

```
<style>
/* ===== GLOBAL ===== */
:root {
    --top-offset: 140px;
    --sidebar-width: 220px;
    --primary-color: #007bff;
    --secondary-color: #0056b3;
    --success-color: #00ff00;
    --danger-color: #ff0000;
    --warning-color: #ff9900;
    --info-color: #00ccff;
    --purple-color: #9d4edd;
    --dark-bg: #0d0d0d;
    --panel-bg: #141414;
    --border-color: #222;
    --accent-color: #00ccff;
}

* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: var(--dark-bg);
    color: #fff;
    min-height: 100vh;
    overflow-x: hidden;
}

.hidden { display: none !important; }

/* ===== HEADER ===== */
.header {
    background: linear-gradient(135deg, #000 0%, #222 100%);
    padding: 15px 20px;
```

```
display: flex;
justify-content: space-between;
align-items: center;
position: relative;
border-bottom: 2px solid var(--accent-color);
height: 100px;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.5);
position: fixed;
top: 0;
left: 0;
right: 0;
z-index: 1000;
}

.app-title {
  display: flex;
  align-items: center;
  gap: 15px;
}

.bull, .bear {
  font-size: 42px;
  animation: pulse 2s infinite;
}

.bull {
  color: var(--success-color);
  text-shadow: 0 0 10px rgba(0, 255, 0, 0.5);
}

.bear {
  color: var(--danger-color);
  text-shadow: 0 0 10px rgba(255, 0, 0, 0.5);
}

@keyframes pulse {
  0%, 100% { transform: scale(1); }
  50% { transform: scale(1.1); }
}
```

```
.title-text {  
    font-size: 46px;  
    font-weight: 900;  
    letter-spacing: 3px;  
    background: linear-gradient(90deg, #00ffcc, #00ff00, #00ccff, #ff00ff);  
    -webkit-background-clip: text;  
    -webkit-text-fill-color: transparent;  
    background-clip: text;  
    text-shadow: 0 2px 10px rgba(0, 0, 0, 0.5);  
}  
  
/* ===== LIVE WATCH ===== */  
.live-watch-container {  
    position: absolute;  
    right: 20px;  
    top: 50%;  
    transform: translateY(-50%);  
    text-align: center;  
}  
  
.live-watch {  
    font-size: 14px;  
    color: #aaa;  
    margin-bottom: 2px;  
}  
  
.digital-clock {  
    font-size: 22px;  
    font-weight: bold;  
    color: var(--accent-color);  
    text-shadow: 0 0 10px rgba(0, 204, 255, 0.5);  
    background: rgba(0, 0, 0, 0.3);  
    padding: 5px 12px;  
    border-radius: 8px;  
    border: 1px solid rgba(0, 204, 255, 0.3);  
}  
  
.user-box {  
    position: absolute;
```

```
right: 200px;
top: 50%;
transform: translateY(-50%);
text-align: right;
}

.user-name {
    font-size: 25px;
    font-weight: bold;
    color: #fff;
    margin-bottom: 5px;
}

.pl-box {
    margin-top: 6px;
    padding: 8px 16px;
    font-size: 20px;
    font-weight: bold;
    border-radius: 10px;
    background: linear-gradient(135deg, #111 0%, #333 100%);
    display: inline-block;
    transition: all 0.3s ease;
    border: 1px solid rgba(255, 255, 255, 0.1);
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
}

/* ===== ENHANCED HORIZONTAL MARKET TICKER ===== */
.market-ticker-container {
    background: linear-gradient(90deg, #111 0%, #222 50%, #111 100%);
    padding: 10px 0;
    border-bottom: 2px solid rgba(0, 204, 255, 0.5);
    overflow: hidden;
    position: fixed;
    top: 100px;
    left: 0;
    right: 0;
    height: 60px;
    z-index: 999;
}
```

```
.ticker-header {  
    position: absolute;  
    left: 0;  
    top: 0;  
    background: linear-gradient(90deg, #ff0000, #ff9900);  
    color: white;  
    padding: 10px 20px;  
    font-weight: bold;  
    z-index: 10;  
    border-right: 2px solid rgba(255,255,255,0.3);  
}  
  
.market-ticker {  
    display: flex;  
    animation: tickerScroll 30s linear infinite;  
    white-space: nowrap;  
    padding-left: 120px;  
}  
  
.ticker-item-horizontal {  
    display: flex;  
    align-items: center;  
    padding: 0 25px;  
    border-right: 1px solid rgba(255, 255, 255, 0.1);  
    background: rgba(0, 0, 0, 0.3);  
    margin: 5px 0;  
    border-radius: 5px;  
    position: relative;  
    transition: all 0.3s ease;  
}  
  
.ticker-item-horizontal:hover {  
    background: rgba(0, 204, 255, 0.2);  
    transform: scale(1.05);  
}  
  
.ticker-symbol {  
    color: #fff;
```

```
font-weight: bold;
margin-right: 8px;
min-width: 80px;
font-size: 16px;
}

.ticker-price {
  color: var(--success-color);
  font-weight: bold;
  font-size: 18px;
  margin-right: 5px;
  text-shadow: 0 0 5px currentColor;
}

.ticker-change {
  color: var(--danger-color);
  font-size: 14px;
  font-weight: bold;
  padding: 2px 8px;
  border-radius: 3px;
  background: rgba(0,0,0,0.3);
}

.positive {
  color: var(--success-color) !important;
  background: rgba(0, 255, 0, 0.1) !important;
}

.negative {
  color: var(--danger-color) !important;
  background: rgba(255, 0, 0, 0.1) !important;
}

.delete-ticker {
  background: rgba(255, 0, 0, 0.7);
  color: white;
  border: none;
  border-radius: 50%;
  width: 20px;
  height: 20px;
```

```
margin-left: 10px;
cursor: pointer;
font-size: 12px;
font-weight: bold;
display: flex;
align-items: center;
justify-content: center;
opacity: 0;
transition: all 0.3s ease;
}

.ticker-item-horizontal:hover .delete-ticker {
  opacity: 1;
}

.delete-ticker:hover {
  background: rgba(255, 0, 0, 1);
  transform: scale(1.2);
}

@keyframes tickerScroll {
  0% { transform: translateX(0); }
  100% { transform: translateX(-50%); }
}

/* ===== SEPARATED NAVIGATION POPUP (LEFT SIDE) ===== */
.nav-hover-trigger {
  position: fixed;
  top: 180px;
  left: 0;
  background: linear-gradient(135deg, var(--accent-color), #0066cc);
  color: white;
  border: none;
  padding: 12px 15px;
  border-radius: 0 10px 10px 0;
  cursor: pointer;
  z-index: 999;
  font-size: 20px;
  transition: all 0.3s ease;
```

```
        box-shadow: 2px 0 10px rgba(0,0,0,0.3);
    }

.nav-hover-trigger:hover {
    padding-left: 20px;
    background: linear-gradient(135deg, #00ccff, #0099ff);
}

.nav-popup {
    position: fixed;
    top: 180px;
    left: -300px;
    width: 300px;
    background: linear-gradient(180deg, #141414 0%, #1a1a1a 100%);
    border-radius: 0 15px 15px 0;
    padding: 15px 10px;
    overflow-y: auto;
    border-right: 2px solid rgba(0, 204, 255, 0.5);
    border-bottom: 2px solid rgba(0, 204, 255, 0.5);
    border-top: 2px solid rgba(0, 204, 255, 0.5);
    box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.5);
    transition: left 0.3s ease;
    z-index: 998;
    max-height: 80vh;
}

.nav-popup.show {
    left: 0;
}

.nav-popup-header {
    color: var(--accent-color);
    font-size: 18px;
    font-weight: bold;
    margin-bottom: 15px;
    padding-bottom: 10px;
    border-bottom: 2px solid rgba(0, 204, 255, 0.5);
    text-align: center;
    text-transform: uppercase;
```

```
letter-spacing: 2px;  
cursor: pointer;  
user-select: none;  
}  
  
.nav-popup button {  
width: 100%;  
margin-bottom: 8px;  
padding: 12px 15px;  
background: linear-gradient(135deg, #2a2a2a 0%, #1a1a1a 100%);  
border: 1px solid rgba(255, 255, 255, 0.1);  
color: #fff;  
cursor: pointer;  
border-radius: 8px;  
font-size: 14px;  
text-align: left;  
transition: all 0.2s ease;  
display: flex;  
align-items: center;  
gap: 10px;  
position: relative;  
overflow: hidden;  
}  
  
.nav-popup button::before {  
content: " ";  
position: absolute;  
left: 0;  
top: 0;  
width: 5px;  
height: 100%;  
background: var(--accent-color);  
transform: scaleY(0);  
transition: transform 0.3s ease;  
}  
  
.nav-popup button i {  
font-size: 16px;  
width: 20px;
```

```
text-align: center;
color: var(--accent-color);
}

.nav-popup button:hover {
background: linear-gradient(135deg, var(--primary-color) 0%, var(--secondary-c
transform: translateX(5px);
box-shadow: 0 4px 12px rgba(0, 123, 255, 0.4);
}

.nav-popup button:hover::before {
transform: scaleY(1);
}

.nav-popup button.active {
background: linear-gradient(135deg, var(--primary-color) 0%, var(--secondary-c
border-color: var(--accent-color);
}

.nav-popup button.active::before {
transform: scaleY(1);
}

/* ===== SEPARATE WATCHLIST PANEL (RIGHT SIDE - ALWAYS VISIBLE) ===== */
.watchlist-panel {
position: fixed;
top: 160px;
right: 0;
width: 320px;
height: calc(100vh - 160px);
background: linear-gradient(180deg, #141414 0%, #1a1a1a 100%);
border-left: 2px solid rgba(0, 204, 255, 0.5);
padding: 15px;
overflow-y: auto;
overflow-x: hidden;
z-index: 997;
box-shadow: -5px 0 15px rgba(0, 0, 0, 0.5);
}
```

```
.watchlist-panel-header {  
    color: var(--accent-color);  
    font-size: 18px;  
    font-weight: bold;  
    margin-bottom: 15px;  
    padding-bottom: 10px;  
    border-bottom: 2px solid rgba(0, 204, 255, 0.5);  
    text-align: center;  
    text-transform: uppercase;  
    letter-spacing: 1px;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}  
}
```

```
.watchlist-panel .watchlist-container {  
    margin: 0;  
    background: transparent;  
    border-radius: 0;  
    padding: 0;  
    border: none;  
}  
}
```

```
.watchlist-panel .watchlist-header {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    margin-bottom: 15px;  
    padding-bottom: 10px;  
    border-bottom: 1px solid rgba(255, 255, 255, 0.1);  
}  
}
```

```
.watchlist-panel .watchlist-header h3 {  
    color: var(--accent-color);  
    font-size: 16px;  
    margin: 0;  
    display: flex;  
    align-items: center;  
    gap: 8px;  
}
```

```
}

.watchlist-panel .clear-watchlist-btn {
    background: none;
    border: none;
    color: #ff4444;
    cursor: pointer;
    font-size: 12px;
    padding: 4px 8px;
    border-radius: 3px;
}

.watchlist-panel .clear-watchlist-btn:hover {
    background: rgba(255, 68, 68, 0.1);
}

.watchlist-panel .search-container {
    display: flex;
    gap: 8px;
    margin-bottom: 15px;
}

.watchlist-panel .search-container input {
    flex: 1;
    padding: 10px;
    background: #333;
    border: none;
    color: #fff;
    border-radius: 5px;
    font-size: 13px;
}

.watchlist-panel .search-container button {
    padding: 10px 12px;
    background: #0077ff;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
```

```
        font-size: 13px;
        font-weight: bold;
    }

.watchlist-panel .search-container button:hover {
    background: #0055cc;
}

.watchlist-panel .watchlist-stocks {
    max-height: calc(100vh - 320px);
    overflow-y: auto;
    overflow-x: hidden;
}

.watchlist-panel .watchlist-item {
    background: #2f2f2f;
    padding: 10px;
    margin-bottom: 8px;
    border-radius: 5px;
    border: 1px solid #3a3a3a;
    position: relative;
    transition: all 0.3s ease;
}

.watchlist-panel .watchlist-item:hover {
    background: #3a3a3a;
    border-color: #0077ff;
}

.watchlist-panel .watchlist-symbol {
    font-weight: bold;
    font-size: 14px;
    cursor: pointer;
    padding: 4px 0;
    color: #fff;
    display: flex;
    justify-content: space-between;
    align-items: center;
}
```

```
.watchlist-panel .watchlist-name {  
    font-size: 12px;  
    color: #aaa;  
    margin-top: 2px;  
}  
  
.watchlist-panel .delete-watchlist-btn {  
    background: rgba(255, 68, 68, 0.2);  
    border: 1px solid #ff4444;  
    color: #ff4444;  
    cursor: pointer;  
    font-size: 12px;  
    width: 22px;  
    height: 22px;  
    border-radius: 50%;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    transition: all 0.3s ease;  
}  
  
.watchlist-panel .delete-watchlist-btn:hover {  
    background: rgba(255, 68, 68, 0.4);  
    transform: scale(1.1);  
}  
  
.watchlist-panel .no-watchlist-stocks {  
    text-align: center;  
    color: #888;  
    padding: 20px 0;  
    font-style: italic;  
    font-size: 13px;  
}  
  
.watchlist-panel .watchlist-buttons {  
    display: flex;  
    flex-wrap: wrap;  
    gap: 4px;  
}
```

```
margin-top: 8px;  
padding-top: 8px;  
border-top: 1px solid #444;  
}  
  
.watchlist-panel .watchlist-btn {  
flex: 1;  
min-width: 45px;  
padding: 6px 8px;  
font-size: 11px;  
border: none;  
border-radius: 3px;  
cursor: pointer;  
color: #fff;  
text-align: center;  
font-weight: bold;  
}  
  
.watchlist-panel .watchlist-buy-btn {  
background: #00aa66;  
}  
  
.watchlist-panel .watchlist-sell-btn {  
background: #ff4444;  
}  
  
.watchlist-panel .watchlist-chart-btn {  
background: #0077ff;  
}  
  
.watchlist-panel .watchlist-depth-btn {  
background: #888;  
}  
  
.watchlist-panel .watchlist-more-btn {  
background: #555;  
position: relative;  
}
```

```
.watchlist-panel .watchlist-more-menu {  
    position: absolute;  
    bottom: 100%;  
    left: 0;  
    background: #111;  
    display: none;  
    border-radius: 4px;  
    z-index: 100;  
    min-width: 120px;  
}  
  
.watchlist-panel .watchlist-more-menu div {  
    padding: 6px 10px;  
    font-size: 11px;  
    cursor: pointer;  
    white-space: nowrap;  
}  
  
.watchlist-panel .watchlist-more-menu div:hover {  
    background: #444;  
}  
  
.watchlist-panel .watchlist-more-btn:hover .watchlist-more-menu {  
    display: block;  
}  
  
/* ===== LARGER CONTENT (ADJUSTED FOR WATCHLIST PANEL) ===== */  
.content {  
    margin-top: 160px;  
    margin-left: 0;  
    margin-right: 320px; /* Space for watchlist panel */  
    padding: 20px;  
    min-height: calc(100vh - var(--top-offset));  
    transition: all 0.3s ease;  
    overflow: auto;  
    position: relative;  
    width: calc(100% - 320px); /* Adjusted for watchlist panel */  
    height: calc(100vh - 160px);  
    overflow-y: auto;
```

```
        overflow-x: hidden;
    }

/* ===== ENHANCED OPEN POSITIONS CONTAINER ===== */
.open-positions-container {
    display: flex;
    gap: 20px;
    min-height: calc(100vh - var(--top-offset) - 40px);
    height: auto;
    width: 100%;
    overflow: visible;
}

/* ===== ENHANCED POSITIONS MAIN CONTENT ===== */
.positions-main {
    flex: 1;
    background: #1a1a1a;
    border-radius: 10px;
    padding: 25px;
    border: 1px solid #333;
    min-height: calc(100vh - 220px);
    overflow: auto;
}

.positions-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 25px;
}

.positions-header h2 {
    color: var(--accent-color);
    font-size: 28px;
    margin: 0;
    display: flex;
    align-items: center;
    gap: 10px;
}
```

```
.add-position-btn {  
    background: linear-gradient(135deg, #00cc00, #009900);  
    color: white;  
    border: none;  
    padding: 12px 24px;  
    border-radius: 5px;  
    cursor: pointer;  
    font-weight: bold;  
    font-size: 16px;  
    display: flex;  
    align-items: center;  
    gap: 8px;  
}  
  
.add-position-btn:hover {  
    background: linear-gradient(135deg, #00ff00, #00cc00);  
}  
  
/* Table Styles */  
.positions-table {  
    width: 100%;  
    border-collapse: collapse;  
    background: #2b2b2b;  
    font-size: 14px;  
    min-width: 1500px;  
}  
  
.positions-table th {  
    background: #4a4a4a;  
    padding: 15px 10px;  
    border: 1px solid #555;  
    text-align: center;  
    font-weight: bold;  
    color: var(--accent-color);  
    font-size: 15px;  
    white-space: nowrap;  
}
```

```
.positions-table td {  
    padding: 12px 10px;  
    border: 1px solid #555;  
    text-align: center;  
    font-size: 14px;  
}  
  
.positions-table tr:hover {  
    background: #3a3a3a;  
}  
  
/* Order Type Styling */  
.order-type-buy {  
    background: #00aa66;  
    color: #fff;  
    padding: 6px 10px;  
    border-radius: 4px;  
    font-size: 12px;  
    font-weight: bold;  
}  
  
.order-type-sell {  
    background: #ff4444;  
    color: #fff;  
    padding: 6px 10px;  
    border-radius: 4px;  
    font-size: 12px;  
    font-weight: bold;  
}  
  
/* Order Condition Styling */  
.order-condition {  
    padding: 6px 10px;  
    border-radius: 4px;  
    font-size: 12px;  
    font-weight: bold;  
}  
  
.condition-intraday {
```

```
background: #0077ff;
color: #fff;
}

.condition-positional {
background: #aa00ff;
color: #fff;
}

.condition-bo {
background: #ff8800;
color: #fff;
}

.condition-st {
background: #ff0088;
color: #fff;
}

.condition-msl {
background: #008888;
color: #fff;
}

/* Qty Columns */
.buy-qty {
color: #00ff88;
font-weight: bold;
}

.sell-qty {
color: #ff4444;
font-weight: bold;
}

/* P&L Colors */
.pl-positive {
color: #00ff88;
font-weight: bold;
```

```
}

.pl-negative {
  color: #ff4444;
  font-weight: bold;
}

/* Symbol Quick Menu */
.symbol-cell {
  position: relative;
  cursor: pointer;
}

.quick-popup {
  position: absolute;
  top: 100%;
  left: 0;
  background: #1f1f1f;
  padding: 8px;
  border-radius: 5px;
  display: none;
  gap: 4px;
  z-index: 100;
}

.symbol-cell:hover .quick-popup {
  display: flex;
}

.q-btn {
  padding: 6px 10px;
  font-size: 12px;
  border: none;
  border-radius: 3px;
  cursor: pointer;
  color: #fff;
}

.q-btn.buy {
```

```
background: #00aa66;  
}  
  
.q-btn.sell {  
    background: #ff4444;  
}  
  
.q-btn.chart {  
    background: #0077ff;  
}  
  
.q-btn.depth {  
    background: #888;  
}  
  
.q-btn.delete-btn-main {  
    background: #aa0000;  
}  
  
.q-btn.more {  
    background: #555;  
    position: relative;  
}  
  
.more-menu-main {  
    position: absolute;  
    top: 100%;  
    left: 0;  
    background: #111;  
    display: none;  
    border-radius: 4px;  
}  
  
.more-menu-main div {  
    padding: 8px 12px;  
    font-size: 12px;  
    cursor: pointer;  
}
```

```
.more-menu-main div:hover {  
    background: #444;  
}  
  
.q-btn.more:hover .more-menu-main {  
    display: block;  
}  
  
/* Source Menu */  
.source-wrap {  
    position: relative;  
}  
  
.source-btn {  
    background: #0077ff;  
    border: none;  
    color: #fff;  
    padding: 6px 12px;  
    border-radius: 4px;  
    cursor: pointer;  
    font-size: 12px;  
}  
  
.source-menu {  
    position: absolute;  
    top: 100%;  
    left: 0;  
    background: #1f1f1f;  
    display: none;  
}  
  
.source-menu div {  
    padding: 8px 12px;  
    font-size: 12px;  
    cursor: pointer;  
}  
  
.source-menu div:hover {  
    background: #444;
```

```
}

.source-wrap:hover .source-menu {
  display: block;
}

/* Action Button */
.action-btn {
  background: #555;
  border: none;
  color: #fff;
  padding: 8px 12px;
  border-radius: 4px;
  cursor: pointer;
  font-size: 12px;
  font-weight: bold;
}

.action-btn:hover {
  background: #666;
}

/* ===== LARGER MODALS ===== */
.modal {
  position: fixed;
  inset: 0;
  background: rgba(0, 0, 0, 0.8);
  display: none;
  justify-content: center;
  align-items: center;
  z-index: 9999;
  overflow: auto;
  padding: 20px;
}

.modal-box {
  width: 600px;
  max-width: 95%;
  max-height: 90vh;
```

```
background: #2f2f2f;
border-radius: 10px;
padding: 20px;
border: 2px solid var(--accent-color);
overflow-y: auto;
overflow-x: hidden;
}

.modal-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 20px;
}

.modal-header h3 {
  margin: 0;
  font-size: 22px;
  color: var(--accent-color);
}

.modal-header button {
  background: none;
  border: none;
  color: #fff;
  font-size: 24px;
  cursor: pointer;
}

.tabs {
  display: flex;
  margin-top: 10px;
  background: #333;
  border-radius: 5px;
  overflow: hidden;
}

.tabs button {
  flex: 1;
```

```
background: none;
border: none;
padding: 12px;
color: #fff;
cursor: pointer;
font-size: 14px;
transition: all 0.3s ease;
}

.tabs button:hover {
  background: #444;
}

.tabs button.active {
  background: #0077ff;
  font-weight: bold;
}

.tab-content {
  display: none;
  margin-top: 15px;
}

.tab-content.active {
  display: block;
}

.modal-box input,
.modal-box select,
.modal-box textarea {
  width: 100%;
  padding: 10px;
  margin: 8px 0;
  background: #444;
  border: none;
  color: #fff;
  font-size: 14px;
  border-radius: 4px;
}
```

```
.modal-box textarea {  
    height: 120px;  
    resize: vertical;  
    font-family: 'Courier New', monospace;  
}  
  
.full-btn {  
    width: 100%;  
    padding: 12px;  
    margin-top: 15px;  
    border: none;  
    color: #fff;  
    cursor: pointer;  
    font-size: 16px;  
    font-weight: bold;  
    border-radius: 4px;  
}  
  
.success {  
    background: #00aa66;  
}  
  
.danger {  
    background: #ff4444;  
}  
  
.info {  
    background: #0077ff;  
}  
  
/* ===== NEW POSITION SECTION STYLES ===== */  
.position-section {  
    margin: 15px 0;  
    background: #222;  
    border-radius: 8px;  
    border-left: 4px solid var(--accent-color);  
    overflow: hidden;  
}
```

```
.position-section-header {  
    padding: 15px 20px;  
    background: rgba(0, 204, 255, 0.1);  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    cursor: pointer;  
    user-select: none;  
}  
  
.position-section-header h4 {  
    color: var(--accent-color);  
    margin: 0;  
    display: flex;  
    align-items: center;  
    gap: 8px;  
    font-size: 16px;  
}  
  
.position-section-toggle {  
    background: none;  
    border: none;  
    color: var(--accent-color);  
    cursor: pointer;  
    font-size: 14px;  
    padding: 6px 10px;  
    border-radius: 3px;  
    transition: all 0.3s ease;  
}  
  
.position-section-toggle:hover {  
    background: rgba(0, 204, 255, 0.2);  
}  
  
.position-section-body {  
    padding: 20px;  
    max-height: 400px;  
    overflow-y: auto;
```

```
        transition: all 0.3s ease;
    }

.position-section-body.collapsed {
    display: none;
}

/* ====== ENABLE/DISABLE OPTIONS WITH MOUSE CLICK ====== */
.enable-option {
    background: rgba(0, 204, 255, 0.1);
    padding: 12px 15px;
    border-radius: 5px;
    margin: 10px 0;
    cursor: pointer;
    transition: all 0.3s ease;
    border-left: 4px solid transparent;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.enable-option:hover {
    background: rgba(0, 204, 255, 0.2);
    border-left-color: var(--accent-color);
}

.enable-option.active {
    background: rgba(0, 204, 255, 0.3);
    border-left-color: var(--success-color);
}

.enable-status {
    width: 24px;
    height: 24px;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
}
```

```
.enable-status.enabled {  
  background: var(--success-color);  
  color: #000;  
}  
  
.enable-status.disabled {  
  background: var(--danger-color);  
  color: #fff;  
}  
  
/* ===== PARTITION & ADD QUANTITY SECTION STYLES ===== */  
.partition-section {  
  margin: 15px 0;  
  background: #222;  
  border-radius: 8px;  
  border-left: 4px solid var(--warning-color);  
  overflow: hidden;  
}  
  
.partition-section-header {  
  padding: 15px 20px;  
  background: rgba(255, 153, 0, 0.1);  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  cursor: pointer;  
  user-select: none;  
}  
  
.partition-section-header h4 {  
  color: var(--warning-color);  
  margin: 0;  
  display: flex;  
  align-items: center;  
  gap: 8px;  
  font-size: 16px;  
}
```

```
.partition-section-toggle {  
    background: none;  
    border: none;  
    color: var(--warning-color);  
    cursor: pointer;  
    font-size: 14px;  
    padding: 6px 10px;  
    border-radius: 3px;  
    transition: all 0.3s ease;  
}  
  
.partition-section-toggle:hover {  
    background: rgba(255, 153, 0, 0.2);  
}  
  
.partition-section-body {  
    padding: 20px;  
    transition: all 0.3s ease;  
}  
  
.partition-section-body.collapsed {  
    display: none;  
}  
  
.partition-grid {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    gap: 20px;  
}  
  
.partition-column {  
    display: flex;  
    flex-direction: column;  
    gap: 15px;  
}  
  
.partition-item {  
    background: #333;  
    padding: 15px;  
}
```

```
border-radius: 5px;
border: 1px solid #444;
}

.partition-item h5 {
  color: var(--accent-color);
  font-size: 14px;
  margin-bottom: 10px;
  display: flex;
  align-items: center;
  gap: 5px;
}

.partition-row {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 10px;
}

/* ===== ENHANCED SCROLLING ===== */
.content::-webkit-scrollbar {
  width: 12px;
}

.content::-webkit-scrollbar-track {
  background: rgba(0, 0, 0, 0.3);
  border-radius: 5px;
}

.content::-webkit-scrollbar-thumb {
  background: linear-gradient(135deg, var(--accent-color), #0066cc);
  border-radius: 5px;
}

.content::-webkit-scrollbar-thumb:hover {
  background: linear-gradient(135deg, #00ccff, #0099ff);
}

/* Position Section Scrollbar */
```

```
.position-section-body::-webkit-scrollbar {  
    width: 8px;  
}  
  
.position-section-body::-webkit-scrollbar-track {  
    background: rgba(0, 0, 0, 0.2);  
    border-radius: 3px;  
}  
  
.position-section-body::-webkit-scrollbar-thumb {  
    background: var(--accent-color);  
    border-radius: 3px;  
}  
  
/* Watchlist Scrollbar */  
.watchlist-panel .watchlist-stocks::-webkit-scrollbar {  
    width: 6px;  
}  
  
.watchlist-panel .watchlist-stocks::-webkit-scrollbar-track {  
    background: rgba(0, 0, 0, 0.3);  
    border-radius: 3px;  
}  
  
.watchlist-panel .watchlist-stocks::-webkit-scrollbar-thumb {  
    background: #0077ff;  
    border-radius: 3px;  
}  
  
/* Modal Scrollbar */  
.modal-box::-webkit-scrollbar {  
    width: 8px;  
}  
  
.modal-box::-webkit-scrollbar-track {  
    background: rgba(0, 0, 0, 0.2);  
    border-radius: 3px;  
}
```

```
.modal-box::-webkit-scrollbar-thumb {  
    background: var(--warning-color);  
    border-radius: 3px;  
}  
  
/* ===== RESPONSIVE ===== */  
@media (max-width: 1200px) {  
    .open-positions-container {  
        flex-direction: column;  
    }  
  
    .positions-main {  
        min-height: auto;  
    }  
}  
  
@media (max-width: 1100px) {  
    .watchlist-panel {  
        width: 280px;  
    }  
  
    .content {  
        margin-right: 280px;  
        width: calc(100% - 280px);  
    }  
}  
  
@media (max-width: 900px) {  
    .content {  
        margin-top: 160px;  
        padding: 15px;  
    }  
  
    .watchlist-panel {  
        display: none; /* Hide watchlist on smaller screens */  
    }  
  
    .content {  
        margin-right: 0;  
    }  
}
```

```
width: 100%;  
}  
  
.nav-hover-trigger {  
    top: 160px;  
    padding: 10px 12px;  
    font-size: 18px;  
}  
  
.nav-popup {  
    top: 160px;  
    width: 250px;  
}  
}  
  
@media (max-width: 768px) {  
    .header {  
        flex-direction: column;  
        height: auto;  
        padding: 15px;  
    }  
  
    .app-title {  
        margin-bottom: 15px;  
    }  
  
    .live-watch-container {  
        position: relative;  
        right: auto;  
        top: auto;  
        transform: none;  
        margin: 10px 0;  
    }  
  
    .user-box {  
        position: relative;  
        right: auto;  
        top: auto;  
        transform: none;  
    }
```

```
    text-align: center;
    margin-top: 10px;
}

.content {
    margin: 10px;
    margin-top: 160px;
}

.nav-popup button {
    padding: 10px 12px;
    font-size: 13px;
}

.positions-table {
    font-size: 12px;
}

.positions-table th,
.positions-table td {
    padding: 8px 6px;
}

.q-btn {
    padding: 5px 8px;
    font-size: 11px;
}

.modal-box {
    width: 95%;
    margin: 10px;
}

.section-grid,
.partition-grid {
    grid-template-columns: 1fr;
}

.position-row {
```

```
flex-direction: column;
align-items: flex-start;
}

.position-toggle {
width: 100%;
justify-content: space-between;
}

.position-controls {
width: 100%;
}
}

@media (max-width: 480px) {
.title-text {
font-size: 32px;
}

.bull, .bear {
font-size: 32px;
}

.user-name {
font-size: 20px;
}

.pl-box {
font-size: 16px;
padding: 6px 12px;
}

.partition-row {
grid-template-columns: 1fr;
}

.nav-popup {
width: 220px;
}
```

```

        }
    </style>
<!-- Font Awesome for icons -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.

