API Documentation

Authentication Endpoints

1. Register a New User

Request:

```
POST {{baseUrl}}/auth/register
```

Headers:

```
Content-Type: application/json
```

Body:

```
{
  "name": "Test Captain",
  "email": "captain@example.com",
  "password": "password123",
  "role": "admin"
}
```

Expected Response (200 OK):

```
{
  "success": true,
  "message": "User registered successfully",
  "userId": "user123"
}
```

2. Login

Request:

```
POST {{baseUrl}}/auth/login
```

Headers:

```
Content-Type: application/json
```

Body:

```
{
  "email": "captain@example.com",
  "password": "password123"
}
```

Expected Response (200 OK):

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "_id": "user123",
    "name": "Test Captain",
    "email": "captain@example.com",
    "role": "admin"
  }
}
```

3. Get Current User

Request:

```
GET {{baseUrl}}/auth/me
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
{
  "user": {
    "_id": "user123",
    "name": "Test Captain",
    "email": "captain@example.com",
    "role": "captain"
  }
}
```

4. Refresh Token

Request:

```
POST {{baseUrl}}/auth/refresh-token
```

Headers:

```
Content-Type: application/json
```

Body:

```
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Expected Response (200 OK):

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

5. Logout

Request:

```
POST {{baseUrl}}/auth/logout
```

Headers:

```
 Authorization: Bearer {{token}}
Content-Type: application/json
```

Body:

```
 {
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Expected Response (200 OK):

```
 {
  "success": true,
  "message": "Logged out successfully"
}
```

6. Forgot Password

Request:

```
POST {{baseUrl}}/auth/forgot-password
```

Headers:

```
Content-Type: application/json
```

Body:

```
 {
  "email": "captain@example.com"
}
```

Expected Response (200 OK):

```
 {
  "success": true,
  "message": "Password reset email sent"
}
```

7. Reset Password

Request:

```
POST {{baseUrl}}/auth/reset-password
```

Headers:

```
Content-Type: application/json
```

Body:

```
 {
  "token": "reset-token-from-email",
  "password": "newpassword123"
}
```

Expected Response (200 OK):

```
{
  "success": true,
  "message": "Password reset successful"
}
```

Route Planning Endpoints

1. Create a New Route Plan

Request:

```
POST {{baseUrl}}/route-plan
```

Headers:

```
Authorization: Bearer {{token}}
Content-Type: application/json
```

Body:

```
{
  "shipId": "ship123",
  "startLocation": {
    "lat": 40.7128,
    "lon": -74.0060,
    "port": "New York"
  },
  "endLocation": {
    "lat": 48.8566,
    "lon": 2.3522,
    "port": "Paris"
  },
  "plannedStartDate": "2025-05-15T10:00:00Z",
  "cargoWeight": 25000
}
```

Expected Response (201 Created):

```
{
  "message": "Route plan created successfully",
  "routePlan": {
    "_id": "route123",
    "shipId": "ship123",
    "startLocation": {
      "lat": 40.7128,
      "lon": -74.0060,
      "port": "New York"
    },
    "endLocation": {
      "lat": 48.8566,
      "lon": 2.3522,
      "port": "Paris"
    },
    "plannedStartDate": "2025-05-15T10:00:00Z",
    "estimatedEndDate": "2025-05-29T06:12:00Z",
    "distance": 5842.3,
    "duration": 336.2,
    "estimatedFuelConsumption": 345.6,
    "waypoints": [
      [-74.0060, 40.7128],
      [-65.2356, 43.1234],
      [-55.7865, 46.4321],
      [2.3522, 48.8566]
    ],
    "averageSpeed": 17.5,
    "routeType": "hybrid",
    "weatherImpact": {
      "multiplier": 1.12,
      "details": [
        {
          "segment": 2,
          "factor": "wind",
          "value": 18.5,
          "impact": "3.4%"
        }
      ]
    },
    "confidenceLevel": "high",
    "aiFactors": {
      "weather": 0.6,
      "fuel": 0.3,
      "historical": 0.1
    }
  }
}
```

2. Get Routes for a Specific Ship

Request:

