

# userModel

```
firstName // Type => String , max => 15
lastName  // Type => String , max => 10
headline  // Type => String , max => 50
email     // Type => email , lowercase => true
photo     // Type => String , /AWS\
password  // Type => String , encryption => *sha-256* , salt => 12
gender    // Type => String , enum = ['Male', 'Female', 'Other']
policySigned // Type => Boolean , /This field must be true always, but with consent of user\
subscribedToMailList // Type => Boolean , enum => ['True', 'False'] , default => False
whatAreYouPlanningToDo // Type => String, enum => [ "I'm looking for events to attend",
                                                    'I want to host an event',
                                                    'I want to do both' ]
interests // Type => String , enum => [ 'Technology', 'Education', 'Lifestyle',
                                         'Professional Development', 'Arts and crafts',
                                         'Business & Entrepreneurship', 'Job Search',
                                         'Entertainment', 'Health' ]
socialMediaHandles // Type => Map , of => String
communities // [Array] // of embedded docs // {communityId: Type => ObjectId, ref => 'Community'
                                                  Access: Type => String, enum => [List of Features] }
registredInEvents // [Array] // of ObjectIds // Type => ObjectId , ref => 'Event'
registrations     // [Array] // of ObjectIds // Type => ObjectId , ref => 'Registration'
reviews           // [Array] // of ObjectIds // Type => ObjectId , ref => 'Review'
```

Continue ...

Continued ...

# userModel

```
queries // [Array] // of ObjectIds // Type => ObjectId , ref => 'Query'  
active // Type => Boolean , default => True , select => False  
passwordChangedAt // Type => Date  
passwordResetToken // Type => String  
passwordResetExpires // Type => Date  
initialisedAt // Type => Date  
lastUpdatedAt // Type => Date  
deletedAt // Type => Date
```

# sponsorModel

```
name // Type => String , max: 15
socialMediaHandles // Type => Map , of => String
logo // Type => String , /AWS\
status // Type => String , enum = ['Emerald', 'Gold', 'Silver', 'Bronze']
initialisedAt // Type => Date
lastUpdatedAt // Type => Date
```

# speakersIdsCommunityWiseModel

```
initialisedAt // Type => Date
```

```
speakersIds // [Array] // of ObjectIds // Type => ObjectId , ref => 'Speaker'
```

# speakerModel

```
firstName // Type => String , maxLength => 15
lastName  // Type => String , maxLength => 10
email     // Type => email , lowercase => true
image     // Type => String , /AWS\
organisation // Type => String , maxLength => 25
designation // Type => String , maxLength => 25
bio       // Type => String , maxLength => 250
sessions  // [Array] // of ObjectIds // Type => ObjectId , ref => 'Session'
eventId   // Type => ObjectId , ref => 'Event'
status    // Type => String , enum = ['Accepted', 'Pending', 'Rejected']
createdAt // Type => Date
```

# sessionModel

```
name // Type => String , maxlength => 25
startDate // Type => Date
startTime // Type => Date
endDate // Type => Date
endTime // Type => Date
host // [Array] // of ObjectIds // Type => ObjectId , ref => 'User'
speaker // [Array] // of ObjectIds // Type => ObjectId , ref => 'Speaker'
agenda // Type => String , maxlength: 50
summary // Type => String , maxlength: 1000
createdAt // Type => Date
```

# reviewsIdsCommunityWiseModel

```
initialisedAt // Type => Date
```

```
speakersIds // [Array] // of ObjectIds // Type => ObjectId , ref => 'Review'
```

# registrationsIdsCommunityWiseModel

```
initalisedAt // Type => Date
```

```
speakersIds // [Array] // of ObjectIds // Type => ObjectId , ref => 'Registration'
```



# queriesIdsCommunityWiseModel

```
initialisedAt // Type => Date
```

```
speakersIds // [Array] // of ObjectIds // Type => ObjectId , ref => 'Query'
```

# eventsIdsCommunityWiseModel

```
initialisedAt // Type => Date
```

```
speakersIds // [Array] // of ObjectIds // Type => ObjectId , ref => 'Event'
```