```

MUKESH ALGO

```

<!-- LIVE WATCH -->
<div class="live-watch-container">
    <div class="live-watch">LIVE</div>
    <div class="digital-clock" id="digitalClock">00:00:00</div>
</div>

<!-- USER BOX -->
<div class="user-box">
    <div class="user-name">Mukeshbhai</div>
    <div class="pl-box" id="plBox">P/L : 1,25,430</div>
</div>
</div>

<!-- ===== ENHANCED HORIZONTAL MARKET TICKER ===== -->
<div class="market-ticker-container">
    <div class="ticker-header">LIVE MARKET</div>
    <div class="market-ticker" id="marketTicker">
        <!-- Ticker items will be populated by JavaScript -->
    </div>
</div>

<!-- ===== SEPARATE NAVIGATION POPUP (LEFT SIDE) ===== -->
<button class="nav-hover-trigger" id="navHoverTrigger" onmouseenter="showNav
    <i class="fas fa-bars"></i>
</button>

<div class="nav-popup" id="navPopup" onmouseleave="hideNavPopup()">
    <div class="nav-popup-header">
        <i class="fas fa-bars"></i> NAVIGATION
    </div>
    <button onclick="showPage('openPositionsPage')">
        <i class="fas fa-chart-line"></i> Open Positions

```

```
</button>
<button onclick="showPage('orderbookPage')">
  <i class="fas fa-book"></i> Order Book
</button>
<button onclick="openScreenerPopup()">
  <i class="fas fa-search"></i> Screener
</button>
<button onclick="openStrategyPopup()">
  <i class="fas fa-chess-board"></i> Strategy Creation
</button>
<button onclick="showPage('reportPage')">
  <i class="fas fa-chart-bar"></i> Reports
</button>
<button onclick="showPage('profilePage')">
  <i class="fas fa-user"></i> Profile
</button>
<button onclick="showPage('brokerPage')">
  <i class="fas fa-university"></i> Broker
</button>
<button onclick="showPage('marketplacePage')">
  <i class="fas fa-store"></i> Market Place
</button>
<button onclick="showPage('paperTradingPage')">
  <i class="fas fa-file-invoice-dollar"></i> Paper Trading
</button>
<button onclick="showPage('myPlanPage')">
  <i class="fas fa-calendar-alt"></i> My Plan
</button>
<button onclick="showPage('backtestPage')">
  <i class="fas fa-history"></i> Backtest
</button>
<button onclick="showPage('settingsPage')">
  <i class="fas fa-cog"></i> Settings
</button>
<button onclick="showPage('alPage')">
  <i class="fas fa-robot"></i> AL
</button>

<!-- Dashboard Controls -->
```

```

<div style="margin-top: 20px; padding: 10px; background: rgba(0,0,0,0.3); border
<h4 style="color: var(--accent-color); font-size: 14px; margin-bottom: 10px; text
    <i class="fas fa-expand"></i> VIEW CONTROLS
</h4>
<button onclick="fullScreenMode()" style="background: linear-gradient(135deg
    <i class="fas fa-expand"></i> Full Screen
</button>
<button onclick="resetAllData()" style="background: linear-gradient(135deg, #f
    <i class="fas fa-redo"></i> Reset Data
</button>
</div>

<button onclick="logOut()" style="margin-top: 20px; background: linear-gradient(
    <i class="fas fa-sign-out-alt"></i> Log Out
</button>
</div>

<!-- ===== SEPARATE WATCHLIST PANEL (RIGHT SIDE - ALWAYS VISIBLE) ===== -->
<div class="watchlist-panel" id="watchlistPanel">
    <div class="watchlist-panel-header">
        <span><i class="fas fa-star"></i> WATCHLIST</span>
        <button class="clear-watchlist-btn" id="clearWatchlistBtn" title="Clear All">Clear
    </div>

    <div class="watchlist-container">
        <div class="search-container">
            <input id="watchlistSearch" placeholder="Enter stock symbol (e.g., RELIANCE
                <button id="watchlistSearchBtn">Add Stock</button>
            </div>

            <div class="watchlist-stocks" id="watchlistStocks">
                <div class="no-watchlist-stocks">No stocks in watchlist. Search and add stoc
            </div>
        </div>

        <!-- Quick Actions for Watchlist -->
        <div style="margin-top: 20px; padding-top: 15px; border-top: 1px solid rgba(255, 255, 255, 0.5); background-color: #f9f9f9; padding-bottom: 10px; border-bottom: 1px solid #ccc; position: relative; z-index: 1; width: 100%; height: 100%;">
            <h4 style="color: var(--accent-color); font-size: 14px; margin-bottom: 10px; text-align: center; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);>
                <i class="fas fa-bolt"></i> QUICK ACTIONS
        </div>
    </div>
</div>