```
GET {{baseUrl}}/ships/ship123/routes
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
{
  "routes": [
    {
      "_id": "route123",
      "shipId": "ship123",
      "startLocation": {
        "lat": 40.7128,
        "lon": -74.0060,
        "port": "New York"
      },
      "endLocation": {
        "lat": 48.8566,
        "lon": 2.3522,
        "port": "Paris"
      },
      "plannedStartDate": "2025-05-15T10:00:00Z",
      "estimatedEndDate": "2025-05-29T06:12:00Z",
      "distance": 5842.3,
      "duration": 336.2,
      "status": "planned"
    },
    {
      "_id": "route124",
      "shipId": "ship123",
      "startLocation": {
        "lat": 48.8566,
        "lon": 2.3522,
        "port": "Paris"
      },
      "endLocation": {
        "lat": 40.7128,
        "lon": -74.0060,
        "port": "New York"
      },
      "plannedStartDate": "2025-06-15T10:00:00Z",
      "estimatedEndDate": "2025-06-29T14:45:00Z",
      "distance": 5823.7,
      "duration": 340.5,
      "status": "planned"
    }
  ]
}
```

3. Get a Specific Route

Request:

```
GET {{baseUrl}}/routes/route123
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
{
  "route": {
    "_id": "route123",
    "shipId": "ship123",
    "startLocation": {
      "lat": 40.7128,
      "lon": -74.0060,
      "port": "New York"
    },
    "endLocation": {
      "lat": 48.8566,
      "lon": 2.3522,
      "port": "Paris"
    },
    "plannedStartDate": "2025-05-15T10:00:00Z",
    "estimatedEndDate": "2025-05-29T06:12:00Z",
    "distance": 5842.3,
    "duration": 336.2,
    "estimatedFuelConsumption": 345.6,
    "waypoints": [
      [-74.0060, 40.7128],
      [-65.2356, 43.1234],
      [-55.7865, 46.4321],
      [2.3522, 48.8566]
    ],
    "averageSpeed": 17.5,
    "routeType": "hybrid",
    "status": "planned",
    "weatherImpact": {
      "multiplier": 1.12,
      "details": [
        {
          "segment": 2,
          "factor": "wind",
          "value": 18.5,
          "impact": "3.4%"
        }
      ]
    }
  }
}
```

4. Update Route Status

Request:

```
PATCH {{baseUrl}}/routes/route123/status
```

Headers:

```
Authorization: Bearer {{token}}
Content-Type: application/json
```

Body:

```
{
  "status": "in-progress"
}
```

Expected Response (200 OK):

```
{
  "message": "Route status updated successfully"
}
```

5. Generate Alternative Routes

Request:

```
POST {{baseUrl}}/route-alternatives
```

Headers:

```
Authorization: Bearer {{token}}
Content-Type: application/json
```

Body:

```
{
  "shipId": "ship123",
  "startLocation": {
    "lat": 40.7128,
    "lon": -74.0060,
    "port": "New York"
  },
  "endLocation": {
    "lat": 48.8566,
    "lon": 2.3522,
    "port": "Paris"
  },
  "plannedStartDate": "2025-05-15T10:00:00Z",
  "cargoWeight": 25000
}
```

Expected Response (200 OK):

```
{
  "alternatives": [
    {
      "strategy": "direct",
      "distance": 5823.7,
      "duration": 291.2,
      "averageSpeed": 20.0,
      "waypoints": [[-74.0060, 40.7128], [2.3522, 48.8566]],
      "predictedFuelConsumption": 356.2,
      "startLocation": {
        "lat": 40.7128,
        "lon": -74.0060,
        "port": "New York"
      },
      "endLocation": {
        "lat": 48.8566,
        "lon": 2.3522,
        "port": "Paris"
      },
      "estimatedStartDate": "2025-05-15T10:00:00Z",
      "estimatedEndDate": "2025-05-27T13:12:00Z"
    },
    {
      "strategy": "weather-optimized",
      "distance": 6104.5,
      "duration": 326.8,
      "averageSpeed": 18.7,
      "waypoints": [[-74.0060, 40.7128], [-65.2356, 43.1234], [2.3522, 48.8566]],
      "predictedFuelConsumption": 347.8,
      "startLocation": {
        "lat": 40.7128,
        "lon": -74.0060,
        "port": "New York"
      },
      "endLocation": {
        "lat": 48.8566,
        "lon": 2.3522,
        "port": "Paris"
      },
      "estimatedStartDate": "2025-05-15T10:00:00Z",
      "estimatedEndDate": "2025-05-28T17:48:00Z"
    },
    {
      "strategy": "fuel-optimized",
      "distance": 5842.3,
      "duration": 389.5,
      "averageSpeed": 15.0,
      "waypoints": [[-74.0060, 40.7128], [-55.7865, 46.4321], [2.3522, 48.8566]],
      "predictedFuelConsumption": 290.4,
      "startLocation": {
        "lat": 40.7128,
        "lon": -74.0060,
        "port": "New York"
      },
      "endLocation": {
        "lat": 48.8566,
        "lon": 2.3522,
        "port": "Paris"
      },
      "estimatedStartDate": "2025-05-15T10:00:00Z",
      "estimatedEndDate": "2025-05-31T01:30:00Z"
    }
  ]
}
```

6. Update Route with Latest Weather

Request:

