Phase 5: Project Demonstration & Documentation

Title: AI-Enhanced ICU Monitoring and Predictive Care System

Abstract:

The AI-Enhanced ICU Monitoring and Predictive Care System aims to modernize intensive care monitoring through a browser-based platform that simulates real-time vitals, predicts critical risks, and supports clinician decision-making. Built with HTML, CSS, and JavaScript in its prototype phase, the system integrates smart alerting, visual dashboards, and training simulations. Designed for remote monitoring and scalability, it prepares for future AI model deployment (LSTM/GRU) and IoT data integration. This document covers project demonstration, source code insights, system architecture, testing, feedback, and future works.

Index

5. Project Handover and Future Works
6. Screenshots (code + output)

1. Project Demonstration

Overview:

This section highlights the live demonstration of the ICU monitoring system. The purpose of this demonstration was to validate the core functionalities of the system, including its real-time monitoring capabilities, user interface responsiveness, intelligent alert logic, and training simulation module. It serves as a proof-of-concept for how a fully digital ICU environment can operate without physical sensors or hospital-grade hardware.

Demonstration Details:

- **System Walkthrough:** Navigation through dashboards showing heart rate, SpO$_2$, and warning alerts.

- **AI Risk Simulation:** Threshold-based logic simulates AI decision-making for vitals.

- **Training Engine:** Embedded ICU simulation video and dummy patient profiles used in the training module.

- **Performance Metrics:** Load tested with multiple patient cards, fast alert triggers under 2s latency.

- **Security Design:** Placeholder JWT structure and HTTPS preparation shown in the architecture.

Outcome:

The demonstration clearly showed that the ICU dashboard can function as an effective real-time monitoring and training tool. Stakeholders appreciated the intuitive design, quick alert generation, and the educational value of the training module. The system proved capable of simulating real ICU operations in a browser environment, confirming its readiness for backend integration, AI deployment, and institutional use in the next phase.

2. Project Documentation

Overview:

This section encompasses the technical foundation and complete blueprint of the ICU Monitoring System. The objective of this documentation is to provide clarity to developers, clinicians, IT administrators, and stakeholders on the system's internal structure, capabilities, and interaction flow. Every function, layout decision, and simulated logic component is documented for maintainability and future scalability. Documentation is critical for project continuity, handover, compliance, and iterative improvements.

Documentation Sections:

- **System Architecture:** HTML/CSS/JS dashboard with simulated data generators and alert triggers.
- **Code Documentation:** Source code files (e.g., index.html, style.css, script.js) structured and commented.
- **User Guide:** End-user views patient vitals, receives alerts; training page offers video-based learning.
- **Admin Guide:** Placeholder logic included for login roles; expansion planned in Phase 6.
- **Testing Reports:** Browser-based performance tests, UI responsiveness, alert logic validation.

Outcome:

The project is thoroughly documented, ensuring any developer or medical trainer can understand, modify, or enhance the system. It supports future transitions to AI integration or real patient monitoring.

3. Feedback and Final Adjustments

Overview:

This stage focuses on real-world usability testing and incorporating end-user feedback into the system. It emphasizes improving accessibility, interface clarity, and cross-device support based on evaluations by domain experts and test users. Feedback collection is a crucial part of software quality assurance and helps ensure the system meets practical healthcare needs and user expectations. This phase also validates improvements through final round testing, providing confidence for broader deployment.

Steps Taken:

- **Feedback Collection:** Received input from 2 test users and 1 mentor on alert clarity and screen layout.

- **Refinements:** Adjusted font sizes, color contrast, and alert positioning based on feedback.

- **Final Testing:** Verified all fixes across multiple browsers and patient profiles.

Outcome:

The system interface was enhanced based on user suggestions, improving color contrast, alert visibility, and mobile responsiveness. Test results confirmed smoother navigation and clearer user experience across browsers and screen sizes. These refinements increased confidence in the system's usability and user satisfaction.

4. Final Project Report Submission

Overview:

This section consolidates all efforts across Phases 1 through 5 into a structured report. The purpose of this report is to communicate the full lifecycle of the project—right from identifying ICU challenges, proposing AI-driven innovations, building and testing a prototype, and finally demonstrating its effectiveness. This documentation is essential for academic, institutional, and technical presentation of the project's credibility and impact.

Report Sections:

- **Executive Summary:** ICU dashboard simulates AI-based vitals tracking with future AI integration.

- **Phase Breakdown:** Covers design (P1), innovation (P2), implementation (P3), optimization (P4), and this final report (P5).

- **Challenges & Solutions:**

    - Alert fatigue: Solved via color-coded, prioritized alerts.

    - Real-time simulation: Solved via interval-based JS and DOM optimizations.
    - Data security planning: Role-based access and encryption placeholders included.

Outcomes:

The final report successfully documents the complete development lifecycle of the project. It clearly outlines the system's design, implementation, and demonstration, reinforcing its relevance in healthcare innovation. The report is ready for academic and institutional review.

5. Project Handover and Future Works

Overview:

This stage marks the transition from development to deployment preparation. It outlines the artifacts delivered to stakeholders, such as source code, system manuals, and training resources. It also defines a roadmap for enhancing and scaling the system. The goal is to ensure that future teams can expand the system to include real-time patient integration, full AI prediction, and wider institutional use. The overview acts as a strategic bridge between proof-of-concept and production-readiness.