```

```

</h4>
<div style="display: grid; grid-template-columns: 1fr 1fr; gap: 8px;">
  <button onclick="addMultipleStocks()" style="padding: 8px; background: #00aa00; color: white; border: none; border-radius: 5px; font-size: 14px; font-weight: bold; text-decoration: none; width: 100%; height: 100%;">
    <i class="fas fa-plus-circle"></i> Add Multiple
  </button>
  <button onclick="importWatchlist()" style="padding: 8px; background: #9d4ec1; color: white; border: none; border-radius: 5px; font-size: 14px; font-weight: bold; text-decoration: none; width: 100%; height: 100%;">
    <i class="fas fa-file-import"></i> Import
  </button>
  <button onclick="exportWatchlist()" style="padding: 8px; background: #00aa00; color: white; border: none; border-radius: 5px; font-size: 14px; font-weight: bold; text-decoration: none; width: 100%; height: 100%;">
    <i class="fas fa-file-export"></i> Export
  </button>
  <button onclick="refreshWatchlist()" style="padding: 8px; background: #ff9900; color: white; border: none; border-radius: 5px; font-size: 14px; font-weight: bold; text-decoration: none; width: 100%; height: 100%;">
    <i class="fas fa-sync-alt"></i> Refresh
  </button>
</div>
</div>
</div>

<!-- ===== MAIN CONTENT ===== -->
<div class="content" id="mainContent">
  <!-- Open Positions Page -->
  <div class="content-page" id="openPositionsPage">
    <div class="open-positions-container">
      <!-- ===== ENHANCED POSITIONS MAIN CONTENT ===== -->
      <div class="positions-main">
        <div class="positions-header">
          <h2><i class="fas fa-chart-line"></i> Open Positions</h2>
          <button class="add-position-btn" onclick="openAddPositionModal()">
            <i class="fas fa-plus"></i> Add Position
          </button>
        </div>
        <div style="overflow-x: auto;">
          <table class="positions-table">
            <thead>
              <tr>
                <th>#</th>
                <th>Symbol</th>
                <th>Order Type</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <td>1</td>
                <td>AAPL</td>
                <td>Buy</td>
              </tr>
              <tr>
                <td>2</td>
                <td>GOOGL</td>
                <td>Sell</td>
              </tr>
              <tr>
                <td>3</td>
                <td>MSFT</td>
                <td>Buy</td>
              </tr>
              <tr>
                <td>4</td>
                <td>AMZN</td>
                <td>Sell</td>
              </tr>
              <tr>
                <td>5</td>
                <td>NFLX</td>
                <td>Buy</td>
              </tr>
              <tr>
                <td>6</td>
                <td>TSLA</td>
                <td>Sell</td>
              </tr>
              <tr>
                <td>7</td>
                <td>NVDA</td>
                <td>Buy</td>
              </tr>
              <tr>
                <td>8</td>
                <td>INTC</td>
                <td>Sell</td>
              </tr>
              <tr>
                <td>9</td>
                <td>FB</td>
                <td>Buy</td>
              </tr>
              <tr>
                <td>10</td>
                <td>GOOG</td>
                <td>Sell</td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        <th>Order Condition</th>
        <th>Entry Price</th>
        <th>Current Price</th>
        <th>Buy Qty</th>
        <th>Sell Qty</th>
        <th>Buy Avg Price</th>
        <th>Sell Avg Price</th>
        <th>Day P&L</th>
        <th>Net P&L</th>
        <th>Net %</th>
        <th>Source</th>
        <th>Action</th>
    </tr>
</thead>
<tbody id="positionsTable">
    <!-- Positions will be loaded here -->
</tbody>
</table>
</div>
</div>
</div>
</div>

<!-- Other Pages (simplified) -->
<div class="content-page hidden" id="orderbookPage">
    <h2><i class="fas fa-book"></i> Order Book & Portfolio</h2>
    <p>Order book content will be displayed here.</p>
</div>

<div class="content-page hidden" id="screenerPage">
    <h2><i class="fas fa-search"></i> Stock Screener</h2>
    <p>This page shows advanced stock screening tools.</p>
</div>

<div class="content-page hidden" id="strategyCreationPage">
    <h2><i class="fas fa-chess-board"></i> Strategy Creation</h2>
    <p>This page allows you to create trading strategies.</p>
</div>

```

```

<div class="content-page hidden" id="reportPage">
  <h2><i class="fas fa-chart-bar"></i> Reports</h2>
  <p>This page shows trading reports and analytics.</p>
</div>

<!-- ===== ENHANCED BROKER PAGE ===== -->
<div class="content-page hidden" id="brokerPage">
  <h2><i class="fas fa-university"></i> Broker Management</h2>

  <!-- Broker Form -->
  <div style="background: #1a1a1a; padding: 20px; border-radius: 10px; margin->
    <h3 style="color: var(--accent-color); margin-bottom: 20px;">
      <i class="fas fa-plus-circle"></i> Add/Edit Broker
    </h3>

    <div style="display: grid; grid-template-columns: 1fr 1fr; gap: 15px;">
      <div>
        <label style="display: block; margin-bottom: 8px; color: #aaa;">Broker N
        <select id="brokerSelect" style="width:100%; padding:10px; background->
          <option value="">Select Broker</option>
          <option value="ZERODHA">Zerodha</option>
          <option value="ANGEL">Angel Broking</option>
          <option value="UPSTOX">Upstox</option>
          <option value="ICICI">ICICI Direct</option>
          <option value="HDFC">HDFC Securities</option>
          <option value="ALICE">Alice Blue</option>
          <option value="FIVE">5Paisa</option>
          <option value="CUSTOM">Custom Broker</option>
        </select>
      </div>

      <div>
        <label style="display: block; margin-bottom: 8px; color: #aaa;">Client ID<
        <input type="text" id="clientId" placeholder="Enter Client ID"
              style="width:100%; padding:10px; background:#333; color:#fff; borc
      </div>

      <div>
        <label style="display: block; margin-bottom: 8px; color: #aaa;">API Key<
    
```

```

<input type="password" id="apiKey" placeholder="Enter API Key"
       style="width:100%; padding:10px; background:#333; color:#fff; border:none; border-radius:5px; margin-bottom:10px;" />
</div>

<div>
  <label style="display: block; margin-bottom: 8px; color: #aaa;">API Secret</label>
  <input type="password" id="apiSecret" placeholder="Enter API Secret"
         style="width:100%; padding:10px; background:#333; color:#fff; border:none; border-radius:5px; margin-bottom:10px;" />
</div>

<div>
  <label style="display: block; margin-bottom: 8px; color: #aaa;">Access Token</label>
  <input type="text" id="accessToken" placeholder="Enter Access Token"
         style="width:100%; padding:10px; background:#333; color:#fff; border:none; border-radius:5px; margin-bottom:10px;" />
</div>

<div>
  <label style="display: block; margin-bottom: 8px; color: #aaa;">PIN (Optional)</label>
  <input type="password" id="brokerPin" placeholder="Enter PIN"
         style="width:100%; padding:10px; background:#333; color:#fff; border:none; border-radius:5px; margin-bottom:10px;" />
</div>
</div>

<!-- Action Buttons -->
<div style="display: flex; gap: 10px; margin-top: 20px;">
  <button onclick="addEditBroker()" style="flex:1; padding:12px; background:#333; color:#fff; border:none; border-radius:5px; font-weight:bold;">
    <i class="fas fa-save"></i> Add/Update Broker
  </button>
  <button onclick="deleteBroker()" style="flex:1; padding:12px; background:#333; color:#fff; border:none; border-radius:5px; font-weight:bold;">
    <i class="fas fa-trash"></i> Delete Broker
  </button>
  <button onclick="checkBrokerConnection()" style="flex:1; padding:12px; background:#333; color:#fff; border:none; border-radius:5px; font-weight:bold;">
    <i class="fas fa-plug"></i> Check Connection
  </button>
</div>
</div>

<!-- Saved Brokers List -->
<div style="background: #1a1a1a; padding: 20px; border-radius: 10px; margin-top: 20px;">

```

```
<h3 style="color: var(--accent-color); margin-bottom: 15px;">
  <i class="fas fa-list"></i> Saved Brokers
</h3>
<div id="brokerList" style="max-height: 300px; overflow-y: auto;">
  <!-- Broker list will be loaded here -->
</div>
</div>
</div>

<div class="content-page hidden" id="profilePage">
  <h2><i class="fas fa-user"></i> Profile</h2>
  <p>User profile content will be displayed here.</p>
</div>

<div class="content-page hidden" id="marketplacePage">
  <h2><i class="fas fa-store"></i> Market Place</h2>
  <p>Marketplace content will be displayed here.</p>
</div>

<div class="content-page hidden" id="paperTradingPage">
  <h2><i class="fas fa-file-invoice-dollar"></i> Paper Trading</h2>
  <p>Paper trading content will be displayed here.</p>
</div>

<div class="content-page hidden" id="myPlanPage">
  <h2><i class="fas fa-calendar-alt"></i> My Plan</h2>
  <p>Plan details will be displayed here.</p>
</div>

<div class="content-page hidden" id="backtestPage">
  <h2><i class="fas fa-history"></i> Backtest</h2>
  <p>Backtesting tools will be displayed here.</p>
</div>

<div class="content-page hidden" id="settingsPage">
  <h2><i class="fas fa-cog"></i> Settings</h2>
  <p>Settings content will be displayed here.</p>
</div>
```

```

<div class="content-page hidden" id="aiPage">
  <h2><i class="fas fa-robot"></i> AL</h2>
  <p>AI trading tools will be displayed here.</p>
</div>
</div>

<!-- ===== ENHANCED ADD POSITION MODAL ===== -->
<div id="addPositionModal" class="modal" onclick="outsideCloseAddModal(event)">
  <div class="modal-box">
    <div class="modal-header">
      <h3 id="modalSymbolTitle">Add Position</h3>
      <button onclick="closeAddModal()"> </button>
    </div>

    <div>
      <div style="margin-bottom:15px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
          <input id="positionSymbol" placeholder="Enter stock symbol (e.g., RELIAN
        </div>

      <div style="display:grid;grid-template-columns:1fr 1fr;gap:15px;margin-bottom:15px">
        <div>
          <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
            <select id="orderType">
              <option value="BUY">BUY</option>
              <option value="SELL">SELL</option>
            </select>
          </div>
        <div>
          <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
            <select id="orderCondition">
              <option value="Intraday">Intraday</option>
              <option value="Positional">Positional</option>
              <option value="BO">BO (Bracket Order)</option>
              <option value="St">St (Stop Loss)</option>
              <option value="MSL">MSL (Multiple Stop Loss)</option>
            </select>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

<div style="display:grid;grid-template-columns:1fr 1fr;gap:15px;margin-bottom:15px;">
  <div>
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa;">Buy Qty
    <input id="buyQty" type="number" placeholder="Buy Qty" value="0">
  </div>
  <div>
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa;">Sell Qty
    <input id="sellQty" type="number" placeholder="Sell Qty" value="0">
  </div>
</div>

<div style="display:grid;grid-template-columns:1fr 1fr;gap:15px;margin-bottom:15px;">
  <div>
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa;">Entry Price
    <input id="entryPrice" type="number" placeholder="Entry Price">
  </div>
  <div>
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa;">Current Price
    <input id="currentPrice" type="number" placeholder="Current Price">
  </div>
</div>

<div style="display:grid;grid-template-columns:1fr 1fr;gap:15px;margin-bottom:15px;">
  <div>
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa;">Buy Avg Price
    <input id="buyAvgPrice" type="number" placeholder="Buy Avg Price">
  </div>
  <div>
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa;">Sell Avg Price
    <input id="sellAvgPrice" type="number" placeholder="Sell Avg Price">
  </div>
</div>

<!-- ====== ENABLE/DISABLE OPTIONS ====== -->
<div style="margin: 20px 0; padding: 15px; background: #222; border-radius: 10px;">
  <h4 style="color: var(--accent-color); margin-bottom: 15px;">
    <i class="fas fa-toggle-on"></i> Enable/Disable Options
  </h4>

```

```

<div class="enable-option" onclick="toggleEnableOption('positionSettings')>
  <div>
    <i class="fas fa-cog"></i> Position Settings
  </div>
  <div class="enable-status enabled" id="positionSettingsStatus">
    <i class="fas fa-check"></i>
  </div>
</div>

<div class="enable-option" onclick="toggleEnableOption('partitionSettings')>
  <div>
    <i class="fas fa-layer-group"></i> Partition Settings
  </div>
  <div class="enable-status enabled" id="partitionSettingsStatus">
    <i class="fas fa-check"></i>
  </div>
</div>

<div class="enable-option" onclick="toggleEnableOption('trailingSL')">
  <div>
    <i class="fas fa-chart-line"></i> Trailing Stop Loss
  </div>
  <div class="enable-status enabled" id="trailingSLStatus">
    <i class="fas fa-check"></i>
  </div>
</div>

<div class="enable-option" onclick="toggleEnableOption('profitLock')">
  <div>
    <i class="fas fa-lock"></i> Profit Lock
  </div>
  <div class="enable-status disabled" id="profitLockStatus">
    <i class="fas fa-times"></i>
  </div>
</div>
</div>

<!-- ===== POSITION SECTION (UPDATED WITH SCROLLABLE & ON/OFF TOGGLE) -->

```

```

<div class="position-section" id="positionSettingsSection">
  <div class="position-section-header" onclick="togglePositionSection()">
    <h4><i class="fas fa-cog"></i> Position Settings</h4>
    <button class="position-section-toggle" id="positionSectionToggle">
      <i class="fas fa-chevron-up"></i> Collapse
    </button>
  </div>

  <div class="position-section-body" id="positionSectionBody">
    <!-- Order Type with Toggle -->
    <div class="position-row">
      <div class="position-toggle">
        <label class="toggle-switch">
          <input type="checkbox" id="toggleOrderType" checked>
          <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Order Type</span>
      </div>
      <div class="position-controls">
        <select id="positionOrderType" class="section-input">
          <option value="limit">Limit</option>
          <option value="market">Market</option>
          <option value="sl">Stop Loss (SL)</option>
          <option value="market_sl">Market Stop Loss</option>
        </select>
      </div>
    </div>
  </div>

  <!-- Part with Toggle -->
  <div class="position-row">
    <div class="position-toggle">
      <label class="toggle-switch">
        <input type="checkbox" id="togglePart" checked>
        <span class="toggle-slider"></span>
      </label>
      <span class="toggle-label">Part</span>
    </div>
    <div class="position-controls">
      <select id="positionPart" class="section-input">

```

```

        <option value="buy">Buy</option>
        <option value="sell">Sell</option>
    </select>
</div>
</div>

<!-- Quantity with Toggle -->
<div class="position-row">
    <div class="position-toggle">
        <label class="toggle-switch">
            <input type="checkbox" id="toggleQty" checked>
            <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Quantity</span>
    </div>
    <div class="position-controls">
        <div style="display: flex; gap: 10px; width: 100%;">
            <input id="positionQty" type="number" placeholder="Buy Qty" value="100" min="1" max="1000" style="width: 100px;">
            <input id="positionSellQty" type="number" placeholder="Sell Qty" value="100" min="1" max="1000" style="width: 100px;">
        </div>
    </div>
</div>

<!-- Trailing Stop Loss with Toggle -->
<div class="position-row" id="trailingSLRow">
    <div class="position-toggle">
        <label class="toggle-switch">
            <input type="checkbox" id="toggleTrailingSL" checked>
            <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Trailing SL</span>
    </div>
    <div class="position-controls">
        <div style="display: flex; gap: 10px; width: 100%;">
            <select id="trailingSLType" style="flex: 1;">
                <option value="point">Point</option>
                <option value="price">Price</option>
                <option value="spot">Spot</option>
                <option value="fut">Futures</option>
            </select>
        </div>
    </div>
</div>