```
GET {{baseUrl}}/routes/route123/update-weather
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
{
  "message": "Route updated with latest weather and optimization",
  "updatedRoute": {
    "_id": "route123",
    "shipId": "ship123",
    "startLocation": {
      "lat": 40.7128,
      "lon": -74.0060,
      "port": "New York"
    },
    "endLocation": {
      "lat": 48.8566,
      "lon": 2.3522,
      "port": "Paris"
    },
    "plannedStartDate": "2025-05-15T10:00:00Z",
    "estimatedEndDate": "2025-05-29T10:15:00Z",
    "distance": 5863.1,
    "duration": 338.7,
    "waypoints": [
      [-74.0060, 40.7128],
      [-65.2356, 43.1234],
      [-53.1234, 45.6789],
      [2.3522, 48.8566]
    ],
    "averageSpeed": 17.3,
    "routeType": "weather-optimized",
    "weatherImpact": {
      "multiplier": 1.15,
      "details": [
        {
          "segment": 1,
          "factor": "wind",
          "value": 20.2,
          "impact": "4.1%"
        }
      ]
    },
    "fuelConsumption": 352.4
  }
}
```

7. Get Optimal Speed Recommendation

Request:

```
GET {{baseUrl}}/routes/route123/optimal-speed?currentPosition=
{"lat":42.3601,"lon":-71.0589}&remainingFuel=150
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
{
  "routeId": "route123",
  "currentPosition": {
    "lat": 42.3601,
    "lon": -71.0589
  },
  "remainingDistance": 5234.8,
  "optimalSpeed": 16.5,
  "adjustmentReason": "Speed decreased for fuel efficiency while maintaining schedule",
  "estimatedArrival": "2025-05-29T04:13:00Z"
}
```

8. Complete a Route

Request:

```
POST {{baseUrl}}/routes/route123/complete
```

Headers:

```
 Authorization: Bearer {{token}}
Content-Type: application/json
```

Body:

```
 {
  "actualEndDate": "2025-05-29T15:32:00Z",
  "actualDistance": 5856.2,
  "actualDuration": 342.5,
  "fuelConsumed": 352.8
}
```

Expected Response (200 OK):

```
 {
  "message": "Route marked as completed successfully"
}
```

Fuel Estimation Endpoints

1. Get Fuel Estimate for a Specific Route

Request:

```
GET {{baseUrl}}/fuel-estimate?routeId=route123
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
 {
  "routeId": "route123",
  "shipId": "ship123",
  "fuelEstimate": {
    "totalFuelConsumption": 352.8,
    "fuelConsumptionBySegment": [
      {
        "from": 0,
        "to": 1,
        "distance": 1240.5,
        "consumption": 84.2
      },
      {
        "from": 1,
        "to": 2,
        "distance": 1358.7,
        "consumption": 97.3
      },
      {
        "from": 2,
        "to": 3,
        "distance": 1825.3,
        "consumption": 120.1
      },
      {
        "from": 3,
        "to": 4,
        "distance": 1431.7,
        "consumption": 51.2
      }
    ],
    "averageFuelConsumptionPerHour": 1.05,
    "averageFuelConsumptionPerMile": 0.06,
    "weatherImpact": {
      "multiplier": 1.12,
      "details": [
        {
          "segment": 2,
          "factor": "wind",
          "value": 18.5,
          "impact": "3.4%"
        },
        {
          "segment": 3,
          "factor": "precipitation",
          "value": 2.7,
          "impact": "2.7%"
        }
      ]
    },
    "confidenceLevel": "high"
  },
  "fuelEfficiencyTips": [
    {
      "category": "speed",
      "tip": "Reducing speed by 10% could reduce fuel consumption by approximately 20%",
      "potentialSavings": "15-25%",
      "priority": "high"
    },
    {
      "category": "weather",
      "tip": "Consider alternative routes to avoid adverse weather conditions",
      "potentialSavings": "12.0%",
      "priority": "medium"
    },
    {
      "category": "cargo",
      "tip": "Ensure optimal cargo distribution for stability and fuel efficiency",
      "potentialSavings": "2-5%",
      "priority": "low"
    }
  ]
}
```

2. Get Fuel Estimate for a Hypothetical Journey

Request:

