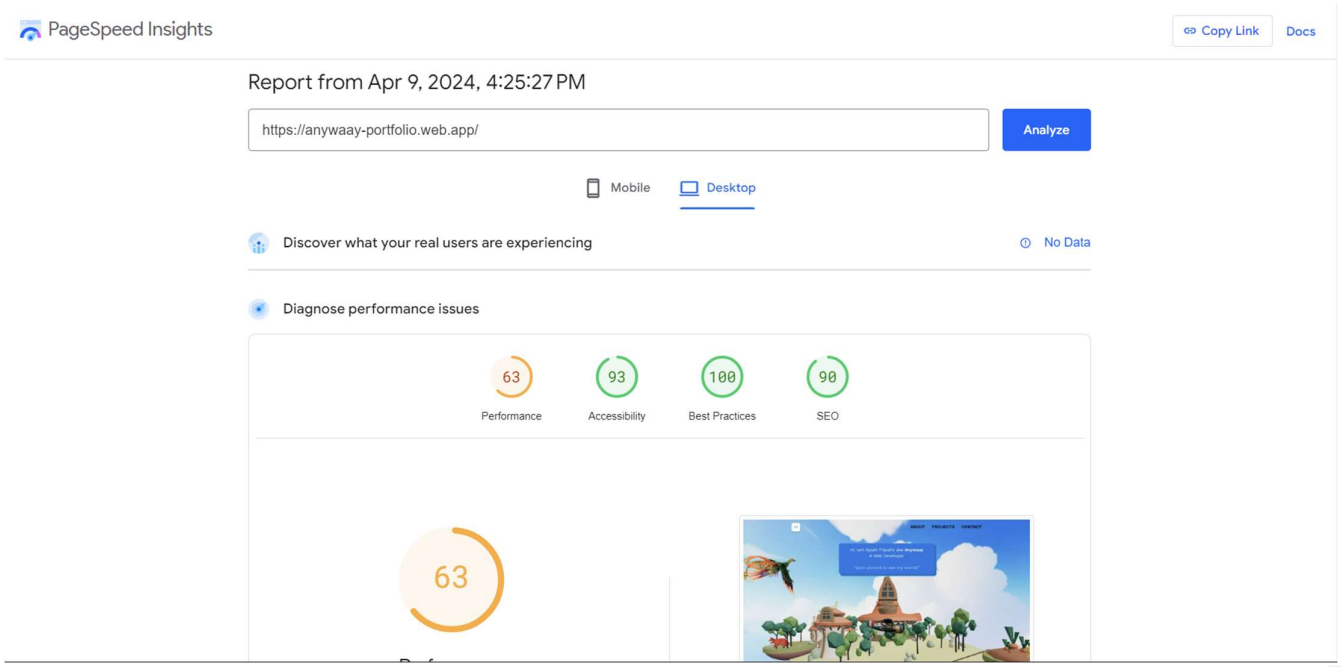
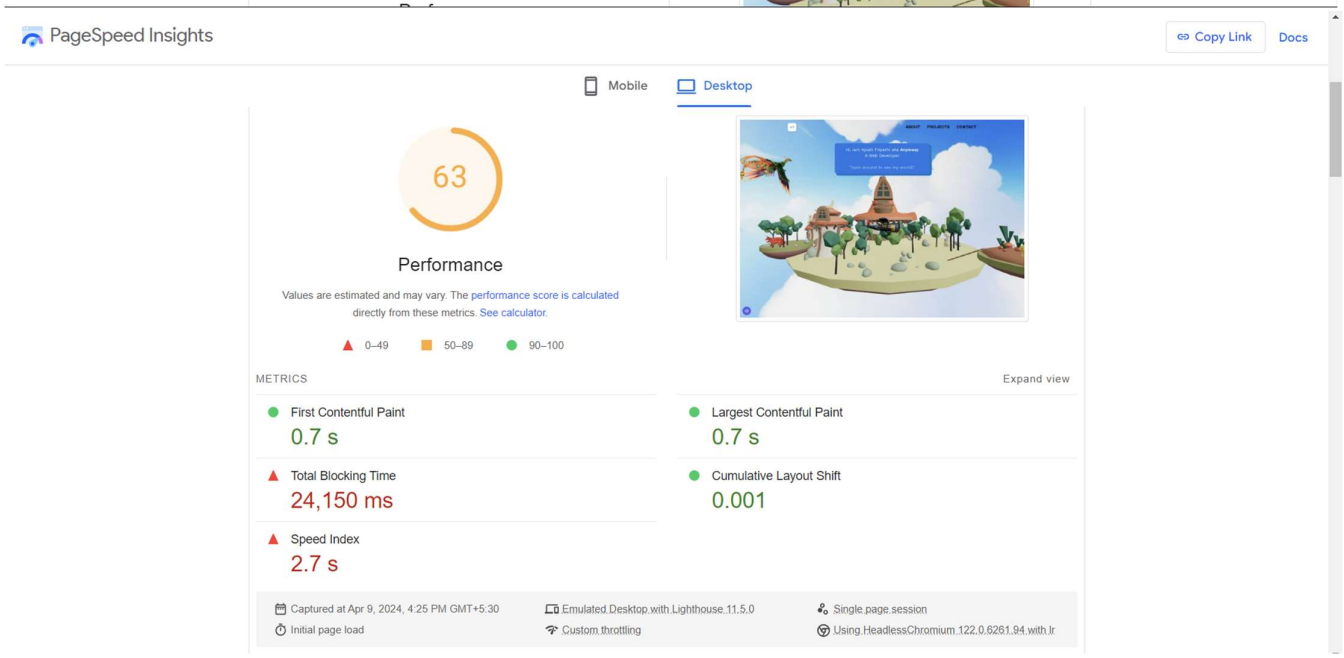
Ayush Tripathi
22CS3021