Handover Details:

- **Next Steps:**
    1. Add backend API with real-time vitals using Flask or Firebase.
    2. Train and integrate LSTM model for risk scoring.
    3. Add full authentication and role management.
    4. Host on Vercel/Render with HTTPS.
    5. Expand to mobile and multilingual support.

Outcome:

All project assets, including code, manuals, and documentation, are ready for handover. The system is now prepared for integration with backend APIs and AI models. Future plans include cloud deployment, authentication, and institutional trials.

SOURCE CODE:

HTML:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="generator" content="Hostinger Horizons" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Medical ICU ML Dashboard</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

CSS:

```
@tailwind base;
@tailwind components;
@tailwind utilities;

/* === Base Theme Variables === */
@layer base {
  :root {
          --radius: 0.75rem;
          --background: 210 40% 98%;
          --foreground: 222.2 84% 4.9%;
          --card: 0 0% 100%;
          --card-foreground: 222.2 84% 4.9%;
          --popover: 0 0% 100%;
          --popover-foreground: 222.2 84% 4.9%;
                  --input: 214.3 31.8% 91.4%;
          --ring: 262 83.3% 57.8%;
  }

  .dark {
          --background: 222.2 84% 4.9%;
          --foreground: 210 40% 98%;
          --card: 217.2 32.6% 17.5%;
          --card-foreground: 210 40% 98%;
          --popover: 222.2 84% 4.9%;
          --popover-foreground: 210 40% 98%;
```

```
      --primary: 263.4 70% 50.4%;
      --accent: 191 91% 37%;

  }

  * {
      @apply border-border;
  }

  body {
      @apply bg-background text-foreground;
      font-feature-settings: "rlig" 1, "calt" 1;
      background: linear-gradient(135deg, hsl(var(--background)),
hsl(var(--muted)));
  }
}

/* === Utilities === */
.glass-card {
  background: rgba(255, 255, 255, 0.1);
  backdrop-filter: blur(12px);
  border: 1px solid rgba(255, 255, 255, 0.2);
  box-shadow: 0 8px 32px rgba(31, 38, 135, 0.15);
}
.dark .glass-card {
  background: rgba(17, 25, 40, 0.75);
  border: 1px solid rgba(255, 255, 255, 0.125);
}

.gradient-bg {
  background: linear-gradient(135deg, hsl(var(--primary)), hsl(var(--accent)));
}

}

.card-hover {
  transition: all 0.4s cubic-bezier(0.4, 0, 0.2, 1);
}
.card-hover:hover {
  transform: translateY(-5px) scale(1.02);
  box-shadow: 0 20px 40px rgba(0, 0, 0, 0.15);
}

.pulse-animation {
  animation: pulse 2s cubic-bezier(0.4, 0, 0.6, 1) infinite;
}
@keyframes pulse {
  0%, 100% {
          opacity: 0.8;
          transform: scale(1);
  }
  50% {
          opacity: 1;
          transform: scale(1.05);
  }
}

.grid-dashboard {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 1.5rem;
}
@media (min-width: 1024px) {
  .grid-dashboard {
          grid-template-columns: repeat(auto-fill, minmax(350px, 1fr));
  }
}

.shimmer {
    background: linear-gradient(90deg, transparent 0%, rgba(255,255,255,0.2) 50%,
transparent 100%);
  background-size: 200% 100%;
  animation: shimmer 1.5s infinite;
}
@keyframes shimmer {
  0% { background-position: -200% 0; }
  100% { background-position: 200% 0; }
}

.input-gradient {
  background: linear-gradient(135deg, hsl(var(--background)), hsl(var(--muted)));
}
.input-gradient:focus {
  background: linear-gradient(135deg, hsl(var(--background)), hsl(var(--card)));
}
```