```

```

        <option value="mtm">MTM</option>
        <option value="indicator">Indicator Based</option>
    </select>
    <input id="trailingSLValue" type="number" placeholder="Value" s
    </div>
</div>
</div>

<!-- Indicator Options (Hidden by default) -->
<div id="indicatorOptions" class="indicator-options">
    <div>
        <label class="indicator-label">Period</label>
        <input id="indicatorPeriod" type="number" placeholder="Period" cl
    </div>
    <div>
        <label class="indicator-label">Condition</label>
        <select id="indicatorCondition" class="indicator-input">
            <option value="close">Close</option>
            <option value="cross">Cross</option>
        </select>
    </div>
</div>

<!-- Target with Toggle -->
<div class="position-row">
    <div class="position-toggle">
        <label class="toggle-switch">
            <input type="checkbox" id="toggleTarget" checked>
            <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Target</span>
    </div>
    <div class="position-controls">
        <div style="display: flex; gap: 10px; width: 100%;">
            <select id="targetType" style="flex: 1;">
                <option value="point">Point</option>
                <option value="price">Price</option>
                <option value="spot">Spot</option>
                <option value="fut">Futures</option>
            </select>
        </div>
    </div>
</div>

```

```

        <option value="mtm">MTM</option>
    </select>
    <input id="targetValue" type="number" placeholder="Target Valu
    </div>
</div>

<!-- Exit with Toggle -->
<div class="position-row">
    <div class="position-toggle">
        <label class="toggle-switch">
            <input type="checkbox" id="toggleExit" checked>
            <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Exit</span>
    </div>
    <div class="position-controls">
        <div style="display: flex; gap: 10px; width: 100%;">
            <select id="exitType" style="flex: 1;">
                <option value="point">Point</option>
                <option value="price">Price</option>
                <option value="spot">Spot</option>
                <option value="fut">Futures</option>
                <option value="mtm">MTM</option>
            </select>
            <input id="exitValue" type="number" placeholder="Exit Value" st
        </div>
    </div>
</div>

<!-- Profit Lock with Toggle -->
<div class="position-row" id="profitLockRow">
    <div class="position-toggle">
        <label class="toggle-switch">
            <input type="checkbox" id="toggleProfitLock" checked>
            <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Profit Lock</span>
    </div>

```

```

<div class="position-controls">
    <input id="profitLock" type="number" placeholder="Profit Lock Price" />
</div>

<!-- Profit Gap with Toggle -->
<div class="position-row">
    <div class="position-toggle">
        <label class="toggle-switch">
            <input type="checkbox" id="toggleProfitGap" checked>
            <span class="toggle-slider"></span>
        </label>
        <span class="toggle-label">Profit Gap</span>
    </div>
    <div class="position-controls">
        <input id="profitGap" type="number" placeholder="Profit Gap Price" />
    </div>
</div>

<!-- Quick Toggle All Button -->
<div style="display: flex; gap: 10px; margin-top: 15px;">
    <button onclick="toggleAllPositionSettings(true)" style="flex:1; padding: 10px; border-radius: 10px; font-size: 14px; background-color: #f0f0f0; border: none; color: black; text-decoration: none; font-weight: bold; margin-right: 10px;">
        <i class="fas fa-toggle-on"></i> All ON
    </button>
    <button onclick="toggleAllPositionSettings(false)" style="flex:1; padding: 10px; border-radius: 10px; font-size: 14px; background-color: #f0f0f0; border: none; color: black; text-decoration: none; font-weight: bold; margin-right: 10px;">
        <i class="fas fa-toggle-off"></i> All OFF
    </button>
    <button onclick="resetPositionSettings()" style="flex:1; padding: 10px; border-radius: 10px; font-size: 14px; background-color: #f0f0f0; border: none; color: black; text-decoration: none; font-weight: bold;">
        <i class="fas fa-redo"></i> Reset
    </button>
</div>
</div>

<!-- ===== PARTITION & ADD QUANTITY SECTION ===== -->
<div class="partition-section" id="partitionSettingsSection">
    <div class="partition-section-header" onclick="togglePartitionSection()">
        <h4><i class="fas fa-layer-group"></i> Partition & Add Quantity</h4>
        <button class="partition-section-toggle" id="partitionSectionToggle">

```

```

        <i class="fas fa-chevron-up"></i> Collapse
    </button>
</div>

<div class="partition-section-body" id="partitionSectionBody">
    <div class="partition-grid">
        <!-- Left Column: Partition Square Off -->
        <div class="partition-column">
            <div class="partition-item">
                <h5><i class="fas fa-chart-pie"></i> Partition Square Off</h5>
                <div class="partition-row">
                    <div>
                        <label class="indicator-label">Quantity (Number)</label>
                        <input id="partitionSquareOffQty" type="number" placeholder="0" class="input-field">
                    </div>
                    <div>
                        <label class="indicator-label">Price</label>
                        <input id="partitionSquareOffPrice" type="number" placeholder="0.00" class="input-field">
                    </div>
                </div>
            </div>
        <div class="partition-item">
            <h5><i class="fas fa-plus-circle"></i> Add Quantity</h5>
            <div class="partition-row">
                <div>
                    <label class="indicator-label">Quantity (Number)</label>
                    <input id="addQty" type="number" placeholder="0" class="input-field">
                </div>
                <div>
                    <label class="indicator-label">Price</label>
                    <input id="addQtyPrice" type="number" placeholder="0.00" class="input-field">
                </div>
            </div>
        </div>
    </div>
<!-- Right Column: Additional Settings -->
<div class="partition-column">

```

```
<div class="partition-item">
    <h5><i class="fas fa-sliders-h"></i> Additional Settings</h5>
    <div style="display: flex; flex-direction: column; gap: 10px;">
        <div>
            <label class="indicator-label">Max Partition Count</label>
            <input id="maxPartitionCount" type="number" placeholder="1000" style="width: 100%;"/>
        </div>
        <div>
            <label class="indicator-label">Auto Square Off %</label>
            <input id="autoSquareOffPercent" type="number" placeholder="0" style="width: 100%;"/>
        </div>
        <div>
            <label class="indicator-label">Re-entry Attempts</label>
            <input id="reentryAttempts" type="number" placeholder="3" style="width: 100%;"/>
        </div>
    </div>
</div>

<div class="partition-item">
    <h5><i class="fas fa-calculator"></i> Quick Calculate</h5>
    <div style="text-align: center;">
        <button onclick="calculatePositionSizing()" style="width:100%; height: 40px; border-radius: 10px; border: none; background-color: #ccc; padding: 10px; margin-bottom: 10px; font-size: 14px; font-weight: bold; color: #aaa; transition: background-color 0.3s ease; font-family: sans-serif;">
            Calculate Position
        </button>
        <div id="calculationResult" style="font-size:14px; color:#aaa; padding: 10px; border: 1px solid #ccc; border-radius: 10px; width: fit-content; margin: auto; font-family: sans-serif;">
            Enter values to calculate
        </div>
    </div>
</div>
</div>
</div>
```

```

        <option value="SIMULATOR">SIMULATOR</option>
    </select>
</div>

        <button class="full-btn success" onclick="addPosition()">Add Position</button>
    </div>
</div>
</div>

<!-- ===== ACTION MODAL ===== -->
<div id="actionModal" class="modal" onclick="outsideCloseActionModal(event)">
    <div class="modal-box">
        <div class="modal-header">
            <h3 id="actionModalSymbolTitle">Position Actions</h3>
            <button onclick="closeActionModal()"> </button>
        </div>

        <div class="tabs">
            <button class="active" onclick="switchActionTab(event,'modify')">Modify</button>
            <button onclick="switchActionTab(event,'exit')">Exit</button>
            <button onclick="switchActionTab(event,'existingCode')">Existing Code</button>
            <button onclick="switchActionTab(event,'settings')">Settings</button>
            <button onclick="switchActionTab(event,'partition')">Partition</button>
        </div>

        <div id="modify" class="tab-content active">
            <div style="margin-bottom:12px">
                <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
                    <input id="modifyStopLoss" type="number" placeholder="Stop Loss Price">
                </div>
            <div style="margin-bottom:12px">
                <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
                    <input id="modifyTarget" type="number" placeholder="Target Price">
                </div>
            <div style="margin-bottom:12px">
                <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
                    <input id="modifyTrailingSL" type="number" placeholder="Trailing SL Poin">
                </div>
            <button class="full-btn success" onclick="modifyPosition()>Update Position</button>
        </div>
    </div>
</div>

```

```

</div>

<div id="exit" class="tab-content">
  <div style="margin-bottom:12px">
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
      <input id="exitQty" type="number" placeholder="Quantity to Exit">
    </div>
  <div style="margin-bottom:12px">
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
      <input id="exitPrice" type="number" placeholder="Exit Price">
    </div>
  <div style="margin-bottom:12px">
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
      <select id="exitReason">
        <option value="target_hit">Target Hit</option>
        <option value="stop_loss">Stop Loss Hit</option>
        <option value="manual">Manual Exit</option>
        <option value="trailing_sl">Trailing SL Hit</option>
      </select>
    </div>
    <button class="full-btn danger" onclick="exitPosition()">Exit Position</button>
  </div>

<div id="existingCode" class="tab-content">
  <div style="margin-bottom:12px">
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
      <textarea id="strategyCode" placeholder="Enter your trading strategy cod
    </div>
  <div style="margin-bottom:12px">
    <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
      <div class="code-display" id="currentCodeDisplay">
        <div class="code-line">// Sample Trading Strategy</div>
        <div class="code-line">if (rsi < 30 && macd_signal > 0) {</div>
        <div class="code-line">  entryPrice = currentPrice;</div>
        <div class="code-line">  stopLoss = entryPrice * 0.98;</div>
        <div class="code-line">  target = entryPrice * 1.02;</div>
        <div class="code-line">}</div>
      </div>
    </div>
  </div>

```

```
<div style="display:flex;gap:10px">
    <button class="full-btn info" onclick="saveStrategyCode()">Save Code</button>
    <button class="full-btn" onclick="testStrategyCode()" style="background:#fff">Test</button>
</div>

<div id="settings" class="tab-content">
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Alert Price</label>
        <input id="alertPrice" type="number" placeholder="Alert Price">
    </div>
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Auto Square Off Time</label>
        <input id="autoSquareOff" type="time" value="15:15">
    </div>
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Maximum Loss Amount</label>
        <input id="maxLoss" type="number" placeholder="Maximum Loss Amount">
    </div>
    <button class="full-btn info" onclick="saveSettings()">Save Settings</button>
</div>

<div id="partition" class="tab-content">
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Action Partition Square Off Qty</label>
        <input id="actionPartitionSquareOffQty" type="number" placeholder="Quantity">
    </div>
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Action Partition Square Off Price</label>
        <input id="actionPartitionSquareOffPrice" type="number" placeholder="Price">
    </div>
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Action Add Qty</label>
        <input id="actionAddQty" type="number" placeholder="Quantity">
    </div>
    <div style="margin-bottom:12px">
        <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">Action Add Qty Price</label>
        <input id="actionAddQtyPrice" type="number" placeholder="Price">
    </div>
</div>
```

```

<div style="margin-bottom:12px">
  <label style="display:block;margin-bottom:8px;font-size:14px;color:#aaa">
    <select id="partitionAction">
      <option value="square_off">Square Off Partition</option>
      <option value="add_qty">Add Quantity</option>
      <option value="both">Both</option>
    </select>
  </div>
  <button class="full-btn success" onclick="executePartitionAction()">Execute
  </div>
</div>
</div>

<!-- ===== SCREENER POPUP MODAL ===== -->
<div id="screenerPopupModal" class="modal" onclick="closeScreenerPopup(event)">
  <div class="modal-box" style="width: 600px;">
    <div class="modal-header">
      <h3><i class="fas fa-search"></i> Stock Screener</h3>
      <button onclick="closeScreenerPopup()"> </button>
    </div>

    <div style="display: flex; gap: 10px; margin: 15px 0;">
      <button onclick="openChartink()" style="flex:1; padding:12px; background:#fff; border-radius:10px; color:#000; font-weight:bold; text-decoration:none; font-size:14px">
        <i class="fas fa-external-link-alt"></i> Open Chartink
      </button>
    </div>

    <div style="background: #333; padding: 15px; border-radius: 5px; margin: 15px 0;">
      <h4 style="color: var(--accent-color); margin-bottom: 10px;">
        <i class="fas fa-sliders-h"></i> Indicator Based Screener
      </h4>
      <div style="display: grid; grid-template-columns: 1fr 1fr; gap: 10px;">
        <select id="indicatorSelect" style="padding:8px; background:#444; color:#fff; border-radius:5px; width:100%; height:40px">
          <option>Select Indicator</option>
          <option>RSI (Relative Strength Index)</option>
          <option>MACD (Moving Average Convergence Divergence)</option>
          <option>Bollinger Bands</option>
          <option>Moving Averages</option>
          <option>Stochastic Oscillator</option>
        </select>
      </div>
    </div>
  </div>
</div>

```

```

        </select>
        <input id="indicatorValue" type="number" placeholder="Value" style="pac
    </div>
    <button onclick="deployIndicatorScreener()" style="margin-top:10px; paddin
        <i class="fas fa-play"></i> Deploy Screener
    </button>
</div>

<div style="background: #333; padding: 15px; border-radius: 5px; margin: 15px 0; backgroun
    <h4 style="color: var(--accent-color); margin-bottom: 10px;">
        <i class="fas fa-save"></i> Saved Screeners
    </h4>
    <div id="savedScreenersList" style="margin-bottom:10px;">
        <div style="padding:8px; background:#444; margin:5px 0; border-radius:3px; backgroun
            <i class="fas fa-chart-line"></i> RSI Oversold Strategy
        </div>
        <div style="padding:8px; background:#444; margin:5px 0; border-radius:3px; backgroun
            <i class="fas fa-chart-line"></i> MACD Crossover Strategy
        </div>
    </div>
    <button onclick="deploySavedScreener()" style="margin-top:5px; padding:8px 0; backgroun
        <i class="fas fa-play"></i> Deploy Saved Screener
    </button>
</div>

<div style="display: flex; gap: 10px; margin-top: 15px;">
    <button onclick="createNewScreener()" style="flex:1; padding:10px; background-color: #444; color: white; border: none; border-radius: 5px; font-size: 1em; font-weight: bold;">
        <i class="fas fa-plus"></i> Create New Screener
    </button>
</div>
</div>
</div>

<!-- ===== STRATEGY CREATION POPUP ===== -->
<div id="strategyCreationPopup" class="modal" onclick="closeStrategyPopup(event)" style="display: none; position: fixed; top: 0; left: 0; width: 100%; height: 100%; background-color: black; opacity: 0.5; z-index: -1;">
    <div class="modal-box" style="width: 600px; height: 400px; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); background-color: white; border: 1px solid #ccc; border-radius: 10px; padding: 20px; overflow: auto; z-index: 1;">
        <div class="modal-header">
            <h3><i class="fas fa-chess-board"></i> Strategy Creation</h3>
            <button onclick="closeStrategyPopup()"> </button>
        </div>
        <div class="modal-body" style="height: 250px; padding: 10px; border-top: none; border-left: none; border-right: none; border-bottom: 1px solid #ccc; border-radius: 10px 10px 0 0; position: relative;">
            <div style="border: 1px solid #ccc; border-radius: 10px; width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: #f9f9f9; z-index: 1;">
                <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); font-size: 1.5em; font-weight: bold; color: #ccc; z-index: 2;">
                    <i class="fas fa-chess-board"></i> Strategy Creation
                </div>
            </div>
        </div>
    </div>
</div>

```

```

</div>

<div style="text-align: center; padding: 20px 0;">
  <h4 style="color: var(--accent-color); margin-bottom: 20px;">Select Strategy |</h4>
  <div style="display: grid; grid-template-columns: 1fr 1fr; gap: 15px; margin-bottom: 20px;">
    <button onclick="openOpstra()" style="padding: 15px; background: linear-gradient(to top right, #f0f0f0, #e0e0e0); border-radius: 10px; font-weight: bold; color: inherit; text-decoration: none; font-size: 14px; font-family: inherit; border: none; width: 100%; height: 100%;">
      <i class="fas fa-chart-pie"></i><br>
      <strong>Opstra</strong><br>
      <small>Options Strategy</small>
    </button>

    <button onclick="openSensibull()" style="padding: 15px; background: linear-gradient(to top right, #f0f0f0, #e0e0e0); border-radius: 10px; font-weight: bold; color: inherit; text-decoration: none; font-size: 14px; font-family: inherit; border: none; width: 100%; height: 100%;">
      <i class="fas fa-bullhorn"></i><br>
      <strong>Sensibull</strong><br>
      <small>Options Trading</small>
    </button>
  </div>

  <div style="display: grid; grid-template-columns: 1fr 1fr; gap: 15px;">
    <button onclick="openMarketPlace()" style="padding: 15px; background: linear-gradient(to top right, #f0f0f0, #e0e0e0); border-radius: 10px; font-weight: bold; color: inherit; text-decoration: none; font-size: 14px; font-family: inherit; border: none; width: 100%; height: 100%;">
      <i class="fas fa-store"></i><br>
      <strong>Market Place</strong><br>
      <small>Strategy Marketplace</small>
    </button>

    <button onclick="openCreateStrategy()" style="padding: 15px; background: linear-gradient(to top right, #f0f0f0, #e0e0e0); border-radius: 10px; font-weight: bold; color: inherit; text-decoration: none; font-size: 14px; font-family: inherit; border: none; width: 100%; height: 100%;">
      <i class="fas fa-plus-circle"></i><br>
      <strong>Create Strategy</strong><br>
      <small>Custom Strategy Builder</small>
    </button>
  </div>
</div>

<div style="background: #333; padding: 15px; border-radius: 5px; margin-top: 20px;">
  <h5 style="color: #aaa; margin-bottom: 10px;">Quick Actions:</h5>
  <div style="display: flex; gap: 10px;">
    <button onclick="backtestStrategy()" style="flex: 1; padding: 8px; background: #333; color: inherit; text-decoration: none; border: none; border-radius: 5px; font-size: 14px; font-family: inherit;">
      <i class="fas fa-history"></i> Backtest
    </button>
  </div>
</div>

