3rd YEAR TEST-10 TOTAL TIME \rightarrow 1.5hrs

Below is the representation of the task that you need to perform.

- → let's say your University roll number is 291029063
- \rightarrow your set = (University roll number) % 15 = 291029063%15 = 8
- \rightarrow so your set number is 8
- \rightarrow so you have to follow the below question set-8

Question Set - 0

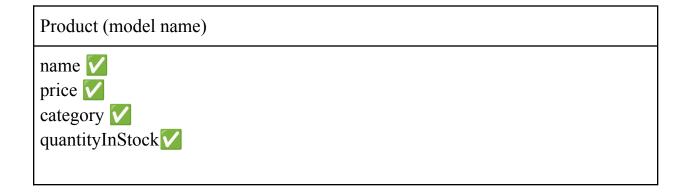
- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed) '/person/:id', '/person/:id/edit', '/person' and '/person/:id/delete'. (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only person name cannot be changed, rest of everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10.No need to have any dummy data.
- 11Do not share the node_modules and package-lock.json.

Person (model name) name age email D.O.B

personSchema (schema name)

```
name: → type , mandatory , trim
age: → type , mandatory , trim
email: → type , mandatory , trim
D.O.B: → type , mandatory
```

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/product/:id', '/product/:id/edit', '/product' and '/product/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only productName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.



```
productSchema (schema name)

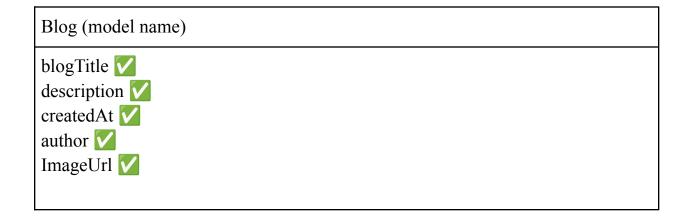
name: → type , mandatory , trim

price: → type , mandatory , trim

category: → type , mandatory , trim

quantityInStock: → type , mandatory
```

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/blog/:id' , '/blog/:id/edit' , '/blogs and '/blog/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only blogTitle cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes → make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.



blogSchema (schema name)

blogTitle: \rightarrow type, mandatory, trim description: \rightarrow type, mandatory, trim createdAt: \rightarrow type, mandatory, trim

author: \rightarrow type, mandatory

 $ImageUrl: \rightarrow type$, mandatory, trim

Question Set - 3

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/song/:id', '/song/:id/edit', '/songs' and '/song/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only songTitle cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public → style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Song (model name) songTitle singer genre releaseDate songImageUrl

songSchema (schema name)

```
songTitle: → type, mandatory, trim
singer: → type, mandatory, trim
genre: → type, mandatory, trim
```

releaseDate: \rightarrow type , mandatory , Date songImageUrl: \rightarrow type , mandatory

Question Set - 4

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/contact/:id', '/contact/:id/edit', '/contacts and '/contact/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only contactName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Contact (model name) contactName phoneNo email imageUrl

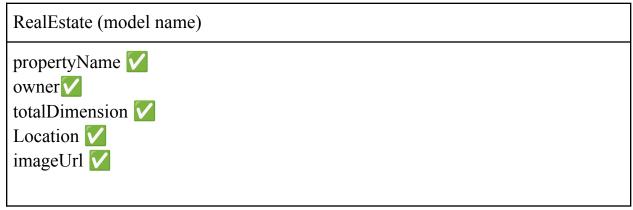
```
contactSchema (schema name)

contactName: → type, mandatory, trim
phoneNo: → type, mandatory, trim
email: → type, mandatory, trim
imageUrl: → type, mandatory
```

- 1. Create a schema-model according to the following diagrams (box shown below).
- $2.\ Perform\ READ\ ,\ EDIT\ ,\ SHOW,\ DELETE\ functionality.\ (use\ forms\ wisely\ if\ needed)\ '/realEstate/:id',\ '/realEstate/:id/edit',\ '/realEstates\ and$

'/realEstate/:id/delete'(use proper routing convention)

- 3. Form should have inputs according to the table below.
- 4. # Only propertyName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.



realestateSchema (schema name)

propertyName: \rightarrow type, mandatory, trim

owner: → type, mandatory, trim

total Dimension: \rightarrow type, mandatory, trim

Location : \rightarrow type , mandatory imageUrl : \rightarrow type , mandatory

Question Set - 6

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/event/:id', '/event/:id/edit', '/event' and '/event/:id/delete'(use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only eventName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Event (model name) name location date attendees

eventSchema (schema name)

name: → type, mandatory, trim location: → type, mandatory, trim date: → type, mandatory, trim attendees: → type, mandatory

