1. To display Student information reports (All, parameterized).

```
<!DOCTYPE html>
<html ng-app="studentApp">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="StudentController">
  <h1>Student Report</h1>
  <select ng-model="filter" ng-change="loadReport()">
    <option value="all">All</option>
    <option value="Learning">Learning</option>
    <option value="Graduated">Graduated</option>
  </select>

  <table border="1">
    <tr><th>Name</th><th>Age</th><th>Status</th></tr>
    <tr ng-repeat="student in students">
      <td>{{ student.name }}</td><td>{{ student.age }}</td><td>{{ student.status }}</td>
    </tr>
  </table>

  <script src="app.js"></script>
</body>
</html>
```

```
app.js
var app=angular.module('studentApp', []);
  app.controller('StudentController', function($scope) {
    const students = [
      { name: "Raj", age: 22, status: "Learning" },
      { name: "Rohan", age: 25, status: "Graduated" },
      { name: "Rakesh", age: 21, status: "Learning" },
      { name: "Rahul", age: 23, status: "Graduated" },
      { name: "Ramao", age: 20, status: "Learning" }
    ];

    $scope.filter = 'all';
    $scope.loadReport = function() {
      $scope.students = $scope.filter === 'all'
        ? students
        : students.filter(s => s.status === $scope.filter);
    };

    $scope.loadReport();
  });
```

2. To create a user-friendly interface with a clean and proper design.

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script src="app.js"></script>
    <title>Login - AngularJS</title>
</head>
<body>
    <div ng-controller="LoginController">
        <h2>Login</h2>
        <form  ng-submit="login()">
                <input type="text"  ng-model="user.username" ng-required="true" placeholder="Username">
                <input type="password"  ng-model="user.password" ng-required="true" placeholder="Password">
            <button type="submit"  ng-disabled="loginForm.$invalid">Login</button>
        </form>
    </div>
</body>
</html>
```

```
var app = angular.module('myApp', []);

app.controller('LoginController', function($scope) {
    $scope.login = function() {
        if ($scope.user.username && $scope.user.password) {
            alert("Username: " + $scope.user.username + "\nPassword: " + $scope.user.password);
        }
    };
});
```

3. To demonstrate AngularJS services.

Data.txt

```
[
  {
    "Name" : "Mohan","RollNo" : 101
},
  {
    "Name" : "Sohan","RollNo" : 102
},
  {
    "Name" : "Rohan","RollNo" : 103
} ]
```

```
<html>
<head>
<title>AngularJS $http Service</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>
<script>
 var app = angular.module('myApp', []);
        app.controller('studentController', function ($scope, $http) { var url = "/DataService.txt";
            $http.get(url).then(function (response) {
                $scope.students = response.data;
            });
        });
</script>
</head>
<body>
<center>
<h1 >AngularJS $http Service</h1>
<div ng-app="myApp" ng-controller="studentController">
<table>
    <tr>
    <th>Name</th>
    <th>Roll No</th>
    </tr>
    <tr ng-repeat="student in students"> <td>{{ student.Name }}</td><td>{{ student.RollNo}}</td>
    </tr>
</table>
</div>
</center>
</body>
</html>
```

4. To demonstrate data bind.

```
<html ng-app="">
<head>
<title>AngularJS Data Binding</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>
</head>
<body style="background-color: {{bgcolor}};">
<h1 >AngularJS Data bind</h1>
<input type="text"  ng-model="bgcolor" placeholder="Type color Name for bgcolor">
</body>
</html>
```

5. To use ng-switch directive to display tasks in the UI.