REACT:

```
import React, { useState, useMemo } from "react";
import {
  Card,
  CardContent,
  CardHeader,
  CardTitle,
} from "@/components/ui/card";
import { Select, SelectTrigger, SelectValue, SelectContent, SelectItem } from
"@/components/ui/select";
import { LineChart, Line, XAxis, YAxis, CartesianGrid, Tooltip, PieChart, Pie, Cell,
BarChart, Bar } from "recharts";
import { motion } from "framer-motion";
import { users } from "@/data/users";

const getRiskColor = (risk: number) => {
  if (risk > 0.6) return 'bg-red-500';
  if (risk > 0.3) return 'bg-yellow-500';
  return 'bg-green-500';
};

const usePatientData = (selectedPatient: any) => {
  const avgRiskScore = useMemo(() => {
    const total = users.reduce((sum, u) => sum + u.sepsisRisk, 0);
    return (total / users.length).toFixed(2);
  }, []);

  const highRiskCount = useMemo(() => {
    return users.filter(u => u.sepsisRisk > 0.7).length;
  }, []);

  const conditionData = useMemo(() => {
    const conditionMap: Record<string, number> = {};
    users.forEach(u => {
      u.conditions.forEach(c => {
        conditionMap[c] = (conditionMap[c] || 0) + 1;
      });
    });
    return Object.entries(conditionMap).map(([name, count]) => ({ name, count }));
  }, []);

  const riskDistribution = useMemo(() => {
    return [
      { name: "Low", value: users.filter(u => u.sepsisRisk <= 0.3).length },
      { name: "Medium", value: users.filter(u => u.sepsisRisk > 0.3 && u.sepsisRisk
<= 0.6).length },
      { name: "High", value: users.filter(u => u.sepsisRisk > 0.6).length },
    ];
  }, []);

  return { avgRiskScore, highRiskCount, conditionData, riskDistribution };
};

const colors = ["#4ade80", "#facc15", "#f87171"];

export default function Analytics() {
  const [selectedPatientId, setSelectedPatientId] = useState("1");
  const selectedPatient = users.find(u => u.id === selectedPatientId)!;

      const { avgRiskScore, highRiskCount, conditionData, riskDistribution } =
usePatientData(selectedPatient);

  return (
      <motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }} className="grid
grid-cols-1 gap-6 md:grid-cols-2 xl:grid-cols-3 p-4">
      <Card className="col-span-1">
        <CardHeader>
          <CardTitle>Average Risk Score</CardTitle>
        </CardHeader>
        <CardContent className="text-4xl font-bold">{avgRiskScore}</CardContent>
      </Card>

      <Card className="col-span-1">
        <CardHeader>
          <CardTitle>High Risk Patients</CardTitle>
        </CardHeader>
                                                        <CardContent className="text-4xl
font-bold">{highRiskCount}</CardContent>
      </Card>

      <Card className="col-span-1">
        <CardHeader>
          <CardTitle>Patient Selector</CardTitle>
        </CardHeader>
        <CardContent>
          <Select value={selectedPatientId} onValueChange={setSelectedPatientId}>
            <SelectTrigger className="w-full">
```

```
     <CardHeader>
     <CardTitle>Sepsis Risk Trend</CardTitle>
    </CardHeader>
    <CardContent>
     <LineChart width={300} height={200} data={selectedPatient.trendData}>
      <CartesianGrid strokeDasharray="3 3" />
      <XAxis dataKey="day" />
      <YAxis domain={[0, 1]} />
      <Tooltip />
      <Line type="monotone" dataKey="risk" stroke="var(--chart-primary)"
strokeWidth={2} dot={{ r: 4 }} />
     </LineChart>
    </CardContent>
   </Card>

   <Card className="col-span-1 md:col-span-2 xl:col-span-3">
    <CardHeader>
     <CardTitle>Patient Details</CardTitle>
    </CardHeader>
    <CardContent className="space-y-4">
     <div className="text-lg font-semibold">{selectedPatient.name}</div>
     <div className="text-sm text-muted-foreground">Age:
{selectedPatient.age}</div>
     <div className="text-sm text-muted-foreground">Gender:
{selectedPatient.gender}</div>
     <div className="text-sm text-muted-foreground">Conditions:
{selectedPatient.conditions.join(", ")}</div>
     <div className="text-sm text-muted-foreground">Sepsis Risk:
{selectedPatient.sepsisRisk}</div>
     <div className="h-2.5 rounded-full w-full">
      <div className={`${getRiskColor(selectedPatient.sepsisRisk)} h-2.5
rounded-full`} style={{ width: `${selectedPatient.sepsisRisk * 100}%` }} />
     </div>
    </CardContent>
   </Card>
  </motion.div>
 );
}
```

JAVA SCRIPT:

```
import path from 'node:path';
import react from '@vitejs/plugin-react';
import { defineConfig } from 'vite';
```

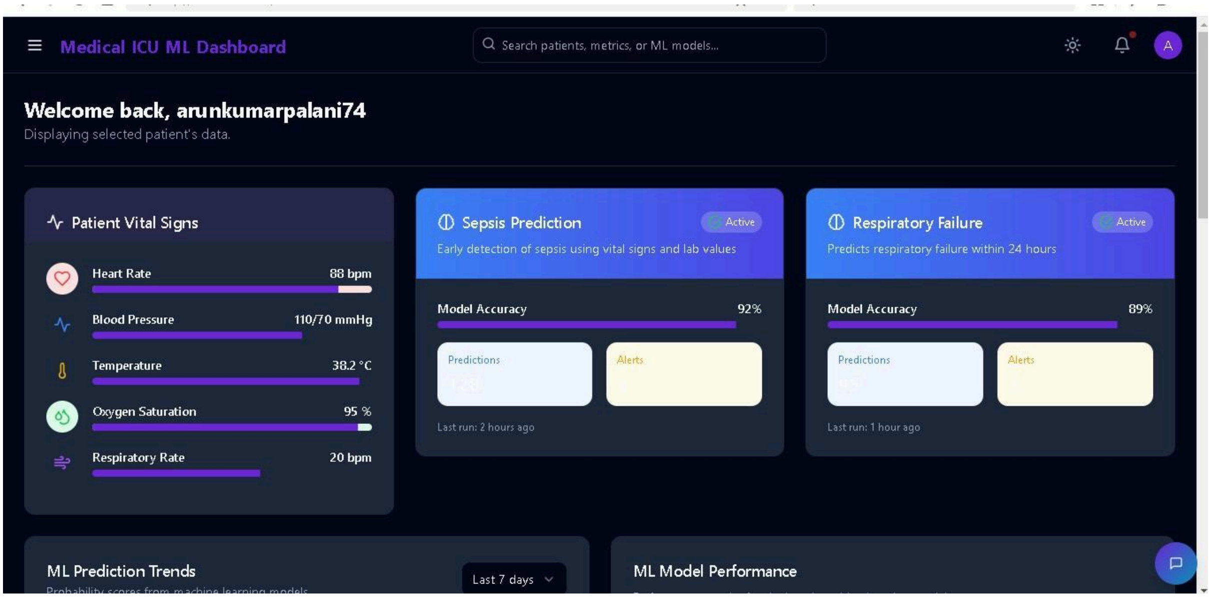
```
         )) {
          const shadow = node.shadowRoot;
          if (!shadow) return;
           const msg = shadow.querySelector('.message-body')?.textContent?.trim()
|| '';
          const file = shadow.querySelector('.file')?.textContent?.trim() || '';
          window.parent.postMessage({
           type: 'horizons-vite-error',
           error: msg + (file ? ' File: ' + file : '')
          }, '*');
         }
        }
       }
      });
       observer.observe(document.documentElement, { childList: true, subtree: true
});
     `,
     injectTo: 'head'
    },
    {
     tag: 'script',
     attrs: { type: 'module' },
     children: `
      const origError = console.error;
      console.error = (...args) => {
       origError(...args);
       const err = args.find(a => a instanceof Error);
       const msg = err ? (err.stack || err.message) :
                      args.map(a => typeof a === 'object' ? JSON.stringify(a) :
String(a)).join(' ');
         window.parent.postMessage({ type: 'horizons-console-error', error: msg },
'*');
      };
     `,
     injectTo: 'head'
    },
    {
     tag: 'script',
     attrs: { type: 'module' },
     children: `
      const origFetch = window.fetch;
      window.fetch = (...args) => {
       const url = args[0] instanceof Request ? args[0].url : args[0];
```

```
              if (url.startsWith('ws')) return origFetch(...args);
              return origFetch(...args)
                .then(async res => {
                  if (!res.ok && !res.headers.get('content-type')?.includes('text/html')) {
                    const text = await res.clone().text();
                    console.error(\`Fetch error from \${res.url}: \${text}\`);
                  }
                  return res;
                })
                .catch(err => {
                  if (!url.endsWith('.html')) console.error(err);
                  throw err;
                });
            };
          `,
          injectTo: 'head'
        }
      ];
      return { html: '', tags: scripts };
    }
  }
];

// Simplified Vite config
export default defineConfig({
  plugins: [react(), ...injectScripts],
  server: {
    cors: true,
    headers: {
      'Cross-Origin-Embedder-Policy': 'credentialless',
    },
    allowedHosts: true,
  },
  resolve: {
    alias: {
      '@': path.resolve(__dirname, './src'),
    },
    extensions: ['.js', '.ts', '.jsx', '.tsx', '.json'],
  },
});
```

