampilkan 500 error
      return res.status(500).json({
        status: false,
        message: 'Interval server error'
      })
    } else {
      //jika data berhasil di update, akan menampilkan 200 dan pesan sukses
      return res.status(200).json({
        status: true,
        message: 'Update data success'
      })
    }
  }
)
```

Query untuk Menghapus Data :

```
//Query untuk menghapus data berdasarkan id
router.delete('/delete/:id',
function(req,res){
  let id = req.params.id
  connect.query(`DELETE FROM posts WHERE id = ${id}`,
    function(error,rows){
      if(error){
        //jika data tidak dapat dihapus, akan menampilkan 500 error
        return res.status(500).json({
          status:false,
          message: 'Server Error!!'
        })
      }else {
        //jika data berhasil dihapus, akan menampilkan 200 dan pesan sukses
        return res.status(200).json({
          status: true,
          message: 'Data has been deleted',
        })
      }
    })
})
})
```

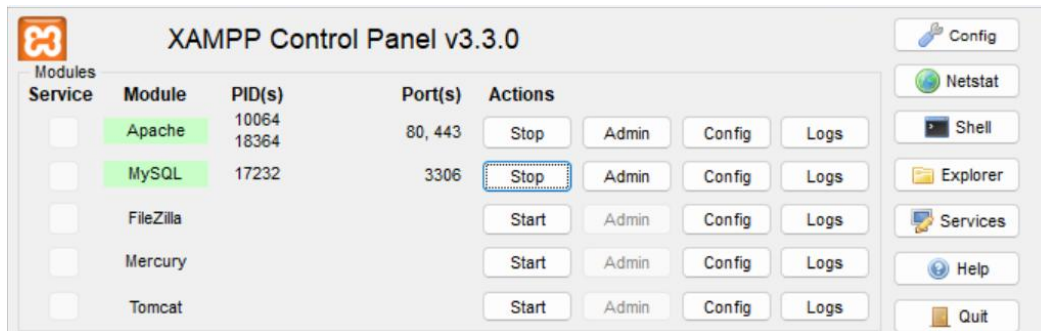
14. Selanjutnya membuat array sesuai table yang telah dibuat di phpMYAdmin

```
router.post('/store',[
  //validasi data input
  body('hero').notEmpty(),
  body('role').notEmpty(),
  body('lane').notEmpty(),
  body('spell').notEmpty(),
  body('weapon').notEmpty(),
  body('skin').notEmpty(),
  body('item_pasif').notEmpty(),
  body('skill_pasif').notEmpty(),
  body('tahun_rilis').notEmpty(),
  body('build_item').notEmpty(),
  body('hero_counter').notEmpty(),
  body('combo_skill').notEmpty(),
],(req,res)=> {
  //cek validasi data input
  const errors = validationResult(req)
  if(!errors.isEmpty()){
    //jika terjadi error validasi, akan menampilkan status 422 error
    return res.status(422).json({
      errors:errors.array()
    })
  }
})
```

//definisi from data

```
let formData = {
  hero: req.body.hero,
  role: req.body.role,