```
GET {{baseUrl}}/fuel-estimate?
shipId=ship123&startLocation[lat]=40.7128&startLocation[lon]=-74.0060&endLocation[lat]=48.856
05-15T10:00:00Z&cargoWeight=25000
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK): Similar to the response for a specific route, but calculated for the hypothetical journey parameters.

Maintenance Scheduling Endpoints

1. Get Maintenance Schedule for a Ship

Request:

```
GET {{baseUrl}}/maintenance-schedule?shipId=ship123&months=6
```

Headers:

```
Authorization: Bearer {{token}}
```

Expected Response (200 OK):

```
{
  "shipId": "ship123",
  "shipDetails": {
    "name": "Pacific Voyager",
    "type": "container",
    "age": 8.3,
    "totalMileage": 356240,
    "totalEngineHours": 24850
  },
  "maintenanceHistory": [
    {
      "maintenanceType": "engine",
      "description": "Full engine service",
      "date": "2025-01-15T08:30:00Z",
      "duration": 3,
      "tasks": ["Oil change", "Filter replacement", "Belt inspection", "Cooling system check"],
      "notes": "Replaced main cooling pump"
    },
    {
      "maintenanceType": "hull",
      "description": "Hull inspection and cleaning",
      "date": "2024-11-08T10:15:00Z",
      "duration": 2,
      "tasks": ["Hull cleaning", "Anti-fouling treatment", "Structural inspection"],
      "notes": "Minor corrosion found on port side"
    }
  ],
  "recommendedSchedule": [
    {
      "maintenanceType": "engine",
      "description": "Engine Maintenance",
      "lastPerformed": "2025-01-15T08:30:00Z",
      "dueDate": "2025-04-15T00:00:00Z",
      "estimatedDuration": 2,
      "priority": "medium",
      "tasks": [
        "Oil change",
        "Filter replacement",
        "Belt inspection",
        "Cooling system check"
      ]
    },
    {
      "maintenanceType": "electronics",
      "description": "Navigation and Communication Systems",
      "lastPerformed": "2024-10-22T14:00:00Z",
      "dueDate": "2025-04-22T00:00:00Z",
      "estimatedDuration": 1,
      "priority": "high",
      "adjustmentReason": "Scheduled before planned route",
      "tasks": [
        "Navigation equipment calibration",
        "Communication systems check",
        "Radar inspection",
        "Software updates"
      ]
    },
    {
      "maintenanceType": "safety",
      "description": "Safety Equipment Inspection",
      "lastPerformed": "2024-12-05T09:45:00Z",
      "dueDate": "2025-05-05T00:00:00Z",
      "estimatedDuration": 1,
      "priority": "urgent",
      "tasks": [
        "Life raft inspection",
        "Fire suppression systems check",
        "Emergency equipment test",
        "Safety drill"
      ]
    }
  ],
  "criticalMaintenanceItems": [
    {
      "maintenanceType": "safety",
      "description": "Safety Equipment Inspection",
      "lastPerformed": "2024-12-05T09:45:00Z",
      "dueDate": "2025-05-05T00:00:00Z",
      "estimatedDuration": 1,
      "priority": "urgent",
      "tasks": [
        "Life raft inspection",
        "Fire suppression systems check",
        "Emergency equipment test",
        "Safety drill"
      ]
    }
  ],
  "nextMaintenanceWindow": {
    "start": "2025-04-02T00:00:00Z",
    "end": "2025-04-09T00:00:00Z",
    "duration": 7,
    "conflictingRoutes": []
  }
}
```

Analytics Endpoints

Get Analytics Data

Request:

```
GET {{baseUrl}}/analytics?shipId=ship123&startDate=2024-01-01T00:00:00Z&endDate=2025-01-01T00:00:00Z&type=all&timeframe=monthly
```