```

```

        </button>
        <button onclick="optimizeStrategy()" style="flex:1; padding:8px; background-color:#e0f2e0; border-radius:10px; margin-right:10px;">
          <i class="fas fa-cogs"></i> Optimize
        </button>
        <button onclick="shareStrategy()" style="flex:1; padding:8px; background-color:#e0f2e0; border-radius:10px;">
          <i class="fas fa-share"></i> Share
        </button>
      </div>
    </div>
  </div>

<script>
// ===== GLOBAL VARIABLES =====
let positions = [];
let watchlistStocks = [];
let currentPositionIndex = -1;
let navPopupTimeout = null;
let positionSectionCollapsed = false;
let partitionSectionCollapsed = false;
let enableOptions = {
  positionSettings: true,
  partitionSettings: true,
  trailingSL: true,
  profitLock: false
};

// Stock database for search
const stockDatabase = [
  { symbol: "RELIANCE", name: "Reliance Industries Ltd", sector: "Oil & Gas" },
  { symbol: "TCS", name: "Tata Consultancy Services", sector: "IT" },
  { symbol: "INFY", name: "Infosys Limited", sector: "IT" },
  { symbol: "HDFCBANK", name: "HDFC Bank Limited", sector: "Banking" },
  { symbol: "ICICIBANK", name: "ICICI Bank Limited", sector: "Banking" },
  { symbol: "HINDUNILVR", name: "Hindustan Unilever Ltd", sector: "FMCG" },
  { symbol: "SBIN", name: "State Bank of India", sector: "Banking" },
  { symbol: "BHARTIARTL", name: "Bharti Airtel Ltd", sector: "Telecom" },
  { symbol: "ITC", name: "ITC Limited", sector: "FMCG" },
  { symbol: "LT", name: "Larsen & Toubro Ltd", sector: "Construction" },

```

```

{ symbol: "KOTAKBANK", name: "Kotak Mahindra Bank", sector: "Banking" },
{ symbol: "AXISBANK", name: "Axis Bank Limited", sector: "Banking" },
{ symbol: "WIPRO", name: "Wipro Limited", sector: "IT" },
{ symbol: "SUNPHARMA", name: "Sun Pharmaceutical", sector: "Pharma" },
{ symbol: "TITAN", name: "Titan Company Ltd", sector: "Consumer Goods" },
{ symbol: "TATAMOTORS", name: "Tata Motors Ltd", sector: "Automobile" },
{ symbol: "BAJFINANCE", name: "Bajaj Finance Ltd", sector: "Finance" },
{ symbol: "HCLTECH", name: "HCL Technologies", sector: "IT" },
{ symbol: "ASIANPAINT", name: "Asian Paints Ltd", sector: "Paints" },
{ symbol: "MARUTI", name: "Maruti Suzuki India", sector: "Automobile" }
];

// ===== MARKET TICKER DATA WITH INDICES =====
let tickerData = [
  { symbol: "NIFTY", price: 22450.75, change: 125.50, name: "NIFTY 50" },
  { symbol: "BANKNIFTY", price: 47520.30, change: 320.75, name: "NIFTY BANK" },
  { symbol: "SENSEX", price: 73845.25, change: 280.40, name: "S&P BSE SENSEX" },
  { symbol: "INDIAVIX", price: 13.25, change: -0.45, name: "INDIA VIX" },
  { symbol: "RELIANCE", price: 2985.60, change: 12.50, name: "Reliance Industrie" },
  { symbol: "TCS", price: 3845.25, change: -24.50, name: "Tata Consultancy Servic" },
  { symbol: "INFY", price: 1560.80, change: 8.20, name: "Infosys Limited" },
  { symbol: "HDFCBANK", price: 1645.90, change: -5.30, name: "HDFC Bank" }
];

// ===== LIVE CLOCK =====
function updateClock() {
  const now = new Date();
  const timeString = now.toLocaleTimeString('en-IN', {
    hour12: false,
    hour: '2-digit',
    minute: '2-digit',
    second: '2-digit'
  });
  document.getElementById('digitalClock').textContent = timeString;
}

// ===== NAVIGATION POPUP FUNCTIONS =====
function showNavPopup() {
  clearTimeout(navPopupTimeout);
}

```

```

        document.getElementById('navPopup').classList.add('show');
    }

    function hideNavPopup() {
        navPopupTimeout = setTimeout(() => {
            document.getElementById('navPopup').classList.remove('show');
        }, 300);
    }

    // Keep popup open when hovering over it
    document.getElementById('navPopup').addEventListener('mouseenter', () => {
        clearTimeout(navPopupTimeout);
    });

    // ===== MARKET TICKER FUNCTIONS =====
    function populateMarketTicker() {
        const ticker = document.getElementById('marketTicker');
        ticker.innerHTML = "";

        // Duplicate data for seamless scrolling
        const duplicateData = [...tickerData, ...tickerData];

        duplicateData.forEach((stock, index) => {
            const changeClass = stock.change >= 0 ? 'positive' : 'negative';
            const changeSign = stock.change >= 0 ? '+' : '';

            const tickerItem = document.createElement('div');
            tickerItem.className = 'ticker-item-horizontal';
            tickerItem.dataset.symbol = stock.symbol;
            tickerItem.dataset.index = index % tickerData.length;
            tickerItem.innerHTML = `
                <span class="ticker-symbol">${stock.symbol}</span>
                <span class="ticker-price">${stock.price.toLocaleString('en-IN', {minimumFractionDigits: 2})}</span>
                <span class="ticker-change ${changeClass}">${changeSign}${stock.change}</span>
            `;
            ticker.appendChild(tickerItem);
        });
    }
}

```

```
// ===== WATCHLIST FUNCTIONS =====
function loadWatchlist() {
    const savedWatchlist = localStorage.getItem('mukeshAlgoWatchlist');
    if (savedWatchlist) {
        watchlistStocks = JSON.parse(savedWatchlist);
    } else {
        // Default watchlist
        watchlistStocks = ["RELIANCE", "TCS", "INFY", "HDFCBANK"];
        saveWatchlist();
    }

    renderWatchlist();
}

function saveWatchlist() {
    localStorage.setItem('mukeshAlgoWatchlist', JSON.stringify(watchlistStocks));
}

function addToWatchlist() {
    const input = document.getElementById("watchlistSearch");
    const symbol = input.value.trim().toUpperCase();

    if (!symbol) {
        alert("Please enter a stock symbol");
        return;
    }

    // Check if stock exists in database
    const stockInfo = stockDatabase.find(stock => stock.symbol === symbol);
    if (!stockInfo) {
        alert("Stock not found. Please enter a valid stock symbol.");
        return;
    }

    // Check if already in watchlist
    if (watchlistStocks.includes(symbol)) {
        alert("Stock already in watchlist!");
        return;
    }
}
```

```

watchlistStocks.push(symbol);
saveWatchlist();
renderWatchlist();
input.value = "";
input.focus();
}

function removeFromWatchlist(symbol) {
  const index = watchlistStocks.indexOf(symbol);
  if (index !== -1) {
    watchlistStocks.splice(index, 1);
    saveWatchlist();
    renderWatchlist();
  }
}

function clearWatchlist() {
  if (watchlistStocks.length > 0 && confirm("Clear all stocks from watchlist?")) {
    watchlistStocks = [];
    saveWatchlist();
    renderWatchlist();
  }
}

function renderWatchlist() {
  const watchlistDiv = document.getElementById('watchlistStocks');

  if (watchlistStocks.length === 0) {
    watchlistDiv.innerHTML = '<div class="no-watchlist-stocks">No stocks in watchlist</div>';
    return;
  }

  let html = '';
  watchlistStocks.forEach((symbol, index) => {
    const stockInfo = stockDatabase.find(stock => stock.symbol === symbol) || {};
    html += `
      <div class="watchlist-item" data-symbol="${symbol}">
        <div style="display:flex; justify-content:space-between; align-items:center;`;
```

```

<div class="watchlist-symbol" onclick="openAddPositionModalForSymbol(${symbol})>
    <div class="watchlist-name">${stockInfo.name}</div>
</div>
<button class="delete-watchlist-btn" onclick="removeFromWatchlist('${symbol}')">
    </button>
</div>
<div class="watchlist-buttons">
    <button class="watchlist-btn watchlist-buy-btn" onclick="openAddPositionModal('buy')">Buy</button>
    <button class="watchlist-btn watchlist-sell-btn" onclick="openAddPositionModal('sell')">Sell</button>
    <button class="watchlist-btn watchlist-chart-btn" onclick="openChart('${symbol}')">Chart</button>
    <button class="watchlist-btn watchlist-depth-btn" onclick="openDepth('${symbol}')">Depth</button>
    <div class="watchlist-btn watchlist-more-btn">MORE</div>
        <div class="watchlist-more-menu">
            <div onclick="openInfo('${symbol}')">INFO</div>
            <div onclick="openGTT('${symbol}')">GTT</div>
            <div onclick="openOptionChain('${symbol}')">OPTION CHAIN</div>
            <div onclick="setAlert('${symbol}')">ALERT</div>
        </div>
    </div>
</div>;
});

watchlistDiv.innerHTML = html;
}

// ===== NEW WATCHLIST QUICK ACTION FUNCTIONS =====
function addMultipleStocks() {
    const symbols = prompt("Enter multiple stock symbols separated by commas (e.g., 'AAPL,GOOGL'):");
    if (symbols) {
        const symbolList = symbols.split(',').map(s => s.trim().toUpperCase()).filter(s => !watchlistStocks.includes(s));
        let addedCount = 0;

        symbolList.forEach(symbol => {
            if (!watchlistStocks.includes(symbol)) {
                watchlistStocks.push(symbol);
                addedCount++;
            }
        });
    }
}

```

```

        }
    });

    saveWatchlist();
    renderWatchlist();
    alert(`Added ${addedCount} new stocks to watchlist.`);
}
}

function importWatchlist() {
    alert("Import watchlist from CSV file feature would open here.");
}

function exportWatchlist() {
    if (watchlistStocks.length === 0) {
        alert("Watchlist is empty!");
        return;
    }

    const csvContent = "data:text/csv;charset=utf-8," + watchlistStocks.join("\n");
    const encodedUri = encodeURI(csvContent);
    const link = document.createElement("a");
    link.setAttribute("href", encodedUri);
    link.setAttribute("download", "watchlist.csv");
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);

    alert("Watchlist exported as CSV!");
}

function refreshWatchlist() {
    alert("Refreshing watchlist data...");
    // In real app, this would fetch latest prices
}

// ===== ENABLE/DISABLE OPTIONS =====
function toggleEnableOption(option) {
    enableOptions[option] = !enableOptions[option];
}

```

```
const statusElement = document.getElementById(`#${option}Status`);  
const sectionElement = document.getElementById(`#${option}Section`);  
  
if (enableOptions[option]) {  
    statusElement.className = 'enable-status enabled';  
    statusElement.innerHTML = '<i class="fas fa-check"></i>';  
  
    if (sectionElement) {  
        sectionElement.style.display = 'block';  
    }  
  
    // Enable specific controls  
    if (option === 'trailingSL') {  
        document.getElementById('trailingSLRow').style.display = 'flex';  
    }  
    if (option === 'profitLock') {  
        document.getElementById('profitLockRow').style.display = 'flex';  
    }  
} else {  
    statusElement.className = 'enable-status disabled';  
    statusElement.innerHTML = '<i class="fas fa-times"></i>';  
  
    if (sectionElement) {  
        sectionElement.style.display = 'none';  
    }  
  
    // Disable specific controls  
    if (option === 'trailingSL') {  
        document.getElementById('trailingSLRow').style.display = 'none';  
    }  
    if (option === 'profitLock') {  
        document.getElementById('profitLockRow').style.display = 'none';  
    }  
}  
  
// Update the enable option UI  
const optionElement = event.target.closest('.enable-option');  
if (optionElement) {
```

```

        if (enableOptions[option]) {
            optionElement.classList.add('active');
        } else {
            optionElement.classList.remove('active');
        }
    }

// ===== POSITION SECTION TOGGLE FUNCTIONS =====
function togglePositionSection() {
    const sectionBody = document.getElementById('positionSectionBody');
    const toggleBtn = document.getElementById('positionSectionToggle');

    positionSectionCollapsed = !positionSectionCollapsed;

    if (positionSectionCollapsed) {
        sectionBody.classList.add('collapsed');
        toggleBtn.innerHTML = '<i class="fas fa-chevron-down"></i> Expand';
    } else {
        sectionBody.classList.remove('collapsed');
        toggleBtn.innerHTML = '<i class="fas fa-chevron-up"></i> Collapse';
    }
}

function togglePartitionSection() {
    const sectionBody = document.getElementById('partitionSectionBody');
    const toggleBtn = document.getElementById('partitionSectionToggle');

    partitionSectionCollapsed = !partitionSectionCollapsed;

    if (partitionSectionCollapsed) {
        sectionBody.classList.add('collapsed');
        toggleBtn.innerHTML = '<i class="fas fa-chevron-down"></i> Expand';
    } else {
        sectionBody.classList.remove('collapsed');
        toggleBtn.innerHTML = '<i class="fas fa-chevron-up"></i> Collapse';
    }
}

```

```
// ===== POSITION SETTINGS TOGGLE FUNCTIONS =====
function initializePositionToggles() {
    // Add event listeners to all toggle switches
    const toggles = [
        'toggleOrderType', 'togglePart', 'toggleQty', 'toggleTrailingSL',
        'toggleTarget', 'toggleExit', 'toggleProfitLock', 'toggleProfitGap'
    ];

    toggles.forEach(toggleId => {
        const toggle = document.getElementById(toggleId);
        const controlId = toggleId.replace('toggle', 'position').replace('toggle', '');
        let control;

        if (toggleId === 'toggleOrderType') control = document.getElementById('positionOrderType');
        else if (toggleId === 'togglePart') control = document.getElementById('positionPart');
        else if (toggleId === 'toggleQty') {
            // For quantity, we have two controls
            document.getElementById('positionQty').disabled = !toggle.checked;
            document.getElementById('positionSellQty').disabled = !toggle.checked;
        }
        else if (toggleId === 'toggleTrailingSL') {
            control = document.getElementById('trailingSLValue');
            document.getElementById('trailingSLType').disabled = !toggle.checked;
        }
        else if (toggleId === 'toggleTarget') {
            control = document.getElementById('targetValue');
            document.getElementById('targetType').disabled = !toggle.checked;
        }
        else if (toggleId === 'toggleExit') {
            control = document.getElementById('exitValue');
            document.getElementById('exitType').disabled = !toggle.checked;
        }
        else if (toggleId === 'toggleProfitLock') control = document.getElementById('positionProfitLock');
        else if (toggleId === 'toggleProfitGap') control = document.getElementById('positionProfitGap');

        if (control && control.id !== 'positionQty' && control.id !== 'positionSellQty') {
            control.disabled = !toggle.checked;
        }
    });
}
```

```

toggle.addEventListener('change', function() {
    const isChecked = this.checked;

    if (toggleId === 'toggleQty') {
        document.getElementById('positionQty').disabled = !isChecked;
        document.getElementById('positionSellQty').disabled = !isChecked;
    }
    else if (toggleId === 'toggleTrailingSL') {
        document.getElementById('trailingSLValue').disabled = !isChecked;
        document.getElementById('trailingSLType').disabled = !isChecked;

        // Show/hide indicator options
        if (isChecked && document.getElementById('trailingSLType').value === 'i
            document.getElementById('indicatorOptions').style.display = 'grid';
            document.getElementById('indicatorPeriod').disabled = false;
            document.getElementById('indicatorCondition').disabled = false;
        } else {
            document.getElementById('indicatorOptions').style.display = 'none';
            document.getElementById('indicatorPeriod').disabled = true;
            document.getElementById('indicatorCondition').disabled = true;
        }
    }
    else if (toggleId === 'toggleTarget') {
        document.getElementById('targetValue').disabled = !isChecked;
        document.getElementById('targetType').disabled = !isChecked;
    }
    else if (toggleId === 'toggleExit') {
        document.getElementById('exitValue').disabled = !isChecked;
        document.getElementById('exitType').disabled = !isChecked;
    }
    else if (control) {
        control.disabled = !isChecked;
    }
});
});

// Add event listener for trailing SL type change
document.getElementById('trailingSLType').addEventListener('change', function() {
    const toggleTrailingSL = document.getElementById('toggleTrailingSL');
}