  lane: req.body.lane,
  spell: req.body.spell,
  weapon: req.body.weapon,
  skin: req.body.skin,
  item_pasif: req.body.item_pasif,
  skill_pasif: req.body.skill_pasif,
  tahun_rilis: req.body.tahun_rilis,
  build_item: req.body.build_item,
  hero_counter: req.body.hero_counter,
  combo_skill: req.body.combo_skill,
}
```

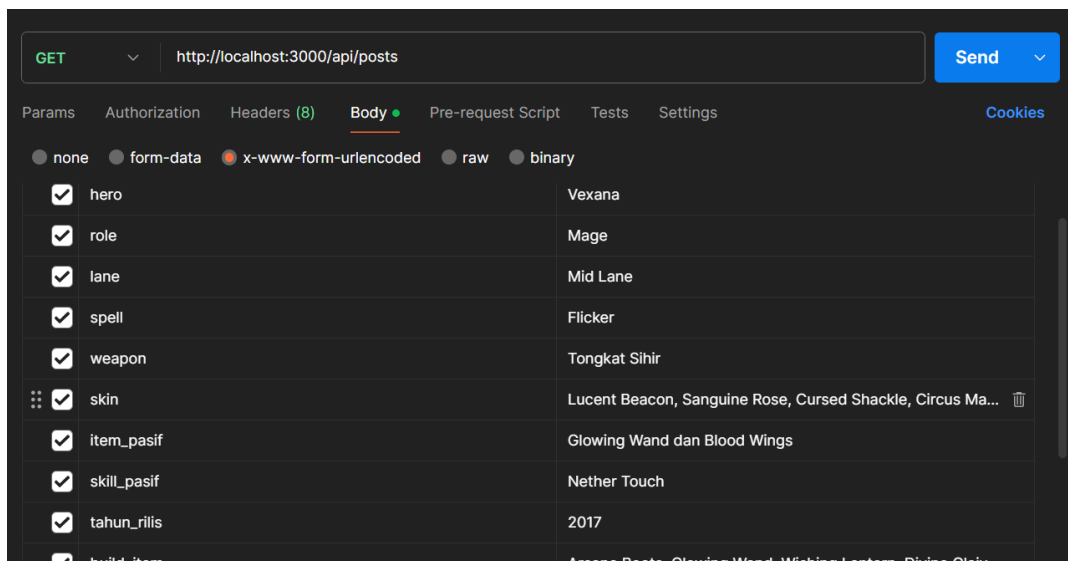
15. Wajib menyalakan Xampp dengan kondisi Apache & MySQL menyala

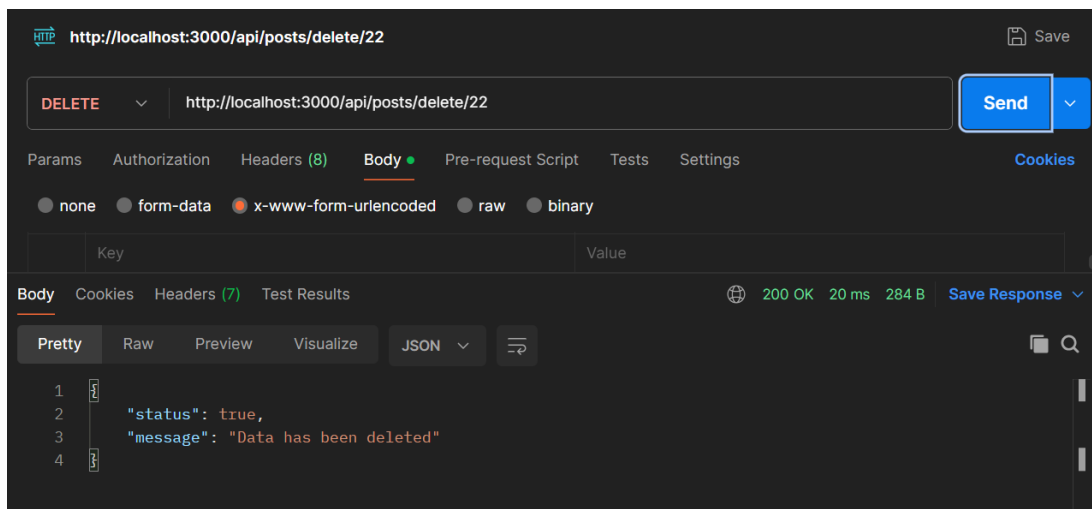
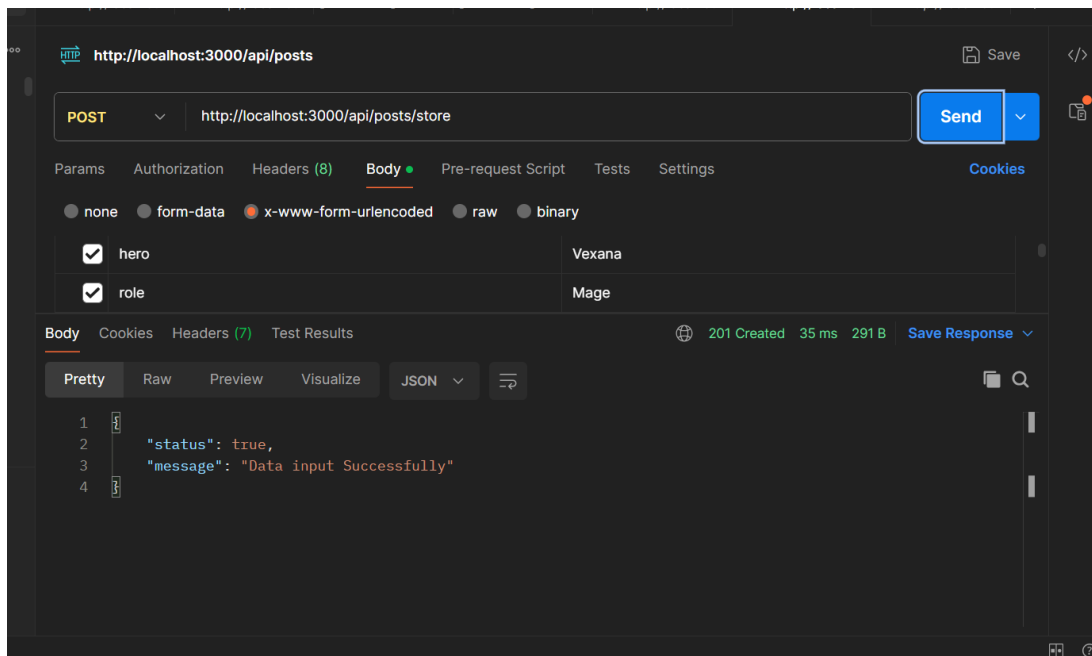
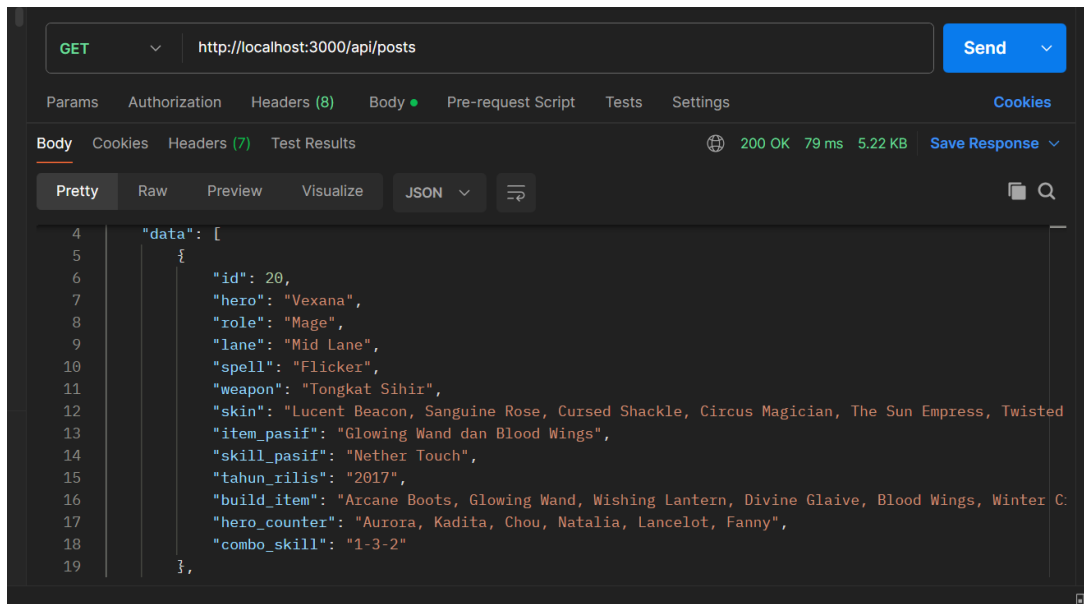


16. Selanjutnya kita mencoba run di terminal dengan command nodemon npm start

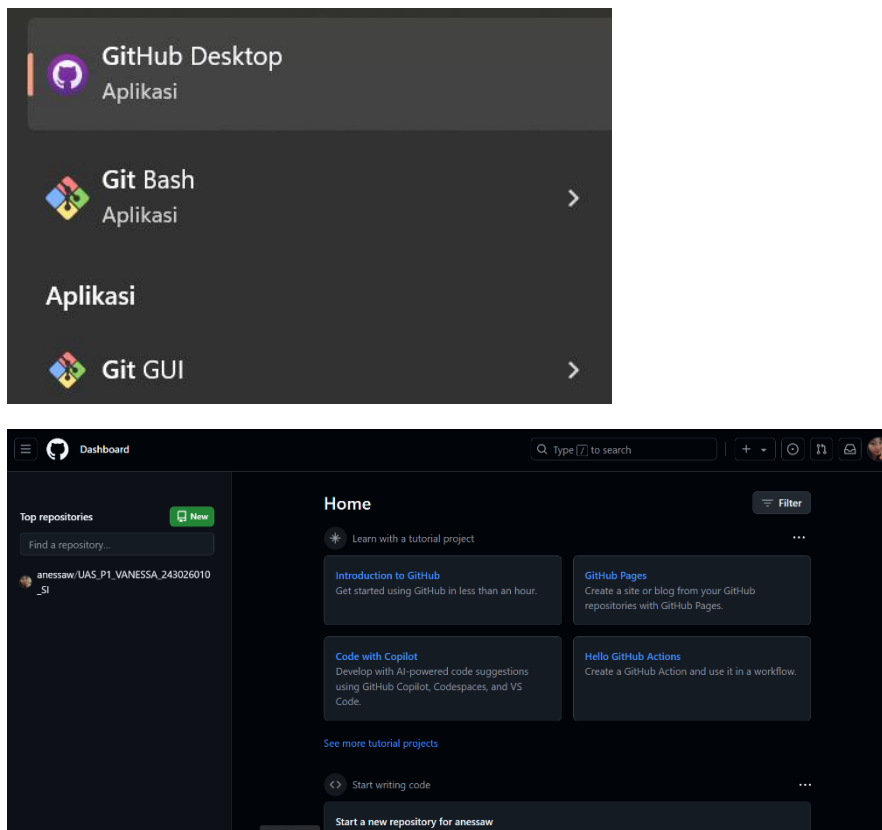
```
PS C:\Users\Dell Latitude 7390\Documents\api_vanessa_p1> nodemon npm start
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node npm start index.js`
this application running in http://localhost:3000
Connection Success
```