Headers:

```
Authorization: Bearer {{token}}
```

Query Parameters: - `shipId` (optional) - Specific ship or all ships if omitted - `startDate` (optional) - Start of analysis period (defaults to 1 year ago) - `endDate` (optional) - End of analysis period (defaults to now) - `type` (optional) - Analysis type: ‘fuel’, ‘routes’, ‘efficiency’, or ‘all’ (default) - `timeframe` (optional) - Period grouping: ‘daily’, ‘weekly’, ‘monthly’ (default), ‘quarterly’, ‘yearly’

Expected Response (200 OK):

```
{
  "timeframe": "monthly",
  "period": {
    "start": "2024-01-01T00:00:00Z",
    "end": "2025-01-01T00:00:00Z"
  },
  "summary": {
    "routesCompleted": 24,
    "totalDistance": 135240,
    "totalFuelUsed": 8652.3,
    "averageSpeed": 17.8,
    "averageFuelEfficiency": 15.6,
    "totalShips": 1,
    "topShip": {
      "shipId": "ship123",
      "efficiency": 15.6,
      "totalDistance": 135240,
      "routesCompleted": 24
    },
    "mostEfficientRoute": {
      "routeId": "route135",
      "shipId": "ship123",
      "startLocation": "New York (40.7128, -74.0060)",
      "endLocation": "London (51.5074, -0.1278)",
      "distance": 5575.2,
      "time": 312.8,
      "date": "2024-06-12T09:45:00Z"
    }
  },
  "trends": {
    "fuelEfficiency": {
      "periods": [
        {
          "period": "Jan 2024",
          "value": 14.2
        },
        {
          "period": "Feb 2024",
          "value": 14.8
        }
      ],
      "overallTrend": "improving",
      "changePercent": 12.5
    },
    "routeEfficiency": {
      "periods": [
        {
          "period": "Jan 2024",
          "value": 0.82
        },
        {
          "period": "Feb 2024",
          "value": 0.85
        }
      ],
      "overallTrend": "stable",
      "changePercent": 3.1
    },
    "speed": {
      "periods": [
        {
          "period": "Jan 2024",
          "value": 18.2
        },
        {
          "period": "Feb 2024",
          "value": 17.8
        }
      ],
      "overallTrend": "declining",
      "changePercent": -2.2
    }
  },
  "fuelConsumption": {
    "total": 8652.3,
    "average": {
      "perRoute": 360.5,
      "perMile": 0.064,
      "perDay": 23.7
    },
    "byShip": [
      {
        "shipId": "ship123",
        "totalFuel": 8652.3,
        "totalDistance": 135240,
        "averagePer1000Miles": 64.0,
        "routeCount": 24
      }
    ],
    "factors": {
      "weather": {
        "Clear": {
          "count": 14,
          "totalDistance": 82456.3,
          "totalFuel": 4982.1,
          "efficiency": 16.5
        },
        "Rain": {
          "count": 8,
          "totalDistance": 45234.1,
          "totalFuel": 3120.2,
          "efficiency": 14.5
        },
        "Storm": {
          "count": 2,
          "totalDistance": 7549.6,
          "totalFuel": 550.0,
          "efficiency": 13.7
        }
      },
      "cargoWeight": {
        "0-10000": {
          "count": 5,
          "totalDistance": 28750.2,
          "totalFuel": 1602.3,
          "efficiency": 17.9
        },
        "10001-20000": {
          "count": 12,
          "totalDistance": 68432.1,
          "totalFuel": 4205.8,
          "efficiency": 16.3
        },
        "20001-30000": {
          "count": 7,
          "totalDistance": 38057.7,
          "totalFuel": 2844.2,
          "efficiency": 13.4
        }
      }
    }
  },
  "routeAnalysis": {
    "total": {
      "routes": 24,
      "distance": 135240,
      "time": 7625.4
    },
    "averages": {
      "speed": 17.8,
      "routeLength": 5635.0,
      "timePerRoute": 317.7,
      "deviation": 8.2
    },
    "mostEfficient": [
      {
        "routeId": "route135",
        "shipId": "ship123",
        "startLocation": "New York (40.7128, -74.0060)",
        "endLocation": "London (51.5074, -0.1278)",
        "distance": 5575.2,
        "timeTaken": 312.8,
        "averageSpeed": 17.8,
        "deviationPercent": 2.8,
        "efficiencyScore": 6.35,
        "date": "2024-06-12T09:45:00Z"
      }
    ],
    "byWeather": {
      "Clear": {
        "count": 14,
        "totalDistance": 82456.3,
        "totalTime": 4582.0,
        "averageSpeed": 18.0,
        "deviation": 5.2
      },
      "Rain": {
        "count": 8,
        "totalDistance": 45234.1,
        "totalTime": 2542.4,
        "averageSpeed": 17.8,
        "deviation": 9.7
      }
    }
  },
  "efficiency": {
    "overall": {
      "score": 0.82,
      "trend": "improving",
      "comparisonToIndustry": "above-average"
    },
    "byShip": [
      {
        "shipId": "ship123",
        "overallScore": 0.82,
        "routeEfficiency": 0.85,
        "fuelEfficiency": 0.78,
        "maintenanceEfficiency": 0.88,
        "weatherEfficiency": 0.76,
        "routeCount": 24
      }
    ],
    "byFactor": {
      "route": 0.85,
      "fuel": 0.78,
      "maintenance": 0.88,
      "weather": 0.76,
      "speed": 0.80
    },
    "topFactors": [
      {
        "factor": "maintenance",
        "score": 0.88,
        "contribution": 0.21
      },
      {
        "factor": "route",
        "score": 0.85,
        "contribution": 0.21
      },
      {
        "factor": "speed",
        "score": 0.80,
        "contribution": 0.20
      }
    ],
    "improvementAreas": [
      {
        "factor": "weather",
        "score": 0.76,
        "potentialImprovement": 14.0
      },
      {
        "factor": "fuel",
        "score": 0.78,
        "potentialImprovement": 12.0
      }
    ]
  },
  "recommendations": [
    {
      "category": "weather",
      "title": "Improve Weather Routing",
      "description": "Weather conditions are significantly impacting fuel efficiency. Enhance weather routing to avoid adverse conditions.",
      "potentialImpact": "high",
      "difficultyToImplement": "medium"
    },
    {
      "category": "speed",
      "title": "Optimize Speed Management",
      "description": "Implement dynamic speed optimization based on weather, cargo, and schedule requirements to balance fuel efficiency and timely delivery.",
      "potentialImpact": "high",
      "difficultyToImplement": "low"
    },
    {
      "category": "fuel",
      "title": "Reduce Fuel Consumption",
      "description": "Fuel consumption per mile is higher than optimal. Consider engine maintenance and optimizing cruising speed.",
      "potentialImpact": "medium",
      "difficultyToImplement": "medium"
    }
  ]
}
```