9th April, 2024

Web Technology Lab Assignment-11

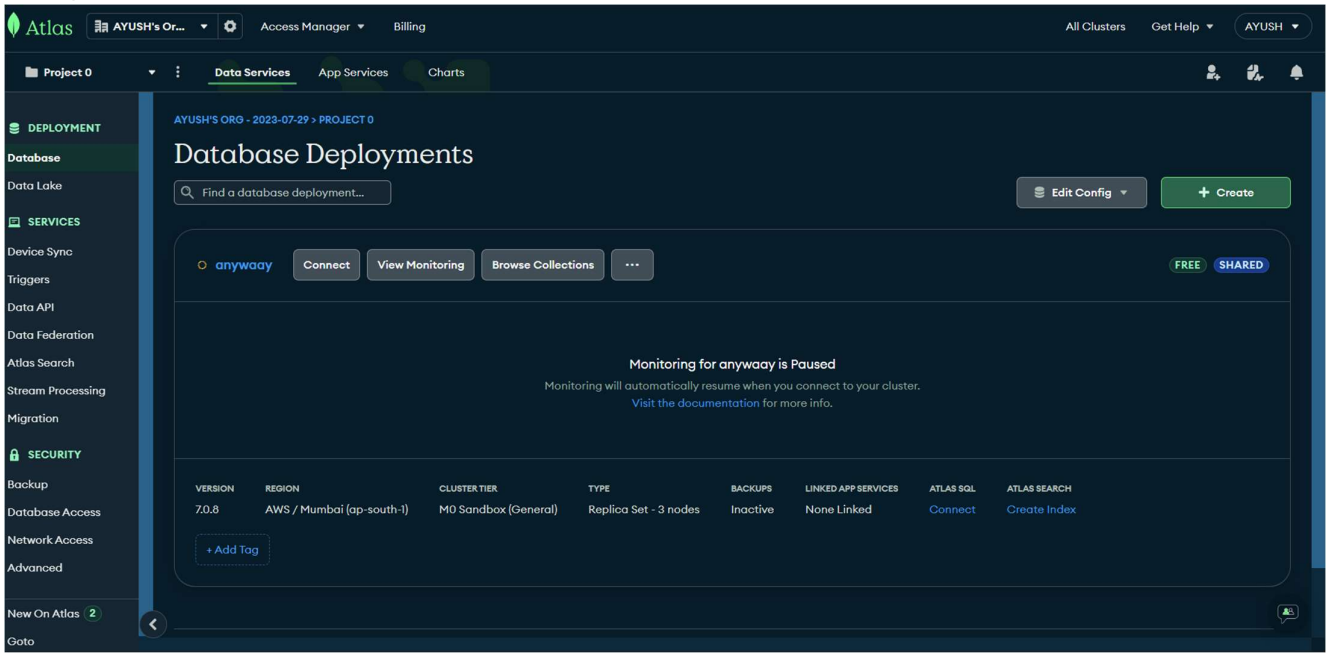
Task 1: Portfolio Performance Score on pageSpeed insights