Question Set - 7

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed)'/car/:id', '/car/:id/edit', '/car' and '/car/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only car company cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public → style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10.No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Car (model name)



carSchema (schema name)

company: → type, mandatory, trim model: → type, mandatory, trim year: → type, mandatory, trim mileage: → type, mandatory

Question Set - 8

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed) '/music-album/:id', '/music-album/:id/edit', '/music-album' and '/music-album/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only musicTitle cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public → style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Music (model name)

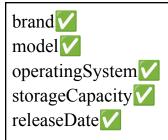
```
musicTitle 
artist 
genre 
releaseYear
```

```
musicSchema (schema name)
```

```
musicTitle: → type, mandatory, trim
artist: → type, mandatory, trim
genre: → type, mandatory, trim
releaseYear: → type, mandatory
```

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed) '/mobile-phone/:id', '/mobile-phone/:id/edit', '/mobile-phone' and '/mobile-phone/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only brand cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Mobile (model name)



mobileSchema (schema name)

brand: \rightarrow type, mandatory, trim model: \rightarrow type, mandatory, trim

operating System: \rightarrow type, mandatory, trim

storageCapacity: → type , mandatory releaseDate: → type , mandatory , trim

Question Set - 10

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed) '/university/:id', '/university/:id/edit', '/universities' and '/university/:id/delete'(use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only universityName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node modules and package-lock.json.

University (model name)

```
universityName 
affiliateBy 
location 
numberOfCourses 
ImagesUrl
```

```
mobileSchema (schema name)
```

```
universityName: → type, mandatory, trim affiliateBy: → type, mandatory, trim location: → type, mandatory, trim numberOfCourses: → type, mandatory
```

ImagesUrl: \rightarrow type, mandatory, trim

Question Set - 11

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed) '/restaurant/:id', '/restaurant/:id/edit','/restaurants' and '/restaurant/:id/delete'(use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only restaurantName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Restaurant (model name)

```
restaurantName Cuisine Cuisine Chef phoneNo ImageUrl
```

restaurantSchema (schema name)

restaurantName : → type , mandatory , trim

Cuisine: \rightarrow type, mandatory, trim Chef: \rightarrow type, mandatory, trim

 $phoneNo: \rightarrow type \ , \, mandatory$

ImageUrl: \rightarrow type, mandatory, trim

Question Set - 12

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/airline/:id', '/airline/:id/edit', '/airlines' and '/airline/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only companyName be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Airline (model name)

```
companyName 
owner 
numberOfPlane 
planeCapacity 
Model
```

```
airlineSchema (schema name)
```

```
companyName: \rightarrow type \ , \ mandatory \ , \ trim
```

owner: \rightarrow type, mandatory, trim

 $numberOfPlane: \rightarrow type \ , \ mandatory \ , \ trim$

planeCapacity: → type, mandatory Model: → type, mandatory, trim

Question Set - 13

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ, EDIT, SHOW, DELETE functionality. (use forms wisely if needed) '/hotel/:id', '/hotel/:id/edit', '/hotels' and '/hotel/:id/delete'(use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only hotelName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node_modules and package-lock.json.

Hotel (model name)

```
hotelName  rating  numberOfRoom  Location  roomImageUrl
```

hotelSchema (schema name)

hotelName: \rightarrow type, mandatory, trim

rating: \rightarrow type, mandatory, trim

 $numberOfRoom : \rightarrow type$, mandatory, trim

Location: \rightarrow type, mandatory

roomImageUrl: → type, mandatory, trim

Question Set - 14

- 1. Create a schema-model according to the following diagrams (box shown below).
- 2. Perform READ , EDIT , SHOW, DELETE functionality. (use forms wisely if needed) '/zoo/:id', '/zoo/:id/edit', '/zoo' and '/zoo/:id/delete' (use proper routing convention)
- 3. Form should have inputs according to the table below.
- 4. # Only zooName cannot be changed, rest everything can change.
- 5. Follow proper folder structures
- 6. Public \rightarrow style according to the template shown (just for reference)
- 7. Views \rightarrow create proper template files
- 8. Models → make different schema
- 9. Routes \rightarrow make different routing files
- 10. No need to have any dummy data.
- 11.Do not share the node modules and package-lock.json.

Zoo (model name)

zooName \(\sqrt{}\)
numberOfAnimal \(\sqrt{}\)
location \(\sqrt{}\)
openingDuration \(\sqrt{}\)
zooImageUrl \(\sqrt{}\)

zoomSchema (schema name)

zooName: \rightarrow type , mandatory , trim

 $numberOfAnimal: \rightarrow type \;, \, mandatory \;, \, trim$

location: \rightarrow type , mandatory , trim openingDuration: \rightarrow type , mandatory zooImageUrl: \rightarrow type , mandatory , trim