17. Selanjutnya membuka aplikasi postman mencoba menampilkan, menambah, mengupdate, dan menghapus data di postman

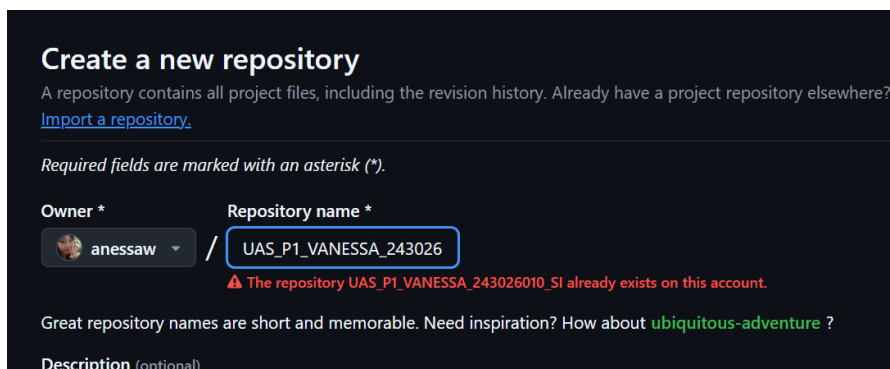




18. Selanjutnya kita mengunduh git dan login ke github



19. Jika sudah, membuat repositori baru



20. Selanjutnya melakukan commit lalu melakukan push file ke github

The image shows the GitHub repository page for 'anessaw' with the repository name 'Delete LANGKAH LANGKAH MEMBUAT API MOBILE LEGEND.pdf'. The page shows the commit history with 9 commits. The commit history table is as follows:

File	Commit Message	Time
config	first commit	5 days ago
node_modules	first commit	5 days ago
routes	first commit	5 days ago
README.md	Update README.md	5 days ago
api_nessa.sql	database	4 days ago
index.js	first commit	5 days ago
package-lock.json	first commit	5 days ago
package.json	first commit	5 days ago

21. Terakhir mengecek codingan yang sudah di push ke github apakah ada terjadi error atau tidak