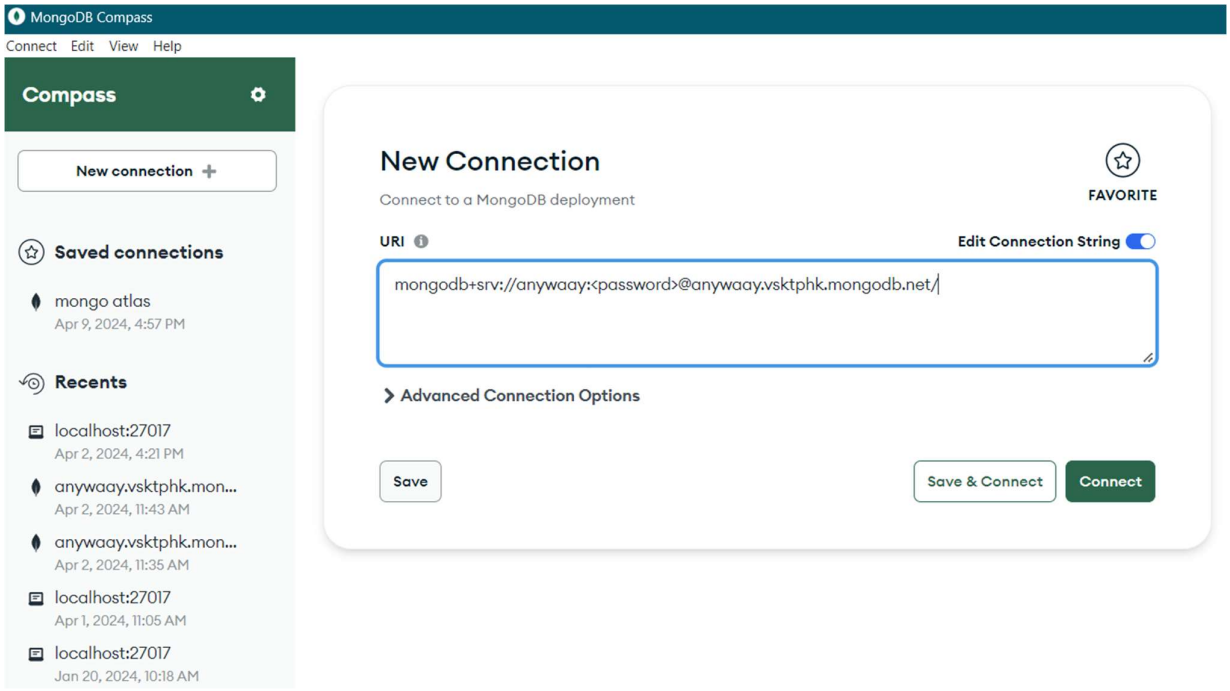


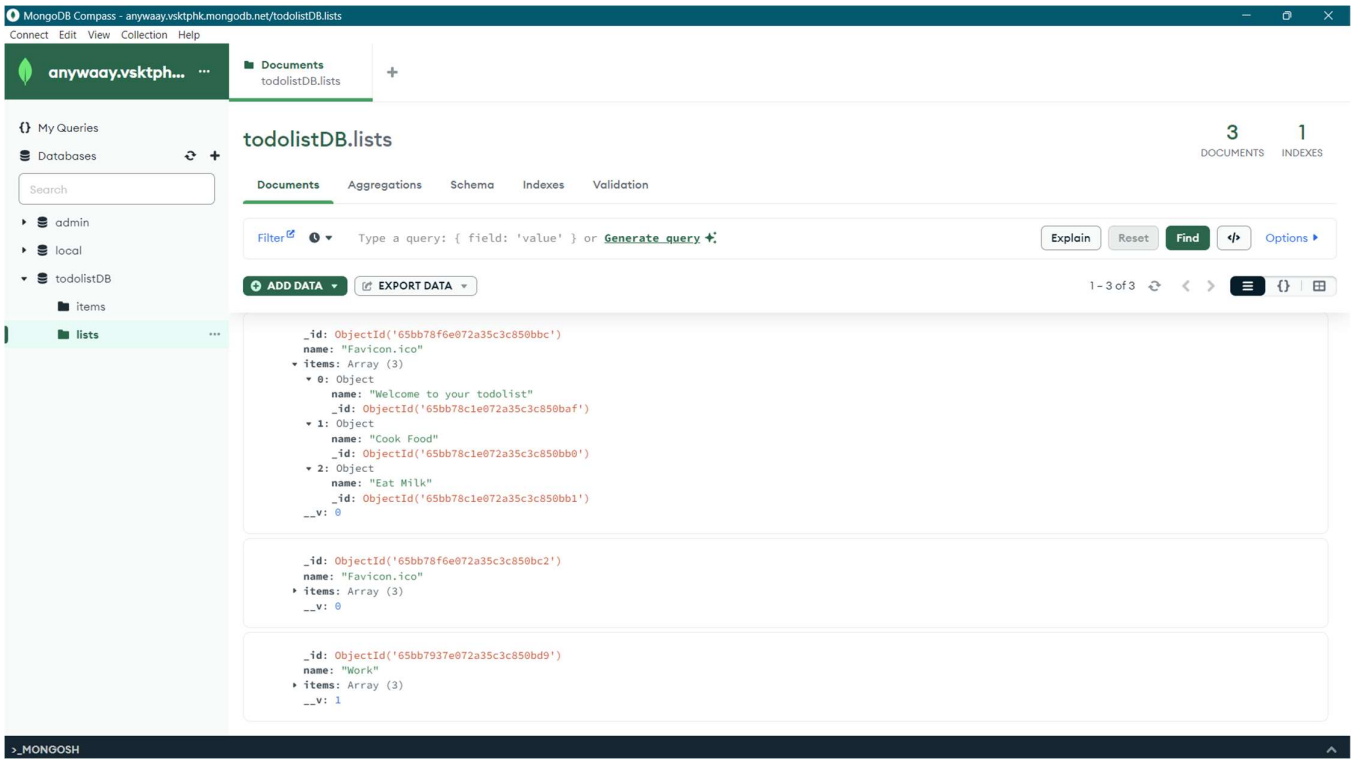
Task 2: MERN (MongoDB)