WebSocket Testing

```
URL: `ws://localhost:3000`
```

1. WebSocket Events to Test

Join a Route Planning Session

Send:

```
{
  "event": "join-route",
  "data": {
    "routeId": "route123"
  }
}
```

Expected Response:

```
{
  "event": "joined-route",
  "data": {
    "routeId": "route123",
    "message": "Successfully joined route planning session"
  }
}
```

Request a Route Plan

Send:

```
{
  "event": "request-route-plan",
  "data": {
    "shipId": "ship123",
    "startLocation": {
      "lat": 40.7128,
      "lon": -74.0060,
      "port": "New York"
    },
    "endLocation": {
      "lat": 48.8566,
      "lon": 2.3522,
      "port": "Paris"
    },
    "plannedStartDate": "2025-05-15T10:00:00Z",
    "cargoWeight": 25000,
    "estimatedDays": 14
  }
}
```

Expected Response:

```
{
  "event": "route-plan",
  "data": {
    "routeId": "route123",
    "startLocation": {
      "lat": 40.7128,
      "lon": -74.0060,
      "port": "New York"
    },
    "endLocation": {
      "lat": 48.8566,
      "lon": 2.3522,
      "port": "Paris"
    },
    "plannedStartDate": "2025-05-15T10:00:00Z",
    "estimatedEndDate": "2025-05-29T10:00:00Z",
    "distance": 5842.3,
    "duration": 336.2,
    "averageSpeed": 17.5,
    "waypoints": [
      [-74.0060, 40.7128],
      [-65.2356, 43.1234],
      [-55.7865, 46.4321],
      [2.3522, 48.8566]
    ],
    "fuelConsumption": 345.6,
    "weatherConditions": [...],
    "criticalConditions": [...],
    "routeType": "hybrid"
  }
}
```