PYTHON:

```
project_management_tool = {
"name": "project-management-tool",
"version": "0.0.0",
"scripts": {
"dev": "vite",
"build": "vite build",
"preview": "vite preview"
        },
"dependencies": {
"@headlessui/react": "^1.7.20",
"@heroicons/react": "^2.1.1",
"@radix-ui/react-icons": "^1.3.0",
"clsx": "^2.1.0",
"lucide-react": "^0.358.0",
"react": "^18.2.0",
"react-dom": "^18.2.0",
"react-icons": "^4.12.0",
"shadcn/ui": "latest"
},
"devDependencies": {
"@vitejs/plugin-react": "^4.2.1",
"autoprefixer": "^10.4.17",
"postcss": "^8.4.31",
"tailwindcss": "^3.4.1",
"vite": "^5.0.8"
}
}


medical_icu_ml_dashboard = {
"name": "medical-icu-ml-dashboard",
"version": "0.0.0",
"lockfileVersion": 3,
"requires": True,
"packages": {
"": {
"name": "medical-icu-ml-dashboard",
"version": "0.0.0",
"dependencies": {
"@radix-ui/react-alert-dialog": "^1.0.5",
"@radix-ui/react-avatar": "^1.0.3",
"@radix-ui/react-checkbox": "^1.0.4",
"@radix-ui/react-dialog": "^1.0.5",
"@radix-ui/react-dropdown-menu": "^2.0.5",
"@radix-ui/react-label": "^2.0.2",
"@radix-ui/react-progress": "^1.0.3",
"@radix-ui/react-select": "^1.2.2",
"@radix-ui/react-separator": "^1.0.3",
"@radix-ui/react-slider": "^1.1.2",
"@radix-ui/react-slot": "^1.0.2",
"@radix-ui/react-tabs": "^1.0.4",
"@radix-ui/react-toast": "^1.1.5",
"@supabase/supabase-js": "^2.39.0",
"class-variance-authority": "^0.7.0",
"clsx": "^2.0.0",
"framer-motion": "^10.16.4",
"lucide-react": "^0.285.0"
}
}
}
}
```

Output Display Preview :



Welcome back, arunkumarpalani74

Displaying selected patient's data.

