STORED PROCEDURES

1. Procedure For Login

```
CREATE PROCEDURE usp_Login(@UserName VARCHAR(50),@Password VARCHAR(50))
AS
BEGIN
        BEGIN TRY
                        IF EXISTS(SELECT UserName FROM tbl_Login WHERE UserName = @UserName)
                BEGIN
                        IF EXISTS(SELECT UserName,Password FROM tbl_Login WHERE UserName =
@UserName AND Password = @Password)
                                BEGIN
                                        RETURN 1
                                END
                        ELSE
                                BEGIN
                                        RETURN 2
                                END
                END
                ELSE
                BEGIN
                        RETURN 3
                END
        END TRY
        BEGIN CATCH
                RETURN -99
        END CATCH
END
```

2. Procedure For Adding Element

```
CREATE PROCEDURE usp_AddElement(@ElementName VARCHAR(50),@ElementDisplayName
VARCHAR(50),@Tool VARCHAR(50),@TimeZone CHAR(2),@UpdatedBy VARCHAR(50))
AS
BEGIN
        DECLARE @Active INT
        SET @Active = 1
        DECLARE @LastUpdatedGMT DATETIME
        SET @LastUpdatedGMT = GETDATE()
        BEGIN TRY
                IF EXISTS(SELECT ElementName FROM tbl_Elements WHERE ElementName =
@ElementName AND ElementDisplayName = @ElementDisplayName AND Tool = @Tool)
                BEGIN
                        RETURN 1
                END
                ELSE
                BEGIN
                         INSERT INTO tbl_Elements
VALUES(@ElementName,@ElementDisplayName,@Tool,@TimeZone,@Active,@LastUpdatedGMT,@UpdatedB
y)
                         RETURN 2
                END
        END TRY
        BEGIN CATCH
                RETURN -99
        END CATCH
```

```
END
```

3. Procedure For Disabling Element

```
CREATE PROCEDURE usp_DisableElement(@ElementId INT)
AS
BEGIN
       BEGIN TRY
              DECLARE @Active INT
              SET @Active = 0
              IF(@Active = (SELECT Active FROM tbl_Elements WHERE ElementId = 
@ElementId))
              BEGIN
                     RETURN 1
              END
              ELSE
              BEGIN
                      UPDATE tbl_Elements SET Active = 0 WHERE ElementId = @ElementId
                      RETURN 2
              END
       END TRY
       BEGIN CATCH
              RETURN -99
       END CATCH
END
```

4. Procedure For Disabling Element

```
CREATE PROCEDURE usp_UpdateElement(@ElementId INT,@ElementName 
VARCHAR(50),@ElementDisplayName VARCHAR(50),@TimeZone CHAR(2),@UpdatedBy VARCHAR(50))
AS
BEGIN
       DECLARE @LastUpdatedGMT DATETIME
       SET @LastUpdatedGMT = GETDATE()
       BEGIN TRY
              IF EXISTS(SELECT ElementName FROM tbl_Elements WHERE ElementName = 
@ElementName AND ElementDisplayName = @ElementDisplayName AND ElementId = @ElementId)
              BEGIN
                     RETURN 1
              END
              ELSE
              BEGIN
                     UPDATE tbl_Elements SET ElementName = @ElementName WHERE ElementId = 
@ElementId
                     UPDATE tbl_Elements SET ElementDisplayName = @ElementDisplayName 
WHERE ElementId = @ElementId
                     UPDATE tbl_Elements SET TimeZone = @TimeZone WHERE ElementId = 
@ElementId
                     UPDATE tbl_Elements SET LastUpdatedGMT = @LastUpdatedGMT WHERE 
ElementId = @ElementId
                     UPDATE tbl_Elements SET UpdatedBy = @UpdatedBy WHERE ElementId = 
@ElementId
                     RETURN 2
              END
       END TRY
       BEGIN CATCH
              RETURN -99
       END CATCH
```

```
END
```

5. Procedure For Adding Schedule

```
CREATE PROCEDURE usp_AddSchedule(@ScheduleName VARCHAR(50),@StartHour INT,@EndHour
INT,@UpdatedBy VARCHAR(50))
AS
BEGIN
    DECLARE @LastUpdatedGMT DATETIME
    SET @LastUpdatedGMT = GETDATE()
    DECLARE @Count INT
    DECLARE @SHour INT
    DECLARE @EHour INT
    DECLARE @Active INT
    BEGIN TRY
        SET @Active = 1
        SET @Count = (SELECT Count(*) FROM tbl_Schedule)
        IF(@Count = 0)
        BEGIN
            INSERT INTO tbl_Schedule
VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)
            RETURN 1
        END
        ELSE
        BEGIN
             IF(@Count = 1)
             BEGIN
                SET @SHour = (SELECT StartHour FROM tbl_Schedule)
                SET @EHour = (SELECT EndHour FROM tbl_Schedule)
                IF(@SHour < @Ehour)
                BEGIN
                    IF(@StartHour > @SHour AND @EndHour < @EHour AND @EHour
> @StartHour)
                    BEGIN
                        RETURN -1
                    END
                    ELSE
                    BEGIN
                        INSERT INTO tbl_Schedule
VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)
                        RETURN 1
                    END
                END
                ELSE IF(@Shour > @Ehour)
                BEGIN
                    IF(@StartHour > @SHour AND @EndHour > @EHour)
                    BEGIN
                        RETURN -1
                    END
                    ELSE IF(@StartHour > @SHour AND @EndHour < @EHour)
                    BEGIN
                        RETURN -1
                    END
                    ELSE
                    BEGIN
                        INSERT INTO tbl_Schedule
VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)
```

```
                                          RETURN 1
                                   END
                             END
                      END
                      ELSE IF(@Count > 1)
                      BEGIN
                            INSERT INTO tbl_Schedule
VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)
                            RETURN 1
                      END
              END
       END TRY
       BEGIN CATCH
              RETURN -99
       END CATCH
END
```