Update Ship Position in Real-time

Send:

```
{
  "event": "update-position",
  "data": {
    "routeId": "route123",
    "currentPosition": {
      "lat": 42.3601,
      "lon": -71.0589
    },
    "currentTime": "2025-05-18T14:30:00Z"
  }
}
```

Expected Response:

```
{
  "event": "route-update",
  "data": {
    "routeId": "route123",
    "waypoints": [...],
    "estimatedEndDate": "2025-05-29T12:30:00Z",
    "duration": 338.5,
    "distance": 5863.1,
    "updatedAt": "2025-05-18T14:30:00Z",
    "reason": "Real-time weather update"
  }
}
```

Note: You should also receive a `position-updated` event:

```
{
  "event": "position-updated",
  "data": {
    "success": true,
    "routeId": "route123",
    "currentPosition": {
      "lat": 42.3601,
      "lon": -71.0589
    },
    "timestamp": "2025-05-18T14:30:00Z"
  }
}
```

Request Alternative Routes

Send:

```
{
  "event": "request-alternatives",
  "data": {
    "routeId": "route123"
  }
}
```

Expected Response:

```
{
  "event": "route-alternatives",
  "data": {
    "routeId": "route123",
    "alternatives": [
      {
        "routeType": "fastest",
        "waypoints": [...],
        "distance": 5823.7,
        "duration": 291.2,
        "averageSpeed": 20.0,
        "fuelConsumption": 356.2
      },
      {
        "routeType": "fuel-efficient",
        "waypoints": [...],
        "distance": 5842.3,
        "duration": 389.5,
        "averageSpeed": 15.0,
        "fuelConsumption": 290.4
      },
      {
        "routeType": "weather-optimized",
        "waypoints": [...],
        "distance": 6104.5,
        "duration": 326.8,
        "averageSpeed": 18.7,
        "fuelConsumption": 347.8
      }
    ]
  }
}
```

Select a Route

Send:

```
{
  "event": "select-route",
  "data": {
    "routeId": "route123",
    "selectedRoute": {
      "waypoints": [[-74.0060, 40.7128], [-55.7865, 46.4321], [2.3522, 48.8566]],
      "distance": 5842.3,
      "duration": 389.5,
      "fuelConsumption": 290.4,
      "routeType": "fuel-optimized"
    }
  }
}
```

Expected Response:

```
{
  "event": "route-selected",
  "data": {
    "routeId": "route123",
    "waypoints": [[-74.0060, 40.7128], [-55.7865, 46.4321], [2.3522, 48.8566]],
    "distance": 5842.3,
    "duration": 389.5,
    "fuelConsumption": 290.4,
    "routeType": "fuel-optimized",
    "selectedBy": "user123",
    "selectedAt": "2025-05-18T15:45:00Z"
  }
}
```

Testing Workflows

1. Authentication Flow

1. **Register** a new user
2. **Login** with the new user credentials
3. **Get current user** info
4. **Refresh token** for extended session
5. **Logout** to end session

2. Route Planning Flow

1. **Create a new route plan**
2. **Get route details** to verify creation
3. **Generate alternative routes** for the same journey
4. **Update route with latest weather**
5. **Update route status** to “in-progress”
6. **Get optimal speed** recommendation at different points
7. **Complete the route** with actual values

3. Fuel Estimation Flow

1. **Create a new route plan**
2. **Get fuel estimate** for the created route
3. **Get fuel estimate** for a hypothetical journey
4. **Update route parameters** and check fuel estimate changes
5. **Complete the route** and compare actual vs. estimated fuel

4. Maintenance Schedule Flow

1. **Get maintenance schedule** for a ship with no completed routes
2. **Create and complete** several routes for the ship
3. **Get maintenance schedule** again to see how usage affects schedule
4. **Check critical maintenance items** and next available window

5. Analytics Flow

1. **Create and complete** multiple routes with various parameters
2. **Get analytics** with different timeframes (monthly, quarterly)
3. **Get analytics** filtered by ship
4. **Check for recommendations** and efficiency scores
5. **Implement suggested changes** and verify improvements

6. WebSocket Flow

1. **Establish WebSocket connection** with authentication
2. **Join a route** planning session
3. **Request a route plan** with specific parameters
4. **Send position updates** at regular intervals
5. **Request alternative routes** based on current position
6. **Select a different route** and verify the change
7. **Test reconnection** and session persistence