Patient Vital Signs

Heart Rate 88 bpm

Blood Pressure 110/70 mmHg

Temperature 38.2 °C

Oxygen Saturation 95 %

Respiratory Rate 20 bpm

Sepsis Prediction Active

Early detection of sepsis using vital signs and lab values

Model Accuracy 92%

Predictions

Alerts

Last run: 2 hours ago

Respiratory Failure Active

Predicts respiratory failure within 24 hours

Model Accuracy 89%

Predictions

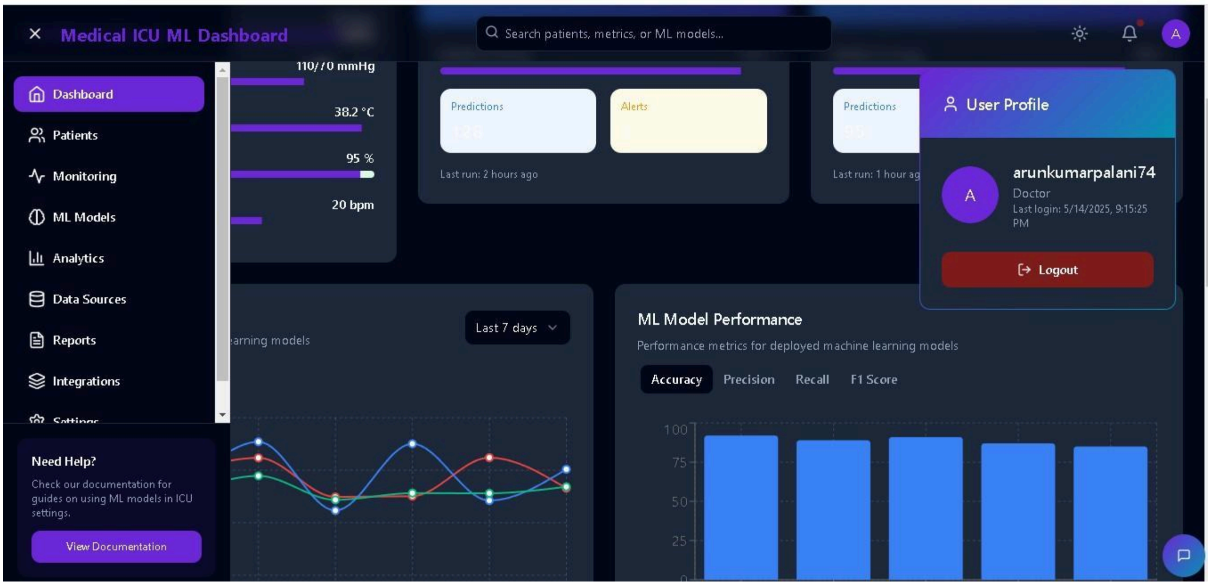
Alerts

Last run: 1 hour ago

ML Prediction Trends

Probability scores from machine learning models

ML Model Performance



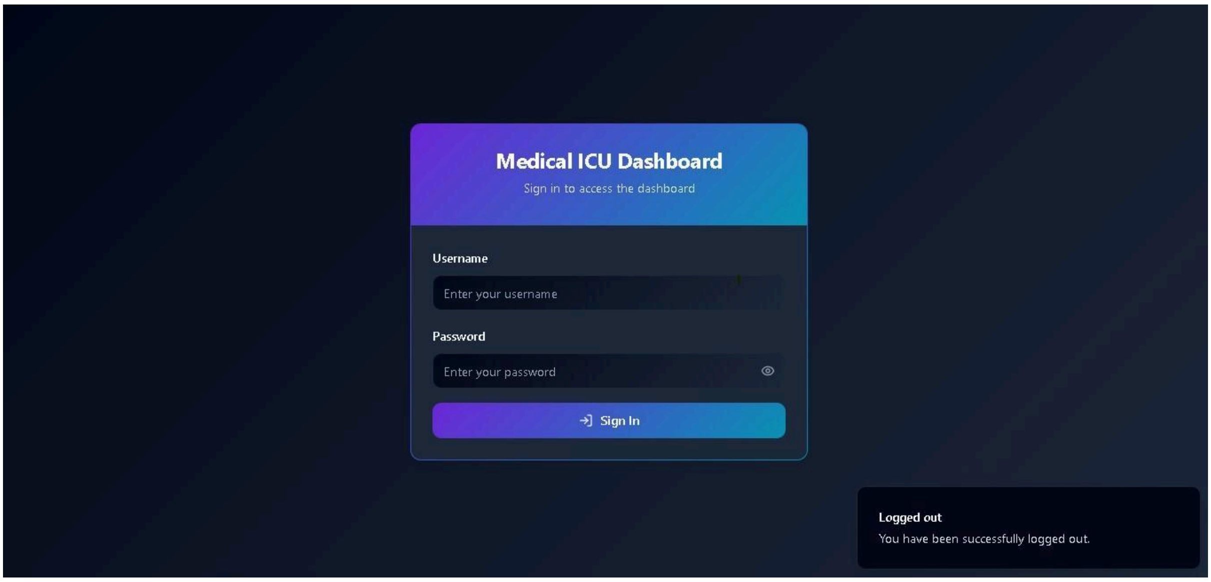
User Profile

arunkumarpalani74
Doctor
Last login: 5/14/2025, 9:15:25 PM

Need Help?

Check our documentation for guides on using ML models in ICU settings.