```

```

if (toggleTrailingSL.checked && this.value === 'indicator') {
    document.getElementById('indicatorOptions').style.display = 'grid';
    document.getElementById('indicatorPeriod').disabled = false;
    document.getElementById('indicatorCondition').disabled = false;
} else {
    document.getElementById('indicatorOptions').style.display = 'none';
    document.getElementById('indicatorPeriod').disabled = true;
    document.getElementById('indicatorCondition').disabled = true;
}
});

}

function toggleAllPositionSettings(state) {
const toggles = [
'toggleOrderType', 'togglePart', 'toggleQty', 'toggleTrailingSL',
'toggleTarget', 'toggleExit', 'toggleProfitLock', 'toggleProfitGap'
];
toggles.forEach(toggleId => {
    const toggle = document.getElementById(toggleId);
    toggle.checked = state;

    // Trigger change event
    const event = new Event('change');
    toggle.dispatchEvent(event);
});

// Show success message
const status = state ? 'enabled' : 'disabled';
alert(`All position settings ${status}`);
}

function resetPositionSettings() {
// Reset toggle states
toggleAllPositionSettings(true);

// Reset values
document.getElementById('positionOrderType').value = 'limit';
document.getElementById('positionPart').value = 'buy';
}

```

```

document.getElementById('positionQty').value = '10';
document.getElementById('positionSellQty').value = '0';
document.getElementById('trailingSLType').value = 'point';
document.getElementById('trailingSLValue').value = '20';
document.getElementById('targetType').value = 'point';
document.getElementById('targetValue').value = '50';
document.getElementById('exitType').value = 'point';
document.getElementById('exitValue').value = '30';
document.getElementById('profitLock').value = "";
document.getElementById('profitGap').value = "";

// Reset indicator options
document.getElementById('indicatorOptions').style.display = 'none';
document.getElementById('indicatorPeriod').value = '14';
document.getElementById('indicatorCondition').value = 'close';

alert('Position settings reset to defaults');
}

// ===== PARTITION & ADD QUANTITY FUNCTIONS =====
function calculatePositionSizing() {
    const entryPrice = parseFloat(document.getElementById('entryPrice').value) || 0;
    const positionQty = parseFloat(document.getElementById('positionQty').value);
    const positionSellQty = parseFloat(document.getElementById('positionSellQty').value);
    const partitionSquareOffQty = parseFloat(document.getElementById('partitionSquareOffQty').value);
    const partitionSquareOffPrice = parseFloat(document.getElementById('partitionSquareOffPrice').value);
    const addQty = parseFloat(document.getElementById('addQty').value) || 0;
    const addQtyPrice = parseFloat(document.getElementById('addQtyPrice').value);

    if (!entryPrice) {
        document.getElementById('calculationResult').innerHTML = "Please enter Entry Price";
        document.getElementById('calculationResult').style.color = "#ff4444";
        return;
    }

    const totalQty = positionQty + positionSellQty;
    if (totalQty === 0) {
        document.getElementById('calculationResult').innerHTML = "Please enter quantity";
        document.getElementById('calculationResult').style.color = "#ff4444";
    }
}

```

```

        return;
    }

let result = `<strong>Position Calculation:</strong><br>`;
result += `Total Quantity: ${totalQty}<br>`;
result += `Entry Price:  ${entryPrice.toFixed(2)}<br>`;

let partitionProfit = 0;
if (partitionSquareOffQty > 0 && partitionSquareOffPrice > 0) {
    partitionProfit = (partitionSquareOffPrice - entryPrice) * partitionSquareOffQ
    result += `Partition Profit:  ${partitionProfit.toFixed(2)}<br>`;
}

let addQtyCost = 0;
if (addQty > 0 && addQtyPrice > 0) {
    addQtyCost = addQty * addQtyPrice;
    result += `Add Quantity Cost:  ${addQtyCost.toFixed(2)}<br>`;
}

const totalInvestment = (entryPrice * totalQty) + addQtyCost;
result += `Total Investment:  ${totalInvestment.toFixed(2)}<br>`;

if (partitionProfit > 0) {
    const roi = (partitionProfit / totalInvestment) * 100;
    result += `ROI from Partition: ${roi.toFixed(2)}%`;
}

document.getElementById('calculationResult').innerHTML = result;
document.getElementById('calculationResult').style.color = "#00ff88";
}

// ===== POSITIONS MANAGEMENT =====
function loadPositions() {
    const savedPositions = localStorage.getItem('mukeshAlgoPositions');
    if (savedPositions) {
        positions = JSON.parse(savedPositions);
    } else {
        // Default sample positions
        positions = [

```

```
{  
    symbol: "RELIANCE",  
    orderType: "BUY",  
    orderCondition: "Intraday",  
    entryPrice: 2450.50,  
    currentPrice: 2465.75,  
    buyQty: 10,  
    sellQty: 0,  
    buyAvgPrice: 2445.00,  
    sellAvgPrice: 0,  
    source: "LIVE",  
    stopLoss: 2400.00,  
    target: 2500.00,  
    trailingSL: 0,  
    entryTime: new Date().toISOString(),  
    positionSettings: {  
        orderType: "limit",  
        part: "buy",  
        qty: 10,  
        trailingSL: {  
            type: "point",  
            value: 20  
        },  
        target: {  
            type: "point",  
            value: 50  
        },  
        exit: {  
            type: "point",  
            value: 30  
        },  
        profitLock: 2475,  
        profitGap: 5,  
        partitionSquareOffQty: 2,  
        partitionSquareOffPrice: 2470,  
        addQty: 5,  
        addQtyPrice: 2430  
    },  
},  
},
```

```
{  
    symbol: "TCS",  
    orderType: "SELL",  
    orderCondition: "Positional",  
    entryPrice: 3850.00,  
    currentPrice: 3825.50,  
    buyQty: 0,  
    sellQty: 5,  
    buyAvgPrice: 0,  
    sellAvgPrice: 3830.00,  
    source: "PAPER",  
    stopLoss: 3900.00,  
    target: 3800.00,  
    trailingSL: 20,  
    entryTime: new Date(Date.now() - 86400000).toISOString(),  
    positionSettings: {  
        orderType: "market",  
        part: "sell",  
        qty: 5,  
        trailingSL: {  
            type: "price",  
            value: 25  
        },  
        target: {  
            type: "price",  
            value: 3800  
        },  
        exit: {  
            type: "price",  
            value: 3810  
        },  
        profitLock: 3815,  
        profitGap: 10,  
        partitionSquareOffQty: 1,  
        partitionSquareOffPrice: 3815,  
        addQty: 2,  
        addQtyPrice: 3880  
    }  
}
```

```

    ];
    savePositions();
}

renderPositions();
updatePLSummary();
}

function savePositions() {
  localStorage.setItem('mukeshAlgoPositions', JSON.stringify(positions));
}

function renderPositions() {
  const table = document.getElementById('positionsTable');

  if (positions.length === 0) {
    table.innerHTML = `
      <tr>
        <td colspan="15" style="text-align:center;padding:30px;color:#888;font-
          No open positions. Click "Add Position" to create one.
        </td>
      </tr>
    `;
    return;
  }

  let html = "";
  positions.forEach((p, i) => {
    const buyValue = p.buyAvgPrice * p.buyQty;
    const sellValue = p.sellAvgPrice * p.sellQty;
    const currentValue = p.currentPrice * (p.buyQty - p.sellQty);
    const pl = currentValue - (buyValue - sellValue);
    const plPercent = (pl / (buyValue - sellValue)) * 100;

    let plClass = pl >= 0 ? "pl-positive" : "pl-negative";
    let orderTypeClass = p.orderType === 'BUY' ? 'order-type-buy' : 'order-type-s
    let conditionClass = `condition-${p.orderCondition.toLowerCase().replace(/\s

    // Get position settings info for tooltip
  
```

```

const posSettings = p.positionSettings || {};
const posInfo = posSettings.orderType ? `

<div style="font-size:12px;color:#aaa;margin-top:5px;">
  Type: ${posSettings.orderType.toUpperCase()} |
  Part: ${posSettings.part.toUpperCase()} |
  TSL: ${posSettings.trailingSL?.type || 'N/A'}
</div>
` : `

html += `

<tr>
  <td>${i + 1}</td>
  <td class="symbol-cell">
    ${p.symbol}
    ${posInfo}
    <div class="quick-popup">
      <button class="q-btn buy" onclick="openAddPositionModalForSymbol(${p.symbol})">Buy</button>
      <button class="q-btn sell" onclick="openAddPositionModalForSymbol(${p.symbol})">Sell</button>
      <button class="q-btn chart" onclick="openChart('${p.symbol}')">Chart</button>
      <button class="q-btn depth" onclick="openDepth('${p.symbol}')">Depth</button>
      <button class="q-btn delete-btn-main" onclick="deletePosition(${i})">Delete</button>
      <div class="q-btn more">More</div>
      <div class="more-menu-main">
        <div onclick="viewPositionDetails(${i})">View Details</div>
        <div onclick="modifyPositionSettings(${i})">Modify Settings</div>
        <div onclick="executePartitionSquareOff(${i})">Partition Square Off</div>
        <div onclick="executeAddQuantity(${i})">Add Quantity</div>
        <div onclick="openInfo('${p.symbol}')">INFO</div>
        <div onclick="openGTT('${p.symbol}')">GTT</div>
        <div onclick="openOptionChain('${p.symbol}')">OPTION CHAIN</div>
        <div onclick="setAlert('${p.symbol}')">ALERT</div>
      </div>
    </div>
  </td>
  <td><span class="${orderTypeClass}">${p.orderType}</span></td>
  <td><span class="order-condition ${conditionClass}">${p.orderCondition}</span></td>
  <td>${p.entryPrice.toFixed(2)}</td>
  <td>${p.currentPrice.toFixed(2)}</td>
</tr>
` : `

`;

```

```

<td class="buy-qty">${p.buyQty}</td>
<td class="sell-qty">${p.sellQty}</td>
<td>${p.buyAvgPrice} ? p.buyAvgPrice.toFixed(2) : '-'</td>
<td>${p.sellAvgPrice} ? p.sellAvgPrice.toFixed(2) : '-'</td>
<td class="${plClass}">${pl} >= 0 ? '+' : "${pl.toFixed(2)}</td>
<td class="${plClass}">${pl} >= 0 ? '+' : "${pl.toFixed(2)}</td>
<td class="${plClass}">${pl} >= 0 ? '+' : "${plPercent.toFixed(2)}%</td>
<td>
  <div class="source-wrap">
    <button class="source-btn">${p.source}</button>
    <div class="source-menu">
      <div onclick="changeSource(${i}, 'LIVE')">LIVE</div>
      <div onclick="changeSource(${i}, 'PAPER')">PAPER</div>
      <div onclick="changeSource(${i}, 'SIMULATOR')">SIMULATOR</div>
    </div>
  </div>
</td>
<td><button class="action-btn" onclick="openActionModal('${p.symbol}', ${p.id})">Action</button>
</td>
</tr>`;
});

table.innerHTML = html;
}

function updatePLSummary() {
  let realizedPL = 1250.50;
  let unrealizedPL = 0;
  let dayPL = 325.75;

  positions.forEach(position => {
    const pl = (position.currentPrice - position.entryPrice) * (position.buyQty - position.sellQty);
    if (position.orderType === 'SELL') {
      unrealizedPL -= pl;
    } else {
      unrealizedPL += pl;
    }
  });
}

const netPL = realizedPL + unrealizedPL;

```

```

const plBox = document.getElementById('plBox');
const formattedPL = netPL >= 0 ?
  `+ ${Math.abs(netPL).toLocaleString('en-IN', {minimumFractionDigits: 2})}` :
  `- ${Math.abs(netPL).toLocaleString('en-IN', {minimumFractionDigits: 2})}`;

plBox.textContent = `P/L : ${formattedPL}`;
plBox.style.color = netPL >= 0 ? 'var(--success-color)' : 'var(--danger-color)';
plBox.style.background = netPL >= 0 ?
  'linear-gradient(135deg, #0a3d0a 0%, #1a5c1a 100%)' :
  'linear-gradient(135deg, #3d0a0a 0%, #5c1a1a 100%)';
}

function updateCurrentPrices() {
  positions.forEach(position => {
    const changePercent = (Math.random() - 0.5) / 100;
    position.currentPrice = position.currentPrice * (1 + changePercent);
  });

  savePositions();
  renderPositions();
  updatePLSummary();
}

function updateMarketData() {
  // Update ticker data with random changes
  tickerData.forEach(stock => {
    const change = (Math.random() - 0.5) * 10;
    stock.price += change;
    stock.change = change;
  });

  populateMarketTicker();
  updateCurrentPrices();
}

// ===== POSITION MODAL FUNCTIONS =====
function openAddPositionModal() {
  document.getElementById('modalSymbolTitle').textContent = 'Add Position';
  document.getElementById('positionSymbol').value = '';
}

```

```

document.getElementById('positionSymbol').removeAttribute('readonly');

// Reset main form
document.getElementById('orderType').value = 'BUY';
document.getElementById('orderCondition').value = 'Intraday';
document.getElementById('buyQty').value = '10';
document.getElementById('sellQty').value = '0';
document.getElementById('entryPrice').value = "";
document.getElementById('currentPrice').value = "";
document.getElementById('buyAvgPrice').value = "";
document.getElementById('sellAvgPrice').value = "";
document.getElementById('positionSource').value = 'LIVE';

// Reset enable options
Object.keys(enableOptions).forEach(option => {
    const statusElement = document.getElementById(`#${option}Status`);
    const optionElement = document.querySelector(`[onclick="toggleEnableOpti

    if (enableOptions[option]) {
        statusElement.className = 'enable-status enabled';
        statusElement.innerHTML = '<i class="fas fa-check"></i>';
        if (optionElement) optionElement.classList.add('active');
    } else {
        statusElement.className = 'enable-status disabled';
        statusElement.innerHTML = '<i class="fas fa-times"></i>';
        if (optionElement) optionElement.classList.remove('active');
    }
});

// Reset position settings
document.getElementById('positionOrderType').value = 'limit';
document.getElementById('positionPart').value = 'buy';
document.getElementById('positionQty').value = '10';
document.getElementById('positionSellQty').value = '0';
document.getElementById('trailingSLType').value = 'point';
document.getElementById('trailingSLValue').value = '20';
document.getElementById('indicatorOptions').style.display = 'none';
document.getElementById('indicatorPeriod').value = '14';
document.getElementById('indicatorCondition').value = 'close';

```

```

document.getElementById('targetType').value = 'point';
document.getElementById('targetValue').value = '50';
document.getElementById('exitType').value = 'point';
document.getElementById('exitValue').value = '30';
document.getElementById('profitLock').value = "";
document.getElementById('profitGap').value = "";

// Reset partition & add quantity fields
document.getElementById('partitionSquareOffQty').value = '2';
document.getElementById('partitionSquareOffPrice').value = "";
document.getElementById('addQty').value = '5';
document.getElementById('addQtyPrice').value = "";
document.getElementById('maxPartitionCount').value = '5';
document.getElementById('autoSquareOffPercent').value = '10';
document.getElementById('reentryAttempts').value = '3';
document.getElementById('calculationResult').innerHTML = 'Enter values to ca
document.getElementById('calculationResult').style.color = '#aaa';

// Reset toggle states
toggleAllPositionSettings(true);

// Reset section states
positionSectionCollapsed = false;
document.getElementById('positionSectionBody').classList.remove('collapsed')
document.getElementById('positionSectionToggle').innerHTML = '<i class="fas

partitionSectionCollapsed = false;
document.getElementById('partitionSectionBody').classList.remove('collapsed')
document.getElementById('partitionSectionToggle').innerHTML = '<i class="fas

// Apply enable/disable states
Object.keys(enableOptions).forEach(option => {
  const sectionElement = document.getElementById(`#${option}Section`);
  if (sectionElement) {
    sectionElement.style.display = enableOptions[option] ? 'block' : 'none';
  }
});

document.getElementById('addPositionModal').style.display = 'flex';

```

```
}
```

```
function openAddPositionModalForSymbol(symbol, orderType = 'BUY') {
    document.getElementById('modalSymbolTitle').textContent = `Add Position - ${symbol}`;
    document.getElementById('positionSymbol').value = symbol;
    document.getElementById('positionSymbol').setAttribute('readonly', true);
    document.getElementById('orderType').value = orderType;
    document.getElementById('positionPart').value = orderType.toLowerCase();

    // Set default prices based on symbol
    const basePrice = symbol === 'RELIANCE' ? 2450 :
        symbol === 'TCS' ? 3850 :
        symbol === 'INFY' ? 1550 :
        Math.random() * 2000 + 1000;

    document.getElementById('entryPrice').value = basePrice.toFixed(2);
    document.getElementById('currentPrice').value = basePrice.toFixed(2);
    document.getElementById('buyAvgPrice').value = basePrice.toFixed(2);
    document.getElementById('sellAvgPrice').value = (basePrice * 1.02).toFixed(2);

    if (orderType === 'BUY') {
        document.getElementById('buyQty').value = 10;
        document.getElementById('sellQty').value = 0;
        document.getElementById('positionQty').value = 10;
        document.getElementById('positionSellQty').value = 0;
    } else {
        document.getElementById('buyQty').value = 0;
        document.getElementById('sellQty').value = 10;
        document.getElementById('positionQty').value = 0;
        document.getElementById('positionSellQty').value = 10;
    }

    // Set default position settings
    const slValue = basePrice * 0.02;
    const targetValue = basePrice * 0.03;

    document.getElementById('trailingSLValue').value = slValue.toFixed(2);
    document.getElementById('targetValue').value = targetValue.toFixed(2);
    document.getElementById('exitValue').value = (slValue * 0.5).toFixed(2);
```

```

document.getElementById('profitLock').value = (basePrice * 1.01).toFixed(2);
document.getElementById('profitGap').value = (basePrice * 0.005).toFixed(2);

// Set partition & add quantity defaults
document.getElementById('partitionSquareOffPrice').value = (basePrice * 1.01)
document.getElementById('addQtyPrice').value = (basePrice * 0.99).toFixed(2);

// Reset toggle states
toggleAllPositionSettings(true);

document.getElementById('addPositionModal').style.display = 'flex';
}

function closeAddModal() {
  document.getElementById('addPositionModal').style.display = 'none';
}

function outsideCloseAddModal(e) {
  if (e.target.id === "addPositionModal") closeAddModal();
}

function addPosition() {
  const symbol = document.getElementById('positionSymbol').value.trim().toUpperCase();
  const orderType = document.getElementById('orderType').value;
  const orderCondition = document.getElementById('orderCondition').value;
  const buyQty = parseInt(document.getElementById('buyQty').value) || 0;
  const sellQty = parseInt(document.getElementById('sellQty').value) || 0;
  const entryPrice = parseFloat(document.getElementById('entryPrice').value);
  const currentPrice = parseFloat(document.getElementById('currentPrice').value);
  const buyAvgPrice = parseFloat(document.getElementById('buyAvgPrice').value);
  const sellAvgPrice = parseFloat(document.getElementById('sellAvgPrice').value);
  const source = document.getElementById('positionSource').value;

  // Get enable options
  const positionSettingsEnabled = enableOptions.positionSettings;
  const partitionSettingsEnabled = enableOptions.partitionSettings;
  const trailingSLEnabled = enableOptions.trailingSL;
  const profitLockEnabled = enableOptions.profitLock;
}