```
<!DOCTYPE html>
<html lang="en" ng-app="taskApp">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script>
    var app = angular.module('taskApp', []);
  app.controller('TaskController', function($scope) {
    $scope.selectedTask = '';
  });
   </script>
</head>
<body>
  <div ng-controller="TaskController">
    <h2>Select a Task</h2>
    <select ng-model="selectedTask">
      <option value="task1">Task 1</option>
      <option value="task2">Task 2</option>
      <option value="task3">Task 3</option>
    </select>

    <h3>Task Description:</h3>
    <div ng-switch="selectedTask">
      <div ng-switch-when="task1">
        You have selected Task 1.
      </div>
      <div ng-switch-when="task2">
        You have selected Task 2.
      </div>
      <div ng-switch-when="task3">
        You have selected Task 3.
      </div>
    </div>
  </div>
</body>
</html>
```

6. To develop AngularJS program to create a login form, with validation for the username and password fields.

```
<html ng-app="loginApp">
<head>
      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
      <script>
         var app = angular.module('loginApp', []);
         app.controller('LoginController', function($scope) {
         $scope.submitForm = function() {
         if ($scope.loginForm.$valid) {
         alert("Username: " + $scope.user.username + "\nPassword: " + $scope.user.password);
         }
   };
});</script>
</head>
<body>
   <div ng-controller="LoginController" class="container">
      <h2>Login Form</h2>
      <form name="loginForm" ng-submit="submitForm()" novalidate>
         <div>
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" ng-model="user.username" required>
            <span ng-show="loginForm.username.$touched && loginForm.username.$invalid">Username is
required.</span>
         </div>
         <br>
         <div>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" ng-model="user.password" ng-minlength="6"
required>
            <span ng-show="loginForm.password.$touched && loginForm.password.$invalid">
               Password is required and must be at least 6 characters long.
            </span>
         </div>
         <br>
         <div>
            <button type="submit" ng-disabled="loginForm.$invalid">Login</button>
         </div>
      </form>
   </div>
</body>
</html>
```

7. To demonstrate insert, update, delete, select operations in MongoDB.

Create database

```
Use test
```

Create collection

```
db.createCollection(“users”)
```

Insert Operation

insert one row in collection

```
db.users.insertOne({  name: "Mohan", age: 25});
```

insert Many row in collection

```
db.users.insertMany([{ name: "rohan", age: 30 },{ name: "sohan", age: 35 }]);
```

Select (Read) Operation

Find all entries

```
db.users.find().pretty();
```

find one

```
db.users.findOne({ name: "rohan" });
```

Update Operation

Update one

```
db.users.updateOne( { name: "rohan" },    { $set: { age: 26 } }  );
```

update many

```
db.users.updateMany({ age: { $gt: 30 } }, { $set: { status: "Senior" } }  );
```

delete

```
db.users.deleteOne({ name: "rohan" });
```

8. To display tasks in a list with checkboxes for marking completion as per user choice.

```
<html ng-app="taskApp">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script >
    var app = angular.module('taskApp', []);
app.controller('TaskController', function($scope) {
  $scope.tasks = [
  ];
  $scope.addTask = function() {
    $scope.tasks.push({
      name: $scope.newTask,
      completed: false
    });
    $scope.newTask = '';
  };
  $scope.checkEnter = function(event) {
    if (event.keyCode === 13) {
      $scope.addTask();
    }
  };
});
  </script>
</head>
<body>
  <div ng-controller="TaskController">
    <h2>Task List</h2>
    <input type="text" ng-model="newTask" placeholder="Enter new task" ng-
keyup="checkEnter($event)">
    <button ng-click="addTask()">Add Task</button>
    <ul>
      <li ng-repeat="task in tasks">
        <label>
          <input type="checkbox" ng-model="task.completed">
          <span ng-class="{'completed': task.completed}">{{task.name}}</span>
        </label>
      </li>
    </ul>
  </div>
</body>
</html>
```

9. To demonstrate data bind and Services.

```
<!DOCTYPE html>
<html ng-app="app">
<head>
   <title>AngularJS Data Binding & Service Example</title>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
   <script>
      var app = angular.module('app', []);
 app.service('UserService', function() {
    this.getUser = function() {
       return { name: '' };
    };
 });
 app.controller('MainController', function($scope, UserService) {
    $scope.user = UserService.getUser();
 });
         </script>
</head>
<body>
   <div ng-controller="MainController">
      <h2>User Info</h2>
      <p>Name: {{ user.name }}</p>
      <input type="text" ng-model="user.name" placeholder="Enter your name">
   </div>
</body>
</html>
```

10. To use ng-if directive to display tasks in the UI.