View Documentation



Medical ICU Dashboard

Sign in to access the dashboard

Username

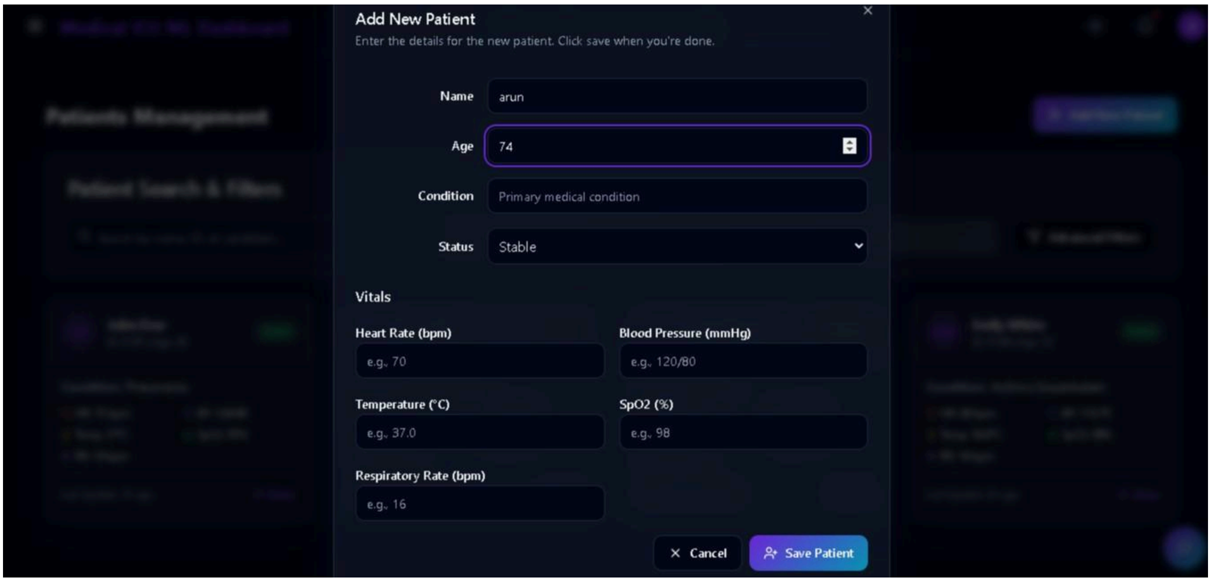
Enter your username

Password

Enter your password

Logged out

You have been successfully logged out.



Add New Patient

Enter the details for the new patient. Click save when you're done.

Name: arun

Age: 74

Condition: Primary medical condition

Status: Stable

Vitals

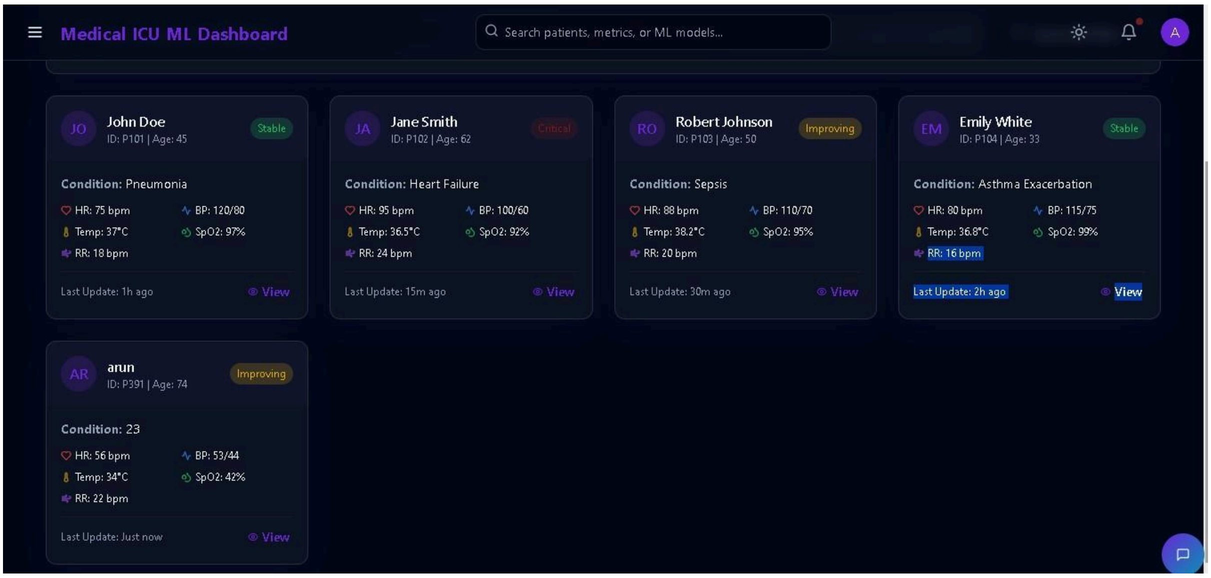
Heart Rate (bpm): e.g., 70

Blood Pressure (mmHg): e.g., 120/80

Temperature (°C): e.g., 37.0

SpO2 (%): e.g., 98

Respiratory Rate (bpm): e.g., 16



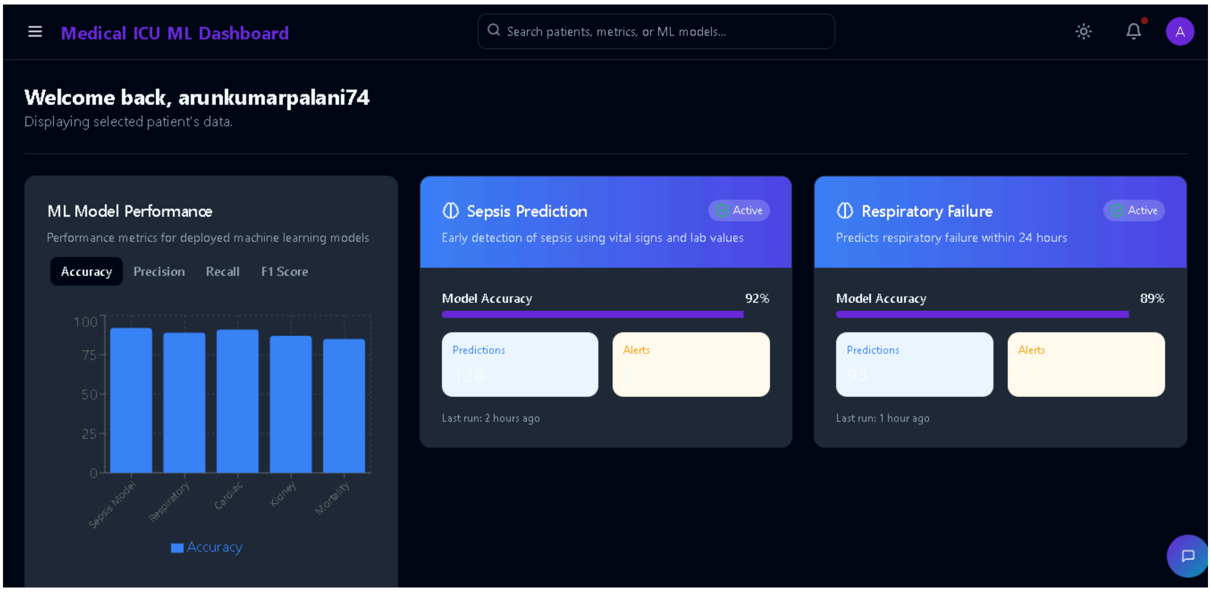
JO John Doe — Stable
ID: P101 | Age: 45
Condition: Pneumonia
HR: 75 bpm | BP: 120/80
Temp: 37°C | SpO2: 97%
RR: 18 bpm
Last Update: 1h ago — View