```

```

// Get toggle states
const orderTypeEnabled = document.getElementById('toggleOrderType').checked;
const partEnabled = document.getElementById('togglePart').checked && positionEnabled;
const qtyEnabled = document.getElementById('toggleQty').checked && positionEnabled;
const trailingSLChecked = document.getElementById('toggleTrailingSL').checked;
const targetEnabled = document.getElementById('toggleTarget').checked && positionEnabled;
const exitEnabled = document.getElementById('toggleExit').checked && positionEnabled;
const profitLockChecked = document.getElementById('toggleProfitLock').checked;
const profitGapEnabled = document.getElementById('toggleProfitGap').checked;

// New position section fields
const positionOrderType = orderTypeEnabled ? document.getElementById('positionOrderType').value : '';
const positionPart = partEnabled ? document.getElementById('positionPart').value : '';
const positionQty = qtyEnabled ? parseInt(document.getElementById('positionQty').value) : 0;
const positionSellQty = qtyEnabled ? parseInt(document.getElementById('positionSellQty').value) : 0;
const trailingSLType = trailingSLChecked ? document.getElementById('trailingSLType').value : '';
const trailingSLValue = trailingSLChecked ? parseFloat(document.getElementById('trailingSLValue').value) : 0;
const indicatorPeriod = document.getElementById('indicatorPeriod').value;
const indicatorCondition = document.getElementById('indicatorCondition').value;
const targetType = targetEnabled ? document.getElementById('targetType').value : '';
const targetValue = targetEnabled ? parseFloat(document.getElementById('targetValue').value) : 0;
const exitType = exitEnabled ? document.getElementById('exitType').value : 'po';
const exitValue = exitEnabled ? parseFloat(document.getElementById('exitValue').value) : 0;
const profitLock = profitLockChecked ? parseFloat(document.getElementById('profitLock').value) : 0;
const profitGap = profitGapEnabled ? parseFloat(document.getElementById('profitGap').value) : 0;

// Partition & Add Quantity fields
const partitionSquareOffQty = partitionSettingsEnabled ? parseInt(document.getElementById('partitionSquareOffQty').value) : 0;
const partitionSquareOffPrice = partitionSettingsEnabled ? parseFloat(document.getElementById('partitionSquareOffPrice').value) : 0;
const addQty = partitionSettingsEnabled ? parseInt(document.getElementById('addQty').value) : 0;
const addQtyPrice = partitionSettingsEnabled ? parseFloat(document.getElementById('addQtyPrice').value) : 0;
const maxPartitionCount = partitionSettingsEnabled ? parseInt(document.getElementById('maxPartitionCount').value) : 0;
const autoSquareOffPercent = partitionSettingsEnabled ? parseFloat(document.getElementById('autoSquareOffPercent').value) : 0;
const reentryAttempts = partitionSettingsEnabled ? parseInt(document.getElementById('reentryAttempts').value) : 0;

if (!symbol) {
    alert("Please enter a stock symbol");
    return;
}

```

```

if (!entryPrice) {
  alert("Please enter an entry price");
  return;
}

if (!buyQty && !sellQty && !positionQty && !positionSellQty) {
  alert("Please enter either buy or sell quantity");
  return;
}

const finalQty = positionQty || buyQty || sellQty || positionSellQty;

const newPosition = {
  symbol: symbol,
  orderType: orderType,
  orderCondition: orderCondition,
  entryPrice: entryPrice,
  currentPrice: currentPrice,
  buyQty: buyQty || positionQty,
  sellQty: sellQty || positionSellQty,
  buyAvgPrice: buyAvgPrice,
  sellAvgPrice: sellAvgPrice,
  source: source,
  stopLoss: entryPrice * 0.98,
  target: entryPrice * 1.02,
  trailingSL: 0,
  entryTime: new Date().toISOString(),

  // New position settings with toggle states
  positionSettings: {
    enabled: positionSettingsEnabled,
    orderType: positionOrderType,
    part: positionPart,
    qty: finalQty,
    trailingSL: {
      type: trailingSLType,
      value: trailingSLValue,
      indicator: trailingSLType === 'indicator' ? {
        ...
      }
    }
  }
}

```

```

        period: indicatorPeriod,
        condition: indicatorCondition
    } : null,
    enabled: trailingSLChecked && trailingSLEnabled
},
target: {
    type: targetType,
    value: targetValue,
    enabled: targetEnabled
},
exit: {
    type: exitType,
    value: exitValue,
    enabled: exitEnabled
},
profitLock: profitLock,
profitLockEnabled: profitLockChecked && profitLockEnabled,
profitGap: profitGap,
profitGapEnabled: profitGapEnabled,

// Partition & Add Quantity settings
partitionEnabled: partitionSettingsEnabled,
partitionSquareOffQty: partitionSquareOffQty,
partitionSquareOffPrice: partitionSquareOffPrice,
addQty: addQty,
addQtyPrice: addQtyPrice,
maxPartitionCount: maxPartitionCount,
autoSquareOffPercent: autoSquareOffPercent,
reentryAttempts: reentryAttempts
}
};

positions.push(newPosition);
savePositions();
renderPositions();
updatePLSummary();
closeAddModal();

alert(`${orderType} position added for ${symbol} with position settings`);

```

```

}

// ===== ACTION MODAL FUNCTIONS =====
function openActionModal(symbol, index) {
    currentPositionIndex = index;
    const position = positions[index];

    document.getElementById('actionModalSymbolTitle').textContent = `Position A ${symbol}`

    // Set current values for modify tab
    document.getElementById('modifyStopLoss').value = position.stopLoss || '';
    document.getElementById('modifyTarget').value = position.target || '';
    document.getElementById('modifyTrailingSL').value = position.trailingSL || '';

    // Set values for exit tab
    const totalQty = Math.abs(position.buyQty - position.sellQty);
    document.getElementById('exitQty').value = totalQty;
    document.getElementById('exitPrice').value = position.currentPrice.toFixed(2);

    // Set values for partition tab
    const posSettings = position.positionSettings || {};
    document.getElementById('actionPartitionSquareOffQty').value = posSettings.squareOffQty || '';
    document.getElementById('actionPartitionSquareOffPrice').value = posSettings.squareOffPrice || '';
    document.getElementById('actionAddQty').value = posSettings.addQty || 0;
    document.getElementById('actionAddQtyPrice').value = posSettings.addQtyPrice || '';

    // Load existing code if any
    const savedCode = localStorage.getItem(`strategyCode_${symbol}`);
    if (savedCode) {
        document.getElementById('strategyCode').value = savedCode;
    }

    document.getElementById('actionModal').style.display = 'flex';
}

function closeActionModal() {
    document.getElementById('actionModal').style.display = 'none';
    currentPositionIndex = -1;
}

```

```

function outsideCloseActionModal(e) {
  if (e.target.id === "actionModal") closeActionModal();
}

function switchActionTab(e, tabId) {
  // Remove active class from all tabs
  document.querySelectorAll('.tabs button').forEach(btn => {
    btn.classList.remove('active');
  });

  // Add active class to clicked tab
  e.target.classList.add('active');

  // Hide all tab content
  document.querySelectorAll('.tab-content').forEach(tab => {
    tab.classList.remove('active');
  });

  // Show selected tab content
  document.getElementById(tabId).classList.add('active');
}

function modifyPosition() {
  if (currentPositionIndex === -1) return;

  const stopLoss = parseFloat(document.getElementById('modifyStopLoss').value);
  const target = parseFloat(document.getElementById('modifyTarget').value);
  const trailingSL = parseFloat(document.getElementById('modifyTrailingSL').value);

  positions[currentPositionIndex].stopLoss = stopLoss;
  positions[currentPositionIndex].target = target;
  positions[currentPositionIndex].trailingSL = trailingSL;

  savePositions();
  renderPositions();

  alert(`Position modified for ${positions[currentPositionIndex].symbol}`);
  closeActionModal();
}

```

```

}

function exitPosition() {
  if (currentPositionIndex === -1) return;

  const exitQty = parseInt(document.getElementById('exitQty').value);
  const exitPrice = parseFloat(document.getElementById('exitPrice').value);
  const exitReason = document.getElementById('exitReason').value;

  const position = positions[currentPositionIndex];
  const totalQty = Math.abs(position.buyQty - position.sellQty);

  if (exitQty > totalQty) {
    alert("Exit quantity cannot be greater than position quantity");
    return;
  }

  // In real app, you would handle partial exits
  positions.splice(currentPositionIndex, 1);
  savePositions();

  renderPositions();
  updatePLSummary();

  alert(`Position exited for ${position.symbol} at ${exitPrice}`);
  closeActionModal();
}

function saveStrategyCode() {
  const code = document.getElementById('strategyCode').value;
  if (code.trim() && currentPositionIndex !== -1) {
    const symbol = positions[currentPositionIndex].symbol;
    localStorage.setItem(`strategyCode_${symbol}`, code);
    alert("Strategy code saved successfully!");
  } else {
    alert("Please enter some code");
  }
}

```

```

function testStrategyCode() {
    const code = document.getElementById('strategyCode').value;
    if (code.trim()) {
        alert("Strategy code tested (simulation). In real app, this would execute the c
    } else {
        alert("Please enter some code to test");
    }
}

function saveSettings() {
    const alertPrice = parseFloat(document.getElementById('alertPrice').value);
    const autoSquareOff = document.getElementById('autoSquareOff').value;
    const maxLoss = parseFloat(document.getElementById('maxLoss').value);

    if (currentPositionIndex !== -1) {
        positions[currentPositionIndex].alertPrice = alertPrice;
        positions[currentPositionIndex].autoSquareOff = autoSquareOff;
        positions[currentPositionIndex].maxLoss = maxLoss;

        savePositions();
        alert("Settings saved for position!");
    }
    closeActionModal();
}

function executePartitionAction() {
    if (currentPositionIndex === -1) return;

    const partitionQty = parseFloat(document.getElementById('actionPartitionSqu
    const partitionPrice = parseFloat(document.getElementById('actionPartitionSq
    const addQty = parseFloat(document.getElementById('actionAddQty').value) |
    const addPrice = parseFloat(document.getElementById('actionAddQtyPrice').va
    const actionType = document.getElementById('partitionAction').value;

    const position = positions[currentPositionIndex];

    if (actionType === 'square_off' || actionType === 'both') {
        if (partitionQty > 0) {
            const profit = (partitionPrice - position.entryPrice) * partitionQty;
        }
    }
}

```

```

    alert(`Partition square off: ${partitionQty} units at  ${partitionPrice.toFixed(2)}`);

    // Update position
    if (position.orderType === 'BUY') {
        position.buyQty = Math.max(0, position.buyQty - partitionQty);
    } else {
        position.sellQty = Math.max(0, position.sellQty - partitionQty);
    }
}

if (actionType === 'add_qty' || actionType === 'both') {
    if (addQty > 0) {
        alert(`Added ${addQty} units at  ${addPrice.toFixed(2)})`);

        // Update position
        if (position.orderType === 'BUY') {
            position.buyQty += addQty;
            // Recalculate average price
            const totalCost = (position.buyAvgPrice * (position.buyQty - addQty)) + (addPrice * addQty);
            position.buyAvgPrice = totalCost / position.buyQty;
        } else {
            position.sellQty += addQty;
            // Recalculate average price
            const totalCost = (position.sellAvgPrice * (position.sellQty - addQty)) + (addPrice * addQty);
            position.sellAvgPrice = totalCost / position.sellQty;
        }
    }
}

savePositions();
renderPositions();
updatePLSummary();
closeActionModal();
}

// ===== POSITION MANAGEMENT FUNCTIONS =====
function deletePosition(index) {
    if (confirm(`Delete position for ${positions[index].symbol}?`)) {

```

```

    positions.splice(index, 1);
    savePositions();
    renderPositions();
    updatePLSummary();
}
}

function changeSource(index, source) {
    positions[index].source = source;
    savePositions();
    renderPositions();
    alert(`Source changed to ${source} for ${positions[index].symbol}`);
}

// New function to view position details
function viewPositionDetails(index) {
    const position = positions[index];
    const posSettings = position.positionSettings || {};

    let details = `
        <strong>Position Details for ${position.symbol}</strong><br><br>
        <strong>Order Details:</strong><br>
        - Order Type: ${position.orderType}<br>
        - Order Condition: ${position.orderCondition}<br>
        - Entry Price: ${position.entryPrice.toFixed(2)}<br>
        - Current Price: ${position.currentPrice.toFixed(2)}<br>
        - Buy Qty: ${position.buyQty}<br>
        - Sell Qty: ${position.sellQty}<br><br>
    `;

    if (posSettings.orderType) {
        details += `
            <strong>Position Settings:</strong><br>
            - Order Type: ${posSettings.orderType.toUpperCase()} ${posSettings.orderTy
            - Part: ${posSettings.part.toUpperCase()} ${posSettings.partEnabled !== false ? ' : ' }<br>
            - Quantity: ${posSettings.qty} ${posSettings.qtyEnabled !== false ? ' : ' }<br>
            - Trailing SL: ${posSettings.trailingSL?.type || 'N/A'} (${posSettings.trailingSL?
            - Target: ${posSettings.target?.type || 'N/A'} (${posSettings.target?.value || '0'})<br>
            - Exit: ${posSettings.exit?.type || 'N/A'} (${posSettings.exit?.value || '0'}) ${pos
    }
}

```

```

    - Profit Lock: ${posSettings.profitLock || '0'} ${posSettings.profitLockEnabled};
    - Profit Gap: ${posSettings.profitGap || '0'} ${posSettings.profitGapEnabled};
    `;

// Partition & Add Quantity details
if (posSettings.partitionSquareOffQty || posSettings.addQty) {
    details += `<strong>Partition & Add Quantity:</strong><br>;
    if (posSettings.partitionSquareOffQty) {
        details += `- Partition Square Off Qty: ${posSettings.partitionSquareOffQty}`;
        details += `- Partition Square Off Price: ${posSettings.partitionSquareOffPrice}`;
    }
    if (posSettings.addQty) {
        details += `- Add Quantity: ${posSettings.addQty}<br>;
        details += `- Add Quantity Price: ${posSettings.addQtyPrice} || '0.00'<br>;
    }
}
}

alert(details);
}

// New function to modify position settings
function modifyPositionSettings(index) {
    const position = positions[index];
    const posSettings = position.positionSettings || {};

    // Open the add position modal with existing values
    openAddPositionModalForSymbol(position.symbol, position.orderType);

    // Fill position settings
    if (posSettings.orderType) {
        document.getElementById('positionOrderType').value = posSettings.orderType;
        document.getElementById('positionPart').value = posSettings.part;
        document.getElementById('positionQty').value = posSettings.qty || position.qty;
        document.getElementById('trailingSLType').value = posSettings.trailingSL?.type;
        document.getElementById('trailingSLValue').value = posSettings.trailingSL?.value;
        document.getElementById('targetType').value = posSettings.target?.type || '';
        document.getElementById('targetValue').value = posSettings.target?.value || '';
        document.getElementById('exitType').value = posSettings.exit?.type || 'point';
    }
}

