konsep: buat app yg bisa buat app lain, framework, mirip wordpres tp beda, ini bisa didownload dan ngga berbayar, mirip framework pada umumnya, dan compatible dg framework lainnya.  
jadi dia ada tampilan:  
  
Fungsi :  
-dia buat tampilan  
-dia buat db  
-dia buat script  
-dia buat css  
-dia buat api  
  
jadi dia buat front sama back end file dipisah, tp semua termanage dlm satu wadah yg singkron  
-bisa generate dlm berbagai baha pemprograman nanti, tp sekarang javascript aja, nanti ada project buat translate semua file kebhasa pemprogramanan lain  
urutan kerja:FrontEnd  
create box , kasi warna, bisa digeser2, bisa diisi cssnya  
trus di tempatkan posisinya, trus bisa diisi text dsb, dan keterkaitan datanya misal dikasi class  
  
  
nanti di folder baru  
nah trus disave, pas disave dia bakal generate html sama cssnya, jsnya.(JS buat style sama data dipisah)  
  
jadi dia di 1 tempat tampilan yg sama  
  
trus input nama API  
hubungin API kebox yg diinginkan  
  
BackENd   
akan meminta nama table dsb yg terhubung ke API  
trus generate, dia run command npm init

dia kasi display API apa aja

trus minta generate front/ back end

trus dia updateto dist

nnti dia copy semua ke distjadi 1 folder aplikasi jadi

## Tampilan:

aplikasi ini memuat dalam layer

jadi yg dibuat nanti di bawah aplikasi ini, trus bisa di hide tombol2nya

-tombol create box, delete box, click/select box u/edit,

-add html component:

text, add image

-add input, add link

-advanced(ini bisa edit html, css, js u/ grontend) dan setting2 untuk back end

apa kita buta 2 mode tampilan

pas buat front sendiri, uat backend sendiri

## ARSITEKTUR APLIKASI(yg digenerate)

APP

-FrontEnd

--nodeModules dkk{package.json, package.lock.json, .env}

--index.html

--src{

---api{}

---assets{}

---theme:{

----style

----js

}

---layout{}

---routes{MainRoutes.js, ProtectedRoutes.js, AuthenticationRoutes.js, }

---store{}

---ui-component{}

---utils{}

---views{}

App.js

Config.js

index.js

}

-BackEnd

-- nodeModules dkk{package.json, package.lock.json, .env}

--server.js

--src

---config{database.js}

---controllers( yg cotroller models sama routes isi file sesuai table, copy paste template nanti)

---models

---routes

---services

---utils{auth.js, itemValidation.js, constant.js, response.js, userValidation.js}

dia bikin konsep seeding jg buat input datanya