# reviewModel

```
createdForEvent // Type => ObjectId , ref => 'Event'  
createdByUser  // Type => ObjectId , ref => 'User'  
createdAt     // Type => Date  
userName      // Type => String  
userImg       // Type => String , /AWS\  
rating        // Type => Number , min => 1 , max => 5  
reviewComment // Type => String , maxlength: 300
```

# registrationsModel

```
eventName // Type => String
userName // Type => String
UserImg // Type => String , /AWS\
email // Type => email , lowercase => true
ticketType // Type => String
bookedByUser // Type => ObjectId , ref => 'User'
bookedForEventId // Type => ObjectId , ref => 'Event'
appliedCouponId // Type => ObjectId , ref => 'Coupon'
ticketPDFaws // Type => String , /AWS\
eventByCommunityId // Type => ObjectId , ref => 'Community'
createdAt // Type => Date
```

# emailListModel

```
name // Type => String
email // Type => String , unique => true
addedAtDateAndTime // Type => Date
```

# communityMailListModel

```
name // Type => String
email // Type => String , unique => true
addedAtDateAndTime // Type => Date
```

# queriesModel

```
createdBy // Type => ObjectId , ref => 'User'
createdAt // Type => Date , default => Date.now() // Time at which query was created will be saved
userName // Type => String , (INTERNAL)
userImg   // Type => String , /AWS\
userIs    // Type => String , (INTERNAL) , enum = ['Registered', 'Unregistered']
queryIs   // Type => String , (INTERNAL) , default => 'Unresolved' , enum = ['Resolved', 'Unresolved']
createdForEventId // Type => ObjectId , ref => 'Event'
createdForEventName // Type => String , (INTERNAL)
questionText // Type => String , max => 300
answerText   // Type => String , max => 500
displayOnLandingPage // Type => Boolean , default => False
questionTextModifiedByCommunity // Type => String , max => 300
answeredBy // Type => ObjectId , ref => 'User'
```

# membersActivityModel

```
memberName // Type => ObjectId
memberImg // Type => String , /AWS\
memberId // Type => ObjectId , ref => User
logs // [Array] // of log Documents // Document => { createdAt : Type => Date ,
                                                    logStatement => String }
lastLogCreatedAt // Type => Date
```



# feedbackModel

[illegible]

# eventModel

```
eventName // Type => String , maxlength => 50
shortDescription // Type => String , max => 200
completeDescription // Type => String
startDate // Type => Date
endDate // Type => Date
startTime // Type => Date
endTime // Type => Date
Timezone // Type => String , enum = [list of all 24 Timezones]
visibility // Type => String , enum = ['Public', 'Private', 'Unlisted']
createdAt // Type => Date , default => Date.now()
createdBy // Type => ObjectId , ref => 'Community'
registrationLimit // Type => Number , (INTERNAL) , It will be applied based on plan used by community
host // [Array] // of ObjectIds // Type => ObjectId , ref => 'User'
speaker // [Array] // of ObjectIds // Type => ObjectId , ref => 'Speaker'
session // [Array] // of ObjectIds // Type => ObjectId , ref => 'Session'
booth // [Array] // of ObjectIds // Type => ObjectId , ref => 'Booth'
sponsor // [Array] // of ObjectIds // Type => ObjectId , ref => 'Sponsor'
registration // [Array] // of ObjectIds // Type => ObjectId , ref => 'Registration'
queries // [Array] // of ObjectIds // Type => ObjectId , ref => 'Query'
```

Continue ...

Continued ...

