

Document Name	HSE-FOR-8.1.1 TLB Prestart Check
Document ID	HSE-140
Version Date	01/11/2024
Version	1
Approved	HSM
NO	ZAM-KAN-IR-005

TLB PRE-START CHECKLIST

DATE	10/14/2025
ODOMETER READING	1003
FUEL LEVEL	345345345


ROLLER	<pre>3import React, { useState, useRef, useEffect } from 'react'; import { View, Text, StyleSheet, TouchableOpacity, StatusBar, TextInput, Image, Dimensions, Animated, Modal, FlatList, ScrollView, KeyboardAvoidingView, Platform, ActivityIndicator, Alert, InteractionManager, } from 'react-native'; import LinearGradient from 'react-native-linear-gradient'; import MaterialIcons from 'react-native-vector- icons/MaterialIcons'; import { useNavigation, useRoute } from '@react- navigation/native'; import AsyncStorage from '@react-native-async-storage/async- storage'; import RNHTMLtoPDF from 'react-native-html-to-pdf'; import RNFS from 'react-native-fs'; import Toast from 'react-native-toast-message'; import NetInfo from '@react-native- community/netinfo'; import JobCardAPI from '../ConfigMongo/JobCardMongoConfig.js'; import InspectionAPI from '../ConfigMongo/InspectionMongoConfig'; import DropDownPicker from 'react-native- dropdown-picker'; import DateTimePicker from '@react-native- community/datetimepicker'; import { getFilteredTeamMembers } from '../Utils/teamMemberUtils'; import { filterEquipmentByPermission } from '../Utils/normalizeEquipmentData'; import * as ElementDropdown from 'react-native- element-dropdown'; const { width, height } = Dimensions.get('window'); const TOTAL_HEADER_HEIGHT = width * 0.3; // Helper for chunked state updates const chunkArray = (array, chunkSize) => { const result = []; for (let i = 0; i < array.length; i += chunkSize) { result.push(array.slice(i, i + chunkSize)); } return result; }; const CreateWorkOrder = () => { const navigation = useNavigation(); const route = useRoute(); const scrollViewRef = useRef(null); const [workOrderTitle, setWorkOrderTitle] = useState(""); const [memo, setMemo] = useState(""); const [priority, setPriority] = useState(null); const [selectedEquipment, setSelectedEquipment] = useState(null); const [assignTo, setAssignTo] = useState([]); const [taskStatusMenuVisibleId,</pre>
--------	--

```
setTaskStatusMenuVisibleId] =
useState(null); const [teamMembers,
setTeamMembers] = useState([]); const
[startDate, setStartDate] = useState(null);
const [dueDate, setDueDate] =
useState(null); const
[showStartDatePicker,
setShowStartDatePicker] =
useState(false); const
[showDueDatePicker,
setShowDueDatePicker] = useState(false);
const [additionalDetails,
setAdditionalDetails] = useState(""); const
[isAdditionalDetailsOpen,
setIsAdditionalDetailsOpen] =
useState(false); const [workOrderType,
setWorkOrderType] = useState('General');
const [workOrderTypeModalVisible,
setWorkOrderTypeModalVisible] =
useState(false); const [estimatedCost,
setEstimatedCost] = useState(""); const
[estimatedLaborTime,
setEstimatedLaborTime] = useState("");
const [linkInspection, setLinkInspection] =
useState(null); const
[linkInspectionModalVisible,
setLinkInspectionModalVisible] =
useState(false); const [linkedWorkOrder,
setLinkedWorkOrder] = useState(null);
const [linkedWorkOrderModalVisible,
setLinkedWorkOrderModalVisible] =
useState(false); const [isCostDetailsOpen,
setIsCostDetailsOpen] = useState(false);
const [partMaterialCost,
setPartMaterialCost] = useState(""); const
[totalLaborCost, setTotalLaborCost] =
useState(""); const [additionalCost,
setAdditionalCost] = useState(""); const
[taxAmount, setTaxAmount] = useState("");
const [taxPercentage, setTaxPercentage] =
useState(""); const [totalWoCost,
setTotalWoCost] = useState(""); const
[selectedCategory, setSelectedCategory] =
useState('Task'); const
[addItemModalVisible,
setAddItemModalVisible] =
useState(false); const [newItemText,
setNewItemText] = useState(""); const
[items, setItems] = useState([]); const
[taskType, setTaskType] =
useState('Repair'); const [note, setNote] =
useState(""); const [tempAssignTo,
setTempAssignTo] = useState([]); const
[statusModalVisible,
setStatusModalVisible] = useState(false);
const [selectedItemId, setSelectedItemId]
= useState(null); const [editModalVisible,
setEditModalVisible] = useState(false);
const [editItem, setEditItem] =
useState(null); const [editMenuVisible,
setEditMenuVisible] = useState(false);
const [addLaborModalVisible,
setAddLaborModalVisible] =
useState(false); const
[addPartsModalVisible,
setAddPartsModalVisible] =
useState(false); const
[addAdditionalCostModalVisible,
```

	<pre>setAddAdditionalCostModalVisible] = useState(false); const [addRemarksModalVisible, setAddRemarksModalVisible] = useState(false); const [laborWorker, setLaborWorker] = useState(""); const [laborCode, setLaborCode] = useState(""); const [laborRate, setLaborRate] = useState(""); const [laborHours, setLaborHours] = useState(""); const [laborMinutes, setLaborMinutes] = useState(""); const [laborNote, setLaborNote] = useState(""); const [partsItemNumber, setPartsItemNumber] = useState(""); const [partsPartNumber, setPartsPartNumber] = useState(""); const [partsItemName, setPartsItemName]</pre>
EQUIPMENT NAME	Test Equipment 8
EQUIPMENT MODEL NUMBER	Test Model

1. ALL DEFECTS UNDER "A" CLASSIFICATION TO BE ATTENDED IMMEDIATELY
2. ALL DEFECTS UNDER "B" CLASSIFICATION TO BE ATTENDED WITHIN 48 HOURS
3. ALL DEFECTS UNDER "C" CLASSIFICATION TO BE ATTENDED TO AS SOON AS POSSIBLE

DESCRIPTION	CLASS	IN ORDER (✓)	DEFECTIVE (X)	N/A	COMMENT
WORKING LIGHTS FRONT	A		X		
WORKING LIGHTS REAR	A		X		
REMARKS	gf				

Operator:	fg	 Signature