

Repository: [tulsyanshubham/TipyDo-Backend](#)

## ■ Repository Files

Name	Size (bytes)	Path
db.js	411	db.js
index.js	12183	index.js
fetchuser.js	577	middleware/fetchuser.js
count.js	364	modals/count.js
employee.js	1826	modals/employee.js
manager.js	1613	modals/manager.js
session.js	593	modals/session.js
package.json	485	package.json

## ■ Microservice Splits

Original File	Split Files
db.js	config/db_config.js, connections/mongodb_connection.js, services/db_services.js
index.js	config/env.py, api/app.py, services/auth.py, services/email.py
middleware/fetchuser.js	services/auth/token_handler.py, services/auth/user_auth.py
modals/count.js	services/db/count/schemas.py, services/db/count/management.py
modals/employee.js	models/employee.schema.js, services/staff.management.js
modals/manager.js	models/manager/schema.py, services/auth/password_hasher.py, models/manager/model.py
modals/session.js	schemas/session.schema.js, models/session.model.js
package.json	

## ■ File Descriptions

File Name	Description
config/db_config.js	Handles database configuration and environment variables.
connections/mongodb_connection.js	Establishes and manages connections to the MongoDB database.
services/db_services.js	Provides endpoints for database operations, such as connecting and disconnecting.
config/env.py	Handles environment variables and setup.
api/app.py	Sets up the Express.js app and defines API routes.
services/auth.py	Provides authentication and authorization services.
services/email.py	Handles email related operations, including sending and sending password reset emails.
services/auth/token_handler.py	Handles JWT token generation and validation.
services/auth/user_auth.py	Authenticates users and populates the req.user object.
services/db/count/schemas.py	Defines the schema for the 'Count' model

File Name	Description
services/db/count/management.py	Manages the 'Count' model, including CRUD operations
models/employee.schema.js	Defines the MongoDB schema for employee data storage.
services/staff.management.js	Handles employee management functionality, including user authentication, staff processing, and business logic.
models/manager/schema.py	Define the manager schema for the database.
services/auth/password_hasher.py	Handles password hashing and validation.
models/manager/model.py	Provides an interface for interacting with the manager data model.
schemas/session.schema.js	Defines the schema for session data using Mongoose
models/session.model.js	Provides a Mongoose model for interacting with session data
package.json	This file is the configuration file for the npm package and does not need to be split.

## ■■ New Microservice Structure

```

config
  -- db_config.js
  -- env.py

connections
  -- mongodb_connection.js

api
  -- app.py

auth
  -- token_handler.py
  -- user_auth.py
  -- password_hasher.py

db
  -- services.js
  -- count
  -- schemas.py
  -- management.py

models
  -- employee.schema.js
  -- manager
  -- schema.py
  -- model.py
  -- session
  -- schema.js
  -- model.js

services
  -- email.py
  -- staff
  -- management.js

package.json

```