# eventModel

```
review // [Array] // of ObjectIds // Type => ObjectId , ref => 'Review'  
coupon // [Array] // of ObjectIds // Type => ObjectId , ref => 'Coupon'  
status // Type => String, default => active , enum = ['active', 'inactive']
```

# couponModel

```
discountForEventId // Type => ObjectId , ref => 'Event'
discountDescription // Type => String, max => 100
discountPercentage // Type => Number, min => 1 , max => 100
discountCode // Type => String , max => 20
validTillTimeAndDate // Type => Date
createdAt // Type => Date
maxNumOfDiscountsPermitted // Type => Number , /Should be less than max Num of registrations allowed\
active // Type => Boolean , default => true
numOfCouponsUsed // Type => Number , /Should be increased by 1 with every use\
```

# communityModel

```
name // Type => String , ref => 'Event' , unique => True
email // Type => String, lowercase => true
superAdmin // [Array] // of ObjectIds // Type => ObjectId , ref => 'User'
headline // Type => String , max => 100
coverPhoto // Type => String , /AWS\
logo // Type => String , /AWS\
avgRating // Type => Number , min => 1 , max => 5 , (INTERNAL)
policySigned // Type => Boolean , /This field must be true always, but with consent of user\
subscribedToCommunityMailList // Type => Boolean , default => False
createdAt // Type => Date , default => Date.now()
coupons // [Array] // of ObjectIds // Type => ObjectId , ref => 'Coupon'
eventsDocIdCommunityWise // Type => ObjectId , /There is no need for ref here as we will not
                                populate it\
reviewsDocIdCommunityWise // Type => ObjectId , /There is no need for ref here as we will not
                                populate it\
queriesDocIdCommunityWise // Type => ObjectId , /There is no need for ref here as we will not
                                populate it\
registrationsDocIdCommunityWise // Type => ObjectId , /There is no need for ref here as we will not
                                populate it\
speakersDocIdCommunityWise // Type => ObjectId , /There is no need for ref here as we will not
                                populate it\
```

Continue ...

# communityModel

```
followersCount // Type => Number , (INTERNAL) , default => 0
accountsReached // Type => Number , default => 0 , (INTERNAL)
totalRevenue // Type => Number , default => 0 , (INTERNAL)
totalRegistrations // Type => Number , default => 0 , (INTERNAL)
followersGainedThisDay // Type => Number , default => 0 , (INTERNAL)
followersGainedThisMonth // Type => Number , default => 0 , (INTERNAL)
followersGainedThisQuarter // Type => Number , default => 0 , (INTERNAL)
followersGainedThisYear // Type => Number , default => 0 , (INTERNAL)
accountsReachedThisDay // Type => Number , default => 0 , (INTERNAL)
accountsReachedThisMonth // Type => Number , default => 0 , (INTERNAL)
accountsReachedThisQuarter // Type => Number , default => 0 , (INTERNAL)
accountsReachedThisYear // Type => Number , default => 0 , (INTERNAL)
revenueThisDay // Type => Number , default => 0 , (INTERNAL)
revenueThisMonth // Type => Number , default => 0 , (INTERNAL)
revenueThisQuarter // Type => Number , default => 0 , (INTERNAL)
revenueThisYear // Type => Number , default => 0 , (INTERNAL)
registrationsThisDay // Type => Number , default => 0 , (INTERNAL)
registrationsThisMonth // Type => Number , default => 0 , (INTERNAL)
registrationsThisQuarter // Type => Number , default => 0 , (INTERNAL)
```

Continued ...

# communityModel

```
registrationsThisYear // Type => Number , default => 0 , (INTERNAL)
```

# communityActivityModel

```
communityName // Type => String , (INTERNAL)
```

```
communityId // Type => ObjectId , ref => 'Community'
```

```
logs // [Array] // of Documents // Document : { 1) name : Type => String,
                                                    2) photo : Type => String , /AWS\
                                                    3) createdAt : Type => Date
                                                    4) logStatement : Type => String , (INTERNAL)
                                                    }
```

```
lastLogCreatedAt // Type => Date
```



# boothModel

```
name // Type => String
shortDescription // Type => String , max: 500
completeDescription // Type => String , max : 5000
boothLogo // Type => String , /AWS\
boothPoster // Type => String , /AWS\
socialMediaHandles // Type => Map , of => String

tags // [Array] // of Strings // maxLengthOfArray : 10
```

# billingModel

```
// TODO I have to create this model
```