```
<html ng-app="">
<head>
   <title>AngularJS ng-if Example</title>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

</head>
<body>
   <label>Click me: <input type="checkbox" ng-model="checked" ng-init="checked=true"
/></label><br/>
   Show when checked:
   <span ng-if="checked" >
     CheckBox is Checked.
   </span>
</body>
</html>
```

11. To develop AngularJS program to create a login form use multiple controls.

```
<html ng-app="loginApp">
<head>
  <title>AngularJS Login Form</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script>
    var app = angular.module('loginApp', []);
app.controller('LoginController', function($scope) {
  $scope.user = {
    username: '',
    password: '',
    rememberMe: false
  };
  $scope.submitForm = function() {
    if ($scope.loginForm.$valid) {
      var rememberMeStatus = $scope.user.rememberMe ? 'Yes' : 'No';
      alert('Login Successful!\nUsername: ' + $scope.user.username + '\nPassword: ' +
$scope.user.password + '\nRemember Me: ' + rememberMeStatus);
    }
  };
});
  </script>
</head>
<body>
  <div ng-controller="LoginController">
    <h2>Login Form</h2>
    <form ng-submit="submitForm()" name="loginForm" novalidate>
      <div>
        <label for="username">Username:</label>
        <input type="text" id="username" ng-model="user.username" required>
      </div>
      <div>
        <label for="password">Password:</label>
        <input type="password" id="password" ng-model="user.password" required>
      </div>
      <div>
        <label>
          <input type="checkbox" ng-model="user.rememberMe"> Remember Me
        </label>
      </div>
      <button type="submit" ng-disabled="loginForm.$invalid">Login</button>
    </form>
  </div>
</body>
</html>
```

12. To create database and structure in MongoDB.

Create database

```
Use test
```

Create collection

```
db.createCollection(“users”)
```

Insert Operation

insert one row in collection

```
db.users.insertOne({  name: "Mohan", age: 25});
```

insert Many row in collection

```
db.users.insertMany([{ name: "rohan", age: 30 },{ name: "sohan", age: 35 }]);
```

Select (Read) Operation

Find all entries

```
db.users.find().pretty();
```

find one

```
db.users.findOne({ name: "rohan" });
```

Update Operation

Update one

```
db.users.updateOne( { name: "rohan" },    { $set: { age: 26 } }  );
```

update many

```
db.users.updateMany({ age: { $gt: 30 } }, { $set: { status: "Senior" } }  );
```

delete

```
db.users.deleteOne({ name: "rohan" });
```

13. To display tasks in a list with checkboxes for marking completion as per user choice.

```
<html ng-app="taskApp">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script >
    var app = angular.module('taskApp', []);
app.controller('TaskController', function($scope) {
  $scope.tasks = [
  ];
  $scope.addTask = function() {
    $scope.tasks.push({
      name: $scope.newTask,
      completed: false
    });
    $scope.newTask = '';
  };
  $scope.checkEnter = function(event) {
    if (event.keyCode === 13) {
      $scope.addTask();
    }
  };
});
  </script>
</head>
<body>
  <div ng-controller="TaskController">
    <h2>Task List</h2>
    <input type="text" ng-model="newTask" placeholder="Enter new task" ng-keyup="checkEnter($event)">
    <button ng-click="addTask()">Add Task</button>
    <ul>
      <li ng-repeat="task in tasks">
        <label>
          <input type="checkbox" ng-model="task.completed">
          <span ng-class="{'completed': task.completed}">{{task.name}}</span>
        </label>
      </li>
    </ul>
  </div>
</body>
</html>
```

14. To create Student interface to stored and update the information.