Mongodb atlas



Compass:





Connection.js:

```
import { MongoClient, ServerApiVersion } from "mongodb";

const url = process.env.ATLAS_URI;
const client = new MongoClient(url, {
   serverApi: {
      version: ServerApiVersion.v1,
      strict: true,
      deprecationErrors: true,
   },
})

try{
   await client.connect();
   await client.db("admin").command({ping: 1});
   console.log("Pinged your deployment. YOu succesfuly connected to MongoDB")
} catch(err){
   console.log(err)
}

let db = client.db("employees");

export default db;
```

record.js:

```
import express from "express";

// This will help us connect to the database
import db from "../db/connection.js";

// This help convert the id from string to ObjectId for the _id.
import { ObjectId } from "mongodb";
const router = express.Router();

// This section will help you get a list of all the records.
router.get("/", async (req, res) => {
  let collection = await db.collection("records");
  let results = await collection.find({}).toArray();
  res.send(results).status(200);
});

// This section will help you get a single record by id
router.get("/:id", async (req, res) => {
  let collection = await db.collection("records");
  let query = { _id: new ObjectId(req.params.id) };
  let result = await collection.findOne(query);

  if (!result) res.send("Not found").status(404);
  else res.send(result).status(200);
});
router.post("/", async (req, res) => {
  try {
    let newDocument = {
      name: req.body.name,
      position: req.body.position,
      level: req.body.level,
    };
    let collection = await db.collection("records");
    let result = await collection.insertOne(newDocument);
    res.send(result).status(204);
  } catch (err) {
    console.error(err);
    res.status(500).send("Error adding record");
  }
});
router.patch("/:id", async (req, res) => {
  try {
    const query = { _id: new ObjectId(req.params.id) };
    const updates = {
      $set: {
        name: req.body.name,
        position: req.body.position,
```

```
      level: req.body.level,
    },
  };
  let collection = await db.collection("records");
  let result = await collection.updateOne(query, updates);
  res.send(result).status(200);
 } catch (err) {
  console.error(err);
  res.status(500).send("Error updating record");
 }
});
router.delete("/:id", async (req, res) => {
 try {
  const query = { _id: new ObjectId(req.params.id) };

  const collection = db.collection("records");
  let result = await collection.deleteOne(query);

  res.send(result).status(200);
 } catch (err) {
  console.error(err);
  res.status(500).send("Error deleting record");
 }
});

export default router;
```

index.js:

```
import express from "express";
import cors from "cors";
import records from "./routes/record.js";

const PORT = process.env.PORT||5050;
const app = express();
app.use(cors());
app.use(express.json());
app.use("/record", records);

app.listen(PORT, ()=>{
  console.log(`app is listening at port  ${PORT}`);
})
```