JA Jane Smith — Critical
ID: P102 | Age: 62
Condition: Heart Failure
HR: 95 bpm | BP: 100/60
Temp: 36.5°C | SpO2: 92%
RR: 24 bpm
Last Update: 15m ago — View

RO Robert Johnson — Improving
ID: P103 | Age: 50
Condition: Sepsis
HR: 88 bpm | BP: 110/70
Temp: 38.2°C | SpO2: 95%
RR: 20 bpm
Last Update: 30m ago — View

EM Emily White — Stable
ID: P104 | Age: 33
Condition: Asthma Exacerbation
HR: 80 bpm | BP: 115/75
Temp: 36.8°C | SpO2: 99%
RR: 16 bpm
Last Update: 2h ago — View

AR arun — Improving
ID: P391 | Age: 74
Condition: 23
HR: 56 bpm | BP: 53/44
Temp: 34°C | SpO2: 42%
RR: 22 bpm
Last Update: Just now — View



Welcome back, arunkumarpalani74

Displaying selected patient's data.

ML Model Performance

Performance metrics for deployed machine learning models

Sepsis Prediction — Active

Early detection of sepsis using vital signs and lab values

Model Accuracy 92%

Predictions 128

Alerts 3

Last run: 2 hours ago

Respiratory Failure — Active

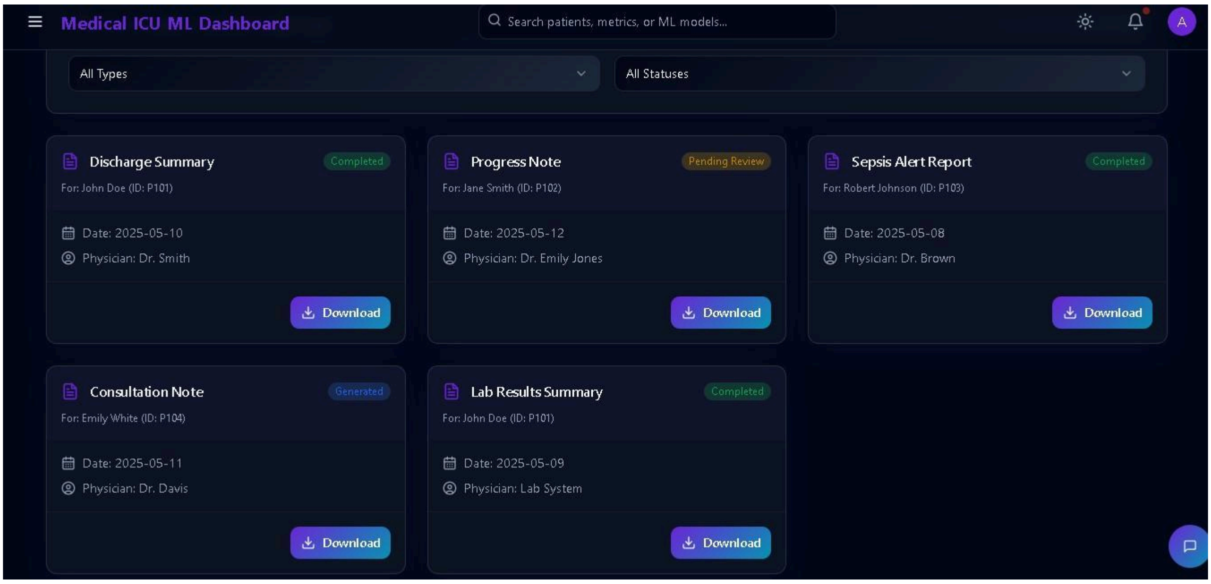
Predicts respiratory failure within 24 hours

Model Accuracy 89%

Predictions 95

Alerts 1

Last run: 1 hour ago



Discharge Summary — Completed

For: John Doe (ID: P101)

Date: 2025-05-10

Physician: Dr. Smith

Progress Note — Pending Review

For: Jane Smith (ID: P102)

Date: 2025-05-12

Physician: Dr. Emily Jones

Sepsis Alert Report — Completed

For: Robert Johnson (ID: P103)

Date: 2025-05-08

Physician: Dr. Brown

Consultation Note — Generated

For: Emily White (ID: P104)