```
Npm init -y
npm install express mongoose body-parser cors
```

SERVER.JS

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
const port = 3000;
app.use(bodyParser.json());
app.use(cors());

mongoose.connect('mongodb://localhost:27017/studentDB', { useNewUrlParser: true,
useUnifiedTopology: true })
  .then(() => console.log('MongoDB connected'))
  .catch((err) => console.log(err));

const studentSchema = new mongoose.Schema({
  name: String,
  age: Number,
  grade: String,
  email: String,
});

const Student = mongoose.model('Student', studentSchema);

app.get('/students', async (req, res) => {
  try {
    const students = await Student.find();
    res.json(students);
  } catch (err) {
    res.status(400).send(err);
  }
});

app.post('/students', async (req, res) => {
  const newStudent = new Student(req.body);
  try {
    await newStudent.save();
    res.json(newStudent);
  } catch (err) {
    res.status(400).send(err);
  }
});

app.put('/students/:id', async (req, res) => {
  try {
    const updatedStudent = await Student.findByIdAndUpdate(req.params.id, req.body, {
new: true });
    res.json(updatedStudent);
  } catch (err) {
    res.status(400).send(err);
  }
});

app.delete('/students/:id', async (req, res) => {
  try {
    await Student.findByIdAndDelete(req.params.id);
    res.json({ message: 'Student deleted' });
  } catch (err) {
    res.status(400).send(err);
  }
});

app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en" ng-app="studentApp">
<head>
<title>Student App</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="StudentController">

  <h2>Student Information</h2>

  <!-- Add Student Form -->
  <form ng-submit="addStudent()" novalidate>
    <label>Name:</label>
    <input type="text" ng-model="newStudent.name" required><br><br>

    <label>Age:</label>
    <input type="number" ng-model="newStudent.age" required><br><br>

    <label>Grade:</label>
    <input type="text" ng-model="newStudent.grade" required><br><br>

    <label>Email:</label>
    <input type="email" ng-model="newStudent.email" required><br><br>

    <button type="submit">Add Student</button>
  </form>

  <hr>

  <h3>Update Student Information</h3>
  <form ng-submit="updateStudent()" novalidate>
    <label>Select Student to Update:</label>
    <select ng-model="selectedStudent" ng-options="student.name for student in students"
ng-change="selectStudent(selectedStudent)"></select><br><br>

    <label>Name:</label>
    <input type="text" ng-model="selectedStudent.name" required><br><br>

    <label>Age:</label>
    <input type="number" ng-model="selectedStudent.age" required><br><br>

    <label>Grade:</label>
    <input type="text" ng-model="selectedStudent.grade" required><br><br>

    <label>Email:</label>
    <input type="email" ng-model="selectedStudent.email" required><br><br>

    <button type="submit">Update Student</button>
  </form>

  <hr>
  <h3>Students List</h3>
  <ul>
    <li ng-repeat="student in students">
      {{student.name}} - {{student.age}} - {{student.grade}} - {{student.email}}
      <button ng-click="deleteStudent(student._id)">Delete</button>
    </li>
  </ul>
</div>
<script src="app.js"></script>
</body>
</html>
```

APP.JS

```
var app = angular.module('studentApp', []);
app.controller('StudentController', function($scope, $http) {
   $scope.students = [];
   $scope.newStudent = {};
   $scope.selectedStudent = {};

   $http.get('http://localhost:3000/students')
      .then(function(response) {
         $scope.students = response.data;
      });
   $scope.addStudent = function() {
      $http.post('http://localhost:3000/students', $scope.newStudent)
         .then(function(response) {
            $scope.students.push(response.data);
            $scope.newStudent = {};
         });
   };
   $scope.selectStudent = function(student) {
      $scope.selectedStudent = angular.copy(student); // Copy selected student data for editing
   };
   $scope.updateStudent = function() {
      $http.put('http://localhost:3000/students/' + $scope.selectedStudent._id,
$scope.selectedStudent)
         .then(function(response) {
            var index = $scope.students.findIndex(student => student._id ===
response.data._id);
            if (index !== -1) {
               $scope.students[index] = response.data;
            }
         });
   };

   $scope.deleteStudent = function(id) {
      $http.delete('http://localhost:3000/students/' + id)
         .then(function(response) {
            $scope.students = $scope.students.filter(student => student._id !== id);
         });
   };
});
```