```

```

document.getElementById('exitValue').value = posSettings.exit?.value || 0;
document.getElementById('profitLock').value = posSettings.profitLock || 0;
document.getElementById('profitGap').value = posSettings.profitGap || 0;

// Set toggle states
document.getElementById('toggleOrderType').checked = posSettings.orderType === 'Market';
document.getElementById('togglePart').checked = posSettings.partEnabled !== false;
document.getElementById('toggleQty').checked = posSettings.qtyEnabled !== false;
document.getElementById('toggleTrailingSL').checked = posSettings.trailingSL;
document.getElementById('toggleTarget').checked = posSettings.target?.enabled;
document.getElementById('toggleExit').checked = posSettings.exit?.enabled;
document.getElementById('toggleProfitLock').checked = posSettings.profitLock;
document.getElementById('toggleProfitGap').checked = posSettings.profitGap;

// Trigger change events
document.getElementById('toggleOrderType').dispatchEvent(new Event('change'));
document.getElementById('togglePart').dispatchEvent(new Event('change'));
document.getElementById('toggleQty').dispatchEvent(new Event('change'));
document.getElementById('toggleTrailingSL').dispatchEvent(new Event('change'));
document.getElementById('toggleTarget').dispatchEvent(new Event('change'));
document.getElementById('toggleExit').dispatchEvent(new Event('change'));
document.getElementById('toggleProfitLock').dispatchEvent(new Event('change'));
document.getElementById('toggleProfitGap').dispatchEvent(new Event('change'));

// Partition & Add Quantity settings
document.getElementById('partitionSquareOffQty').value = posSettings.partitionSquareOffQty;
document.getElementById('partitionSquareOffPrice').value = posSettings.partitionSquareOffPrice;
document.getElementById('addQty').value = posSettings.addQty || 0;
document.getElementById('addQtyPrice').value = posSettings.addQtyPrice || 0;
document.getElementById('maxPartitionCount').value = posSettings.maxPartitionCount;
document.getElementById('autoSquareOffPercent').value = posSettings.autoSquareOffPercent;
document.getElementById('reentryAttempts').value = posSettings.reentryAttempts;

// Handle indicator options
if (posSettings.trailingSL?.type === 'indicator') {
    document.getElementById('indicatorOptions').style.display = 'grid';
    if (posSettings.trailingSL.indicator) {
        document.getElementById('indicatorPeriod').value = posSettings.trailingSL.indicator.period;
        document.getElementById('indicatorCondition').value = posSettings.trailingSL.indicator.condition;
    }
}

```

```

        }
    }

// Change modal title
document.getElementById('modalSymbolTitle').textContent = `Modify Position

// Update the addPosition function to handle modification
const originalAddPosition = window.addPosition;
window.addPosition = function() {
    // Remove old position
    positions.splice(index, 1);
    // Call original function to add modified position
    originalAddPosition();
    // Restore original function
    window.addPosition = originalAddPosition;
};

}

function executePartitionSquareOff(positionIndex) {
    const position = positions[positionIndex];
    if (!position) return;

    const posSettings = position.positionSettings || {};
    const partitionQty = posSettings.partitionSquareOffQty || 0;
    const partitionPrice = posSettings.partitionSquareOffPrice || position.currentF

    if (partitionQty <= 0) {
        alert("No partition square off quantity set for this position");
        return;
    }

    const confirmMsg = `Square off ${partitionQty} units of ${position.symbol} at

if (confirm(confirmMsg)) {
    // In real app, this would execute the trade
    const profit = (partitionPrice - position.entryPrice) * partitionQty;
    const profitType = profit >= 0 ? "Profit" : "Loss";
}

```

```

alert(`Partition square off executed!\n${partitionQty} units of ${position.symbol}`)

// Update position quantity
if (position.orderType === 'BUY') {
  position.buyQty = Math.max(0, position.buyQty - partitionQty);
} else {
  position.sellQty = Math.max(0, position.sellQty - partitionQty);
}

savePositions();
renderPositions();
updatePLSummary();
}

}

function executeAddQuantity(positionIndex) {
  const position = positions[positionIndex];
  if (!position) return;

  const posSettings = position.positionSettings || {};
  const addQty = posSettings.addQty || 0;
  const addPrice = posSettings.addQtyPrice || position.currentPrice;

  if (addQty <= 0) {
    alert("No add quantity set for this position");
    return;
  }

  const confirmMsg = `Add ${addQty} units to ${position.symbol} position at ${addPrice}`;

  if (confirm(confirmMsg)) {
    // In real app, this would execute the trade
    alert(`Added ${addQty} units to ${position.symbol} position at ${addPrice}. Total cost: ${totalCost}`);
  }
}

// Update position
if (position.orderType === 'BUY') {
  position.buyQty += addQty;
  // Recalculate average price
  const totalCost = (position.buyAvgPrice * (position.buyQty - addQty)) + (addPrice * addQty);
  position.buyAvgPrice = totalCost / position.buyQty;
}

```

```

        position.buyAvgPrice = totalCost / position.buyQty;
    } else {
        position.sellQty += addQty;
        // Recalculate average price
        const totalCost = (position.sellAvgPrice * (position.sellQty - addQty)) + (adc
        position.sellAvgPrice = totalCost / position.sellQty;
    }

    savePositions();
    renderPositions();
    updatePLSummary();
}
}

// ===== UTILITY FUNCTIONS =====
function openChart(symbol) {
    alert(`Opening chart for ${symbol}`);
}

function openDepth(symbol) {
    alert(`Opening market depth for ${symbol}`);
}

function openInfo(symbol) {
    alert(`Opening info for ${symbol}`);
}

function openGTT(symbol) {
    alert(`Opening GTT for ${symbol}`);
}

function openOptionChain(symbol) {
    alert(`Opening option chain for ${symbol}`);
}

function setAlert(symbol) {
    const price = prompt(`Set price alert for ${symbol}:`);
    if (price) {
        alert(`Alert set for ${symbol} at ${price}`);
    }
}

```

```

        }
    }

// ===== PAGE NAVIGATION =====
function showPage(pageId) {
    const pages = document.querySelectorAll('.content-page');
    pages.forEach(page => {
        page.classList.add('hidden');
    });
    document.getElementById(pageId).classList.remove('hidden');

    // Update active button in navigation
    const buttons = document.querySelectorAll('.nav-popup button');
    buttons.forEach(btn => {
        btn.classList.remove('active');
        if (btn.onclick && btn.onclick.toString().includes(pageId)) {
            btn.classList.add('active');
        }
    });
}

// Hide navigation popup after clicking
hideNavPopup();
}

// ===== SCREENER POPUP FUNCTIONS =====
function openScreenerPopup() {
    document.getElementById('screenerPopupModal').style.display = 'flex';
    hideNavPopup();
}

function closeScreenerPopup(e) {
    if (!e || e.target.id === "screenerPopupModal") {
        document.getElementById('screenerPopupModal').style.display = 'none';
    }
}

function openChartink() {
    window.open('https://chartink.com/screener', '_blank');
    closeScreenerPopup();
}

```

```

}

function deployIndicatorScreener() {
    const indicator = document.getElementById('indicatorSelect').value;
    const value = document.getElementById('indicatorValue').value;

    if (indicator === 'Select Indicator' || !value) {
        alert('Please select an indicator and enter a value');
        return;
    }

    alert(`Indicator Based Screener Deployed!\n${indicator} = ${value}`);
    closeScreenerPopup();
}

function deploySavedScreener() {
    alert('Saved Screener Deployed!');
    closeScreenerPopup();
}

function createNewScreener() {
    alert('Create New Screener - Opening Screener Builder');
    closeScreenerPopup();
}

// ===== STRATEGY CREATION POPUP FUNCTIONS =====
function openStrategyPopup() {
    document.getElementById('strategyCreationPopup').style.display = 'flex';
    hideNavPopup();
}

function closeStrategyPopup(e) {
    if (!e || e.target.id === "strategyCreationPopup") {
        document.getElementById('strategyCreationPopup').style.display = 'none';
    }
}

function openOpstra() {
    window.open('https://opstra.definedge.com', '_blank');
}

```

```

        closeStrategyPopup();
    }

function openSensibull() {
    window.open('https://sensibull.com', '_blank');
    closeStrategyPopup();
}

function openMarketPlace() {
    showPage('marketplacePage');
    closeStrategyPopup();
}

function openCreateStrategy() {
    alert('Opening Strategy Creation Page');
    // You can redirect to your strategy creation page here
    closeStrategyPopup();
}

function backtestStrategy() {
    showPage('backtestPage');
    closeStrategyPopup();
}

function optimizeStrategy() {
    alert('Strategy Optimization Feature');
}

function shareStrategy() {
    alert('Share Strategy Feature');
}

// ===== BROKER MANAGEMENT FUNCTIONS =====
function loadBrokers() {
    const savedBrokers = localStorage.getItem('mukeshAlgoBrokers');
    const brokerList = document.getElementById('brokerList');

    if (!savedBrokers) {
        brokerList.innerHTML = '<div style="text-align:center; color:#888; padding:20px"><h2>No Brokers Available</h2></div>';
    } else {
        brokerList.innerHTML = savedBrokers;
    }
}

```

```

        return;
    }

const brokers = JSON.parse(savedBrokers);
let html = "";

brokers.forEach((broker, index) => {
    html += `
        <div style="background:#2a2a2a; padding:12px; margin-bottom:8px; border
            <div>
                <strong style="color:var(--accent-color);">${broker.name}</strong>
                <div style="font-size:12px; color:#aaa;">Client ID: ${broker.clientId}</div>
                <div style="font-size:11px; color:#${broker.status === 'connected' ? '00ff
                    </div>
                    <div style="display:flex; gap:5px;">
                        <button onclick="editBroker(${index})" style="padding:5px 10px; backgr
                            Edit
                        </button>
                        <button onclick="testBrokerConnection(${index})" style="padding:5px 1
                            Test
                        </button>
                    </div>
                </div>;
            );
        
```

brokerList.innerHTML = html;

```

    }

function addEditBroker() {
    const brokerSelect = document.getElementById('brokerSelect');
    const clientId = document.getElementById('clientId').value;
    const apiKey = document.getElementById('apiKey').value;
    const apiSecret = document.getElementById('apiSecret').value;
    const accessToken = document.getElementById('accessToken').value;

    if (!brokerSelect.value || !clientId) {
        alert('Please select broker and enter Client ID');
        return;
    }
}

```

```

const brokerName = brokerSelect.options[brokerSelect.selectedIndex].text;

let brokers = JSON.parse(localStorage.getItem('mukeshAlgoBrokers') || '[]');

// Check if broker already exists
const existingIndex = brokers.findIndex(b => b.clientId === clientId);

const brokerData = {
    name: brokerName,
    clientId: clientId,
    apiKey: apiKey,
    apiSecret: apiSecret,
    accessToken: accessToken,
    status: 'pending',
    lastUpdated: new Date().toISOString()
};

if (existingIndex >= 0) {
    brokers[existingIndex] = brokerData;
    alert('Broker updated successfully!');
} else {
    brokers.push(brokerData);
    alert('Broker added successfully!');
}

localStorage.setItem('mukeshAlgoBrokers', JSON.stringify(brokers));
loadBrokers();

// Clear form
document.getElementById('clientId').value = "";
document.getElementById('apiKey').value = "";
document.getElementById('apiSecret').value = "";
document.getElementById('accessToken').value = "";
brokerSelect.value = "";
}

function deleteBroker() {
    const clientId = document.getElementById('clientId').value;

```

```

if (!clientId) {
    alert('Please enter Client ID to delete');
    return;
}

if (confirm('Are you sure you want to delete this broker?')) {
    let brokers = JSON.parse(localStorage.getItem('mukeshAlgoBrokers') || '[]');
    brokers = brokers.filter(b => b.clientId !== clientId);
    localStorage.setItem('mukeshAlgoBrokers', JSON.stringify(brokers));
    loadBrokers();

    // Clear form
    document.getElementById('clientId').value = "";
    document.getElementById('apiKey').value = "";
    document.getElementById('apiSecret').value = "";
    document.getElementById('accessToken').value = "";
    document.getElementById('brokerSelect').value = "";

    alert('Broker deleted successfully!');
}
}

function checkBrokerConnection() {
    const clientId = document.getElementById('clientId').value;

    if (!clientId) {
        alert('Please enter Client ID to check connection');
        return;
    }

    // Simulate connection check
    const status = Math.random() > 0.3 ? 'Connected' : 'Failed';
    const message = status === 'Connected' ?
        ' Broker connection successful!' :
        ' Connection failed. Please check your credentials.';

    alert(message);
}

```

```

function editBroker(index) {
  const brokers = JSON.parse(localStorage.getItem('mukeshAlgoBrokers') || '[]');
  const broker = brokers[index];

  if (broker) {
    // Set form values
    document.getElementById('brokerSelect').value = broker.name.replace(' ', '');
    document.getElementById('clientId').value = broker.clientId;
    document.getElementById('apiKey').value = broker.apiKey;
    document.getElementById('apiSecret').value = broker.apiSecret;
    document.getElementById('accessToken').value = broker.accessToken;

    alert('Broker details loaded. Update and click "Add/Update Broker" to save changes');
  }
}

function testBrokerConnection(index) {
  const brokers = JSON.parse(localStorage.getItem('mukeshAlgoBrokers') || '[]');
  const broker = brokers[index];

  if (broker) {
    // Simulate connection test
    setTimeout(() => {
      const status = Math.random() > 0.2 ? 'connected' : 'disconnected';
      broker.status = status;
      localStorage.setItem('mukeshAlgoBrokers', JSON.stringify(brokers));
      loadBrokers();

      alert(status === 'connected' ?
        'Broker connection test successful!' :
        'Connection failed. Please check your credentials.');
    }, 1000);
  }
}

// Initialize brokers when page loads
function initializeBrokerPage() {
  loadBrokers();
}

```

```

}

// ===== FULL SCREEN MODE =====
function fullScreenMode() {
    const elem = document.documentElement;

    if (!document.fullscreenElement) {
        if (elem.requestFullscreen) {
            elem.requestFullscreen();
        } else if (elem.webkitRequestFullscreen) {
            elem.webkitRequestFullscreen();
        } else if (elem.msRequestFullscreen) {
            elem.msRequestFullscreen();
        }
    } else {
        if (document.exitFullscreen) {
            document.exitFullscreen();
        } else if (document.webkitExitFullscreen) {
            document.webkitExitFullscreen();
        } else if (document.msExitFullscreen) {
            document.msExitFullscreen();
        }
    }
}

// Handle fullscreen change
document.addEventListener('fullscreenchange', updateFullscreenStyles);
document.addEventListener('webkitfullscreenchange', updateFullscreenStyles);
document.addEventListener('msfullscreenchange', updateFullscreenStyles);

function updateFullscreenStyles() {
    const isFullscreen = document.fullscreenElement ||
        document.webkitFullscreenElement ||
        document.msFullscreenElement;

    const header = document.querySelector('.header');
    const ticker = document.querySelector('.market-ticker-container');
    const navTrigger = document.getElementById('navHoverTrigger');
    const content = document.getElementById('mainContent');
}

```

```
const watchlistPanel = document.getElementById('watchlistPanel');

if (isFullscreen) {
    header.style.display = 'none';
    ticker.style.display = 'none';
    navTrigger.style.display = 'none';
    watchlistPanel.style.display = 'none';

    content.style.margin = '0';
    content.style.padding = '20px';
    content.style.width = '100%';
    content.style.height = '100vh';
    content.style.position = 'fixed';
    content.style.top = '0';
    content.style.left = '0';
    content.style.zIndex = '10000';
    content.style.background = 'var(--dark-bg)';
    content.style.overflow = 'auto';
} else {
    header.style.display = 'flex';
    ticker.style.display = 'block';
    navTrigger.style.display = 'block';
    watchlistPanel.style.display = 'block';

    content.style.margin = "";
    content.style.padding = "";
    content.style.width = "";
    content.style.height = "";
    content.style.position = "";
    content.style.top = "";
    content.style.left = "";
    content.style.zIndex = "";
    content.style.background = "";
    content.style.overflow = "";
}

function resetAllData() {
    if (confirm('This will reset ALL positions, watchlist and brokers. Are you sure?'))
```

```
localStorage.removeItem('mukeshAlgoPositions');
localStorage.removeItem('mukeshAlgoWatchlist');
localStorage.removeItem('mukeshAlgoBrokers');
positions = [];
watchlistStocks = ["RELIANCE", "TCS", "INFY", "HDFCBANK"];
saveWatchlist();
renderPositions();
renderWatchlist();
loadBrokers();
updatePLSummary();
alert('All data has been reset!');

}

}

function logOut() {
if (confirm('Are you sure you want to log out?')) {
    alert('Logging out...');

    // In a real application, you would redirect to login page
    // window.location.href = '/login';
}
}

// ===== INITIALIZATION =====
document.addEventListener('DOMContentLoaded', function() {
    // Initialize clock
    updateClock();
    setInterval(updateClock, 1000);

    // Initialize market ticker
    populateMarketTicker();

    // Load watchlist
    loadWatchlist();

    // Load positions
    loadPositions();

    // Initialize broker page
    initializeBrokerPage();
})
```

```
// Initialize position toggles
initializePositionToggles();

// Show open positions page by default
showPage('openPositionsPage');

// Add event listeners for watchlist
document.getElementById('watchlistSearchBtn').addEventListener('click', addToWatchlist);
document.getElementById('watchlistSearch').addEventListener('keydown', function(e) {
    if (e.key === 'Enter') {
        addToWatchlist();
    }
});
document.getElementById('clearWatchlistBtn').addEventListener('click', clearWatchlist);

// Start market data updates
setInterval(updateMarketData, 5000);

// Initial market data update
updateMarketData();

// Add keyboard shortcuts
document.addEventListener('keydown', function(e) {
    if (e.key === 'Escape') {
        closeAddModal();
        closeActionModal();
        closeScreenerPopup();
        closeStrategyPopup();
    }
    if (e.key === 'F11') {
        e.preventDefault();
        fullScreenMode();
    }
    if (e.ctrlKey && e.key === 'p') {
        e.preventDefault();
        openAddPositionModal();
    }
});
```

```
});  
</script>
```