Create database:

nama database : Tree

copy and paste this code to the sql:
(copy paste for each table, don't copy paste all)

First table:

```
CREATE TABLE newforestori (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `BlockX` int(10) NOT NULL,
  `BlockY` int(10) NOT NULL,
  `x` int(100) NOT NULL,
  `y` int(100) NOT NULL,
  `TreeNum` varchar(9) NOT NULL COLLATE utf8mb4_general_ci,
  `species` varchar(4) NOT NULL COLLATE utf8mb4_general_ci,
  `spgroup` int(7) NOT NULL,
  `Diameter` float NOT NULL,
  `DiameterClass` int(10) NOT NULL,
  `StemHeight` float NOT NULL,
  `volume` float NOT NULL,
  `production` float NOT NULL,
  `status_tree` varchar(10) COLLATE utf8mb4_general_ci DEFAULT NULL,
  `Cut_Angle` int(11) NOT NULL,
  `Cut_tree` varchar(10) NOT NULL COLLATE utf8mb4_general_ci,
  `Damage_crown` int(10) NOT NULL,
  `Damage_stem` int(10) NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Second table:

```
CREATE TABLE scientificname (
  SP-Code varchar(5) NOT NULL,
  Local-Name varchar(20) NOT NULL,
  Gen-Code varchar(20) NOT NULL,
  SP-Name varchar(20) NOT NULL,
  Royal-Class varchar(20) NOT NULL
);
```

Third Table :

```
CREATE TABLE speciesname (
  species_id INT PRIMARY KEY AUTO_INCREMENT,  -- Auto-incrementing integer for unique ID
  species_name VARCHAR(50) NOT NULL,  -- Stores species name (up to 50 characters)
  description TEXT,  -- Optional: Detailed description (longer text)
  common_name VARCHAR(30)  -- Optional: Common name of the species (up to 30 characters)
);
```

Fourth table :

```
CREATE TABLE damagetree (
 id INT(10) NOT NULL AUTO_INCREMENT,
 Cut_tree VARCHAR(10) NOT NULL,
 Victim VARCHAR(10) NOT NULL,
 Category_Damage INT(2) NOT NULL,
 PRIMARY KEY (id)
);
```

Fifth table :

```
CREATE TABLE victim (
 Victim varchar(10) NOT NULL,
 cut_tree varchar(500) NOT NULL,
 Category_damage int(2) NOT NULL
);
```

Next Step:

open table speciesname and import species.csv

Next Step:

open table scientificname import scientifname.Csv

Step 1: Run create_forest.php Refresh it for 4 time to get 10 000 ++ data of tree
Step 2: Run this sql code 2. UPDATE newforestori SET Volume = 3.142 * POW((Diameter / 200), 2) * StemHeight * 0.50 3. UPDATE newforestori SET TreeNum = CONCAT('T', LPAD(BlockX, 2, '0'), LPAD(BlockY, 2, '0'), LPAD(x, 2, '0'), LPAD(y, 2, '0')) 4. UPDATE newforestori SET status_tree = CASE WHEN spgroup IN (1, 2, 3, 5) AND Diameter > 45 THEN 'Cut' WHEN spgroup IN (1, 2, 3, 5) AND Diameter <= 45 THEN 'Keep' ELSE status_tree END 5. UPDATE newforestori SET Cut_Angle = CASE WHEN status_tree = 'Cut' THEN FLOOR(RAND() * 360) + 1 ELSE NULL END 6. UPDATE newforestori INNER JOIN speciesname ON speciesname.No = newforestori.species SET newforestori.species = speciesname.species 7. UPDATE newforestori SET Production = 3.142 * POW((Diameter / 200), 2) * StemHeight * 0.50 WHERE status_tree = 'cut'; 8. UPDATE newforestori SET cut_tree = CONCAT(x, ',', y) WHERE status_tree = 'cut';
Step 4: Run Find_damage.php

Step 5: run this sql code : INSERT INTO victim (Victim, cut_tree, Category_damage) SELECT Victim, cut_tree, Category_damage FROM damagetree;
Step 6: UPDATE newforestori nf INNER JOIN (SELECT Cut_tree, SUM(CASE WHEN Category_Damage = 1 THEN 1 ELSE 0 END) AS stem_victim_count, SUM(CASE WHEN Category_Damage = 2 THEN 1 ELSE 0 END) AS crown_victim_count FROM damagetree GROUP BY Cut_tree) AS dt ON nf.cut_tree = dt.Cut_tree SET nf.Damage_stem = dt.stem_victim_count, nf.Damage_crown = dt.crown_victim_count;