6. Procedure For Disabling Schedule

```
CREATE PROCEDURE usp_DisableSchedule(@ScheduleId INT)
AS
BEGIN
       BEGIN TRY
              DECLARE @Active INT
              SET @Active = 0
              IF(@Active = (SELECT Active FROM tbl_Schedule WHERE ScheduleId =
@ScheduleId))
              BEGIN
                     RETURN 1
              END
              ELSE
              BEGIN
                      UPDATE tbl_Schedule SET Active = 0 WHERE ScheduleId = @ScheduleId
                      RETURN 2
              END
       END TRY
       BEGIN CATCH
              RETURN -99
       END CATCH
END
```

7. Procedure For Updating Schedule

```
CREATE PROCEDURE usp_UpdateSchedule(@ScheduleId INT,@ScheduleName VARCHAR(50),@StartHour
INT,@EndHour INT,@UpdatedBy VARCHAR(50))
AS
BEGIN
        DECLARE @LastUpdatedGMT DATETIME
        SET @LastUpdatedGMT = GETDATE()
        BEGIN TRY
                IF EXISTS(SELECT ScheduleName FROM tbl_Schedule WHERE ScheduleName =
@ScheduleName AND StartHour = @StartHour AND EndHour = @EndHour AND ScheduleId =
@ScheduleId)
                BEGIN
                        RETURN 1
                END
                ELSE
                BEGIN
                        UPDATE tbl_Schedule SET ScheduleName = @ScheduleName WHERE
ScheduleId = @ScheduleId
                        UPDATE tbl_Schedule SET StartHour = @StartHour WHERE ScheduleId =
@ScheduleId
                        UPDATE tbl_Schedule SET EndHour = @EndHour WHERE ScheduleId =
@ScheduleId
                        UPDATE tbl_Schedule SET LastUpdatedGMT = @LastUpdatedGMT WHERE
ScheduleId = @ScheduleId
                        UPDATE tbl_Schedule SET UpdatedBy = @UpdatedBy WHERE ScheduleId =
@ScheduleId
                        RETURN 2
                END
        END TRY
        BEGIN CATCH
                RETURN -99
        END CATCH
END
```

8. Procedure For Adding KPI

```
CREATE PROCEDURE usp_AddKPI(@KPIId INT,@ScheduleId INT,@DarkGreenValue INT,@GreenValue
INT,@YellowValue INT,@OrangeValue INT,@RedValue INT,@KpiName VARCHAR(50),@OrderName
CHAR(3),@Tool VARCHAR(50),@UpdatedBy VARCHAR(50))
AS
BEGIN
        DECLARE @Active INT
        SET @Active = 1
        DECLARE @LastUpdatedGMT DATETIME
        SET @LastUpdatedGMT = GETDATE()
        BEGIN TRY
                IF EXISTS(SELECT * FROM tbl_KPI WHERE KPIId = @KPIId AND ScheduleId =
@ScheduleId)
                BEGIN
                        RETURN 1
                END
                ELSE
                BEGIN
                        INSERT INTO tbl_KPI
VALUES(@KPIId,@ScheduleId,@DarkGreenValue,@GreenValue,@YellowValue,@OrangeValue,@RedValue
,@KpiName,@OrderName,@Tool,@Active,@LastUpdatedGMT,@UpdatedBy)
```

```
                        RETURN 2
                END
        END TRY
        BEGIN CATCH
                RETURN -99
        END CATCH
END
```

9. Procedure For Disabling KPI

```
CREATE PROCEDURE usp_DisableKPI(@KPIId INT,@ScheduleId INT)
AS
BEGIN
        BEGIN TRY
                DECLARE @Active INT
                SET @Active = 0
                IF(@Active = (SELECT Active FROM tbl_KPI WHERE KPIId = @KPIId AND ScheduleId = @ScheduleId))
                BEGIN
                        RETURN 1
                END
                ELSE
                BEGIN
                         UPDATE tbl_KPI SET Active = 0 WHERE KPIId = @KPIId AND ScheduleId = @ScheduleId
                         RETURN 2
                END
        END TRY
        BEGIN CATCH
                RETURN -99
        END CATCH
END
```

10. Procedure For Updating KPI

```
CREATE PROCEDURE usp_UpdateKPI(@KPIId INT,@DarkGreenValue INT,@GreenValue INT,@YellowValue INT,@OrangeValue INT,@RedValue INT,@KpiName VARCHAR(50),@OrderName CHAR(3),@Tool VARCHAR(50),@UpdatedBy VARCHAR(50))
AS
BEGIN
        DECLARE @LastUpdatedGMT DATETIME
        SET @LastUpdatedGMT = GETDATE()
        BEGIN TRY
                        UPDATE tbl_KPI SET DarkGreenValue = @DarkGreenValue WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET GreenValue = @GreenValue WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET YellowValue = @YellowValue WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET OrangeValue = @OrangeValue WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET RedValue = @RedValue WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET KPIName = @KpiName WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET OrderName = @OrderName WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET Tool = @Tool WHERE KPIId = @KPIId
                        UPDATE tbl_KPI SET LastUpdatedGMT = @LastUpdatedGMT WHERE KPIId = @KPIId
```

```
            UPDATE tbl_KPI SET UpdatedBy = @UpdatedBy WHERE KPIId = @KPIId
            RETURN 2
    END TRY
    BEGIN CATCH
        RETURN -99
    END CATCH
END
```