2. To develop a task manager application using AngularJS for the frontend.

```
<html ng-app="taskManagerApp">
<head><title>Task Manager</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script>
  var app = angular.module('taskManagerApp', []);
  app.controller('TaskController', function($scope) {
  $scope.tasks = [];
  $scope.newTask = {};
  $scope.addTask = function() {
    if ($scope.newTask.name && $scope.newTask.description) {
      $scope.tasks.push({
        name: $scope.newTask.name,
        description: $scope.newTask.description,
        completed: false});
      $scope.newTask = {};}};
  $scope.deleteTask = function(task) {
    var index = $scope.tasks.indexOf(task);
    if (index !== -1) {
      $scope.tasks.splice(index, 1);}};});
  </script></head>
<body>
  <div ng-controller="TaskController">
    <h2>Task Manager</h2>
    <form ng-submit="addTask()" novalidate>
      <label>Task Name:</label>
      <input type="text" ng-model="newTask.name" required placeholder="Enter task name"><br><br>
      <label>Task Description:</label>
      <input type="text" ng-model="newTask.description" required placeholder="Enter task description"><br><br>
      <button type="submit">Add Task</button>
    </form>
    <hr>
    <h3>Task List</h3>
    <ul>
      <li ng-repeat="task in tasks">
        <span ng-class="{'completed': task.completed}">
          {{ task.name }} - {{ task.description }}
        </span>
        <button ng-click="deleteTask(task)">Delete</button>
      </li>
    </ul>
  </div>
</body>
</html>
```

15. To use ng-Repeat directive to display tasks in the UI.

```
<html ng-app="taskApp">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script>
    var app=angular.module('taskApp', []);
  app.controller('TaskController', function($scope) {
    $scope.tasks = [
      { name: 'Learn AngularJS', completed: false },
      { name: 'Write Blog Post', completed: false },
      { name: 'Go to Home', completed: false }
    ];
    $scope.addTask = function() {
      if ($scope.newTask) {
        $scope.tasks.push({ name: $scope.newTask, completed: false });
        $scope.newTask = '';
    }
    };
$scope.toggleComplete = function(task) {
      task.completed = !task.completed;
    };
    $scope.removeTask = function(index) {
      $scope.tasks.splice(index, 1);
      };
  });

  </script>
</head>
<body ng-controller="TaskController">
  <h1>Task List</h1>
  <div>
    <input type="text" ng-model="newTask" placeholder="Add a new task">
    <button ng-click="addTask()">Add Task</button>
  </div>
  <ul>
    <li ng-repeat="task in tasks">
      {{ task.name }} <span ng-if="task.completed">[Completed]</span>
      <button ng-click="toggleComplete(task)">Toggle Complete</button>
      <button ng-click="removeTask($index)">Remove</button>
    </li>
  </ul>
</body>
</html>
```

16. To display tasks in a list with checkboxes for marking completion as per user choice.

```
<html ng-app="taskApp">
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script >
        var app = angular.module('taskApp', []);
app.controller('TaskController', function($scope) {
    $scope.tasks = [
    ];
    $scope.addTask = function() {
        $scope.tasks.push({
            name: $scope.newTask,
            completed: false
        });
        $scope.newTask = '';
    };
    $scope.checkEnter = function(event) {
        if (event.keyCode === 13) {
            $scope.addTask();
        }
    };
});
    </script>
</head>
<body>
    <div ng-controller="TaskController">
        <h2>Task List</h2>
        <input type="text" ng-model="newTask" placeholder="Enter new task" ng-
keyup="checkEnter($event)">
        <button ng-click="addTask()">Add Task</button>
        <ul>
            <li ng-repeat="task in tasks">
                <label>
                    <input type="checkbox" ng-model="task.completed">
                    <span ng-class="{'completed': task.completed}">{{task.name}}</span>
                </label>
            </li>
        </ul>
    </div>
</body>
</html>
```