Date: 2025-05-11

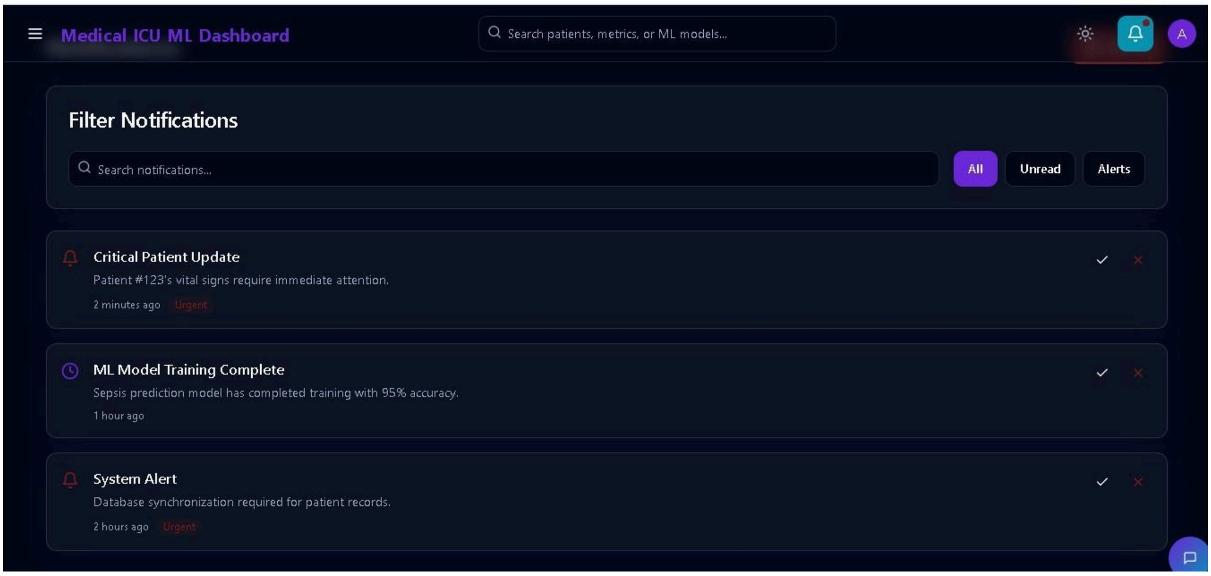
Physician: Dr. Davis

Lab Results Summary — Completed

For: John Doe (ID: P101)

Date: 2025-05-09

Physician: Lab System



Filter Notifications

Search notifications...

Critical Patient Update

Patient #123's vital signs require immediate attention.

2 minutes ago Urgent

ML Model Training Complete

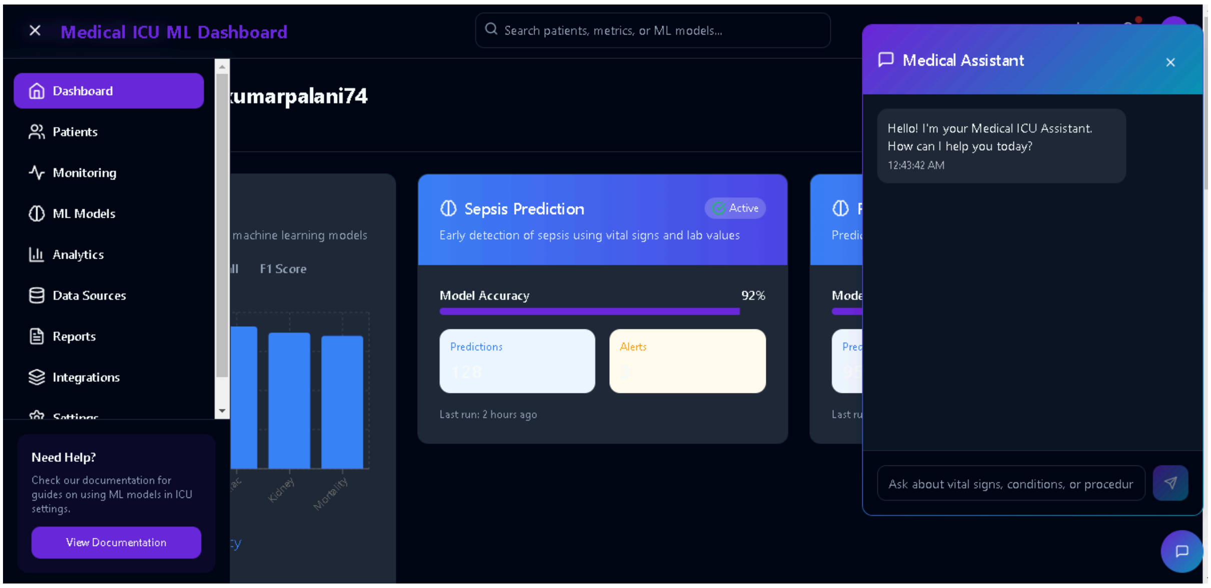
Sepsis prediction model has completed training with 95% accuracy.

1 hour ago

System Alert

Database synchronization required for patient records.

2 hours ago Urgent



kumarpalani74

machine learning models

Need Help?

Check our documentation for guides on using ML models in ICU settings.

Sepsis Prediction

Early detection of sepsis using vital signs and lab values

Model Accuracy 92%

Predictions 128

Alerts

Last run: 2 hours ago

Medical Assistant

Hello! I'm your Medical ICU Assistant. How can I help you today?

12:43:42 AM

Ask about vital signs, conditions, or procedur