

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
employee.js	1826	modals/employee.js
session.js	593	modals/session.js
count.js	364	modals/count.js
manager.js	1613	modals/manager.js
package.json	485	package.json

■ Microservice Splits

Original File	Split Files
index.js	server setup/server_init.py, models/db_operations.py, auth/auth_services.py, email/email_services.py
db.js	src/mongo/connector.js, src/mongo/constants.js
middleware/fetchuser.js	middleware/auth/token_verification.py, services/auth/user_auth_service.py
modals/manager.js	modals/manager/schema.js, auth/manager_password.js
modals/employee.js	services/models/employee_model.py, services/auth/user_credentials.py
modals/session.js	models/session_schema.py, models/session_mongoose.py
modals/count.js	infrastructure/data/models/count/model.py, infrastructure/data/schemas/count/schema.py
package.json	package.json

■ File Descriptions

File Name	Description
server setup/server_init.py	Sets up the Express.js server and configures necessary middleware.
models/db_operations.py	Handles data operations and interactions with the database.
auth/auth_services.py	Provides authentication and authorization services, including JWT token generation.
email/email_services.py	Sending and managing emails using the provided nodemailer functionality.
src/mongo/connector.js	Connects to the MongoDB database and establishes a connection.
src/mongo/constants.js	Contains constants and configurations for MongoDB connections.
middleware/auth/token_verification.py	Handles JWT token verification and validation.
services/auth/user_auth_service.py	Provides user authentication and session management services.
modals/manager/schema.js	Defines the schema for the Manager model.

File Name	Description
auth/manager_password.js	Handles password hashing logic for Managers.
services/models/employee_model.py	Defines the employee schema and database model.
services/auth/user_credentials.py	Handles user credentials and password hashing.
models/session_schema.py	Defines the session schema using Mongoose.
models/session_mongoose.py	Handles session creation, updating, and deletion using Mongoose.
infrastructure/data/models/count/model.py	Defines the database model for counts.
infrastructure/data/schemas/count/schema.py	Defines the schema for the count model using Mongoose.
package.json	This file does not need splitting and remains unchanged.

■■■ New Microservice Structure

```

/
server_setup
-- server_init.py

models
-- db_operations.py
-- session_schema.py
-- session_mongoose.py
-- employee_model.py
-- count
  ■■■ model.py
  -- schema.py

middleware
-- auth
-- token_verification.py

auth
  ■■■ auth_services.py
  -- manager_password.js

email
-- email_services.py

infrastructure
  ■■■ data
  -- models
  -- count
    -- model.py
  -- schemas
  -- count
    -- schema.py

src
-- mongo
  ■■■ connector.js
  -- constants.js

package.json

```