17. To develop a Task Manager application using AngularJS for the frontend and MongoDB for the backend.

App.js

```
angular.module('taskManagerApp', [])
  .controller('TaskController', function($scope, $http) {
    const apiUrl = 'http://localhost:5000/tasks';

    function getTasks() {
      $http.get(apiUrl)
        .then(function(response) {
          $scope.tasks = response.data;
        })
        .catch(function(error) {
          console.error('Error fetching tasks:', error);
        });
    }
    $scope.addTask = function() {
      if ($scope.newTask.name) {
        $http.post(apiUrl, { name: $scope.newTask.name })
          .then(function(response) {
            $scope.tasks.push(response.data);
            $scope.newTask.name = '';
          })
          .catch(function(error) {
            console.error('Error adding task:', error);
          });
      }
    };

    $scope.toggleTaskCompletion = function(task) {
      $http.put(apiUrl + '/' + task._id, { completed: !task.completed })
        .then(function(response) {
          task.completed = response.data.completed;
        })
        .catch(function(error) {
          console.error('Error updating task:', error);
        });
    };

    $scope.deleteTask = function(taskId) {
      $http.delete(apiUrl + '/' + taskId)
        .then(function() {
          $scope.tasks = $scope.tasks.filter(task => task._id !== taskId);
        })
        .catch(function(error) {
```

```
        console.error('Error deleting task:', error);
      });
    };

    getTasks();
  });
```

INDEX.HTML

```
<html ng-app="taskManagerApp">
<head>
    <title>Task Manager</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="TaskController">

  <h1>Task Manager</h1>

  <div>
    <input type="text" ng-model="newTask.name" placeholder="Enter task name">
    <button ng-click="addTask()">Add Task</button>
  </div>

  <ul>
    <li ng-repeat="task in tasks">
      <span ng-class="{'completed': task.completed}">{{ task.name }}</span>
      <button ng-click="toggleTaskCompletion(task)">Toggle Complete</button>
      <button ng-click="deleteTask(task._id)">Delete</button>
    </li>
  </ul>

  <script src="app.js"></script>
</body>
</html>
```

SERVER.JS

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');
const app = express();
const port = 5000;
app.use(cors());
app.use(bodyParser.json());
```

```
mongoose.connect('mongodb://localhost:27017/taskManager', { useNewUrlParser: true,
useUnifiedTopology: true })
  .then(() => console.log('MongoDB Connected'))
  .catch(err => console.log('MongoDB connection error: ', err));

const taskSchema = new mongoose.Schema({
  name: String,
  completed: { type: Boolean, default: false }
});

const Task = mongoose.model('Task', taskSchema);

app.get('/tasks', (req, res) => {
  Task.find()
    .then(tasks => res.json(tasks))
    .catch(err => res.status(500).json({ message: 'Error fetching tasks', error: err }));
});

app.post('/tasks', (req, res) => {
  const newTask = new Task(req.body);
  newTask.save()
    .then(task => res.json(task))
    .catch(err => res.status(500).json({ message: 'Error adding task', error: err }));
});

app.put('/tasks/:id', (req, res) => {
  Task.findByIdAndUpdate(req.params.id, { completed: req.body.completed }, { new: true })
    .then(updatedTask => res.json(updatedTask))
    .catch(err => res.status(500).json({ message: 'Error updating task', error: err }));
});

app.delete('/tasks/:id', (req, res) => {
  Task.findByIdAndDelete(req.params.id)
    .then(() => res.json({ message: 'Task deleted' }))
    .catch(err => res.status(500).json({ message: 'Error deleting task', error: err }));
});

app.listen(port, () => {
  console.log(`Server running on port ${port}`);
});
```