HU Spark Assignment

Using the provided dataset and schema, complete the assessment questions exclusively with the Spark DataFrame API, without utilizing Spark SQL.

Schema:

```
sales:
 - order_id
 - email
 - transaction_timestamp
 - total_item_quantity
 - purchase_revenue_in_usd
 - unique_items
 - items:
     - coupon
     - item_id
     - item_name
     - item_revenue_in_usd
     - price_in_usd
     - quantity
```

```
events:
 - device
 - ecommerce:
     - purchase_revenue_in_usd
     - total_item_quantity
     - unique_items
 - event_name
 - event_previous_timestamp
 - event_timestamp
 - geo:
     - city
     - state
 - items:
     - coupon
     - item_id
     - item_name
     - item_revenue_in_usd
     - price_in_usd
     - quantity
 - traffic_source
 - user_first_touch_timestamp
 - user_id
```

```
products:
 - item_id
 - name
 - price
```

```
users:
 - user_id
 - user_first_touch_timestamp
 - email
 - updated
```

Questions:

1. Read the data from the given CSV files, and create spark dataframes. flatten the nested columns in sales and events tables to create separate columns for each field.
2. Calculate total revenue per user and identify the top 10 users based on the total revenue from the sales table.
3. Identify the top 10 products based on the quantity sold.
4. Identify the overall revenue and quantity of all items purchased with using coupons and without coupons.
5. Find top 3 products per user based on purchase revenue.
6. Identify the users who have not purchased any products in the last 6 months, and display their latest purchase date.
7. Identify the 10 most valuable users based on the purchase amount spent in the last 3 months.
8. Considering email id of a user is PII (Personally Identifiable Information), create a function to perform a simple masking logic on each email id.
   Rules of masking:
   1. If the username (part of the email before the @ symbol) is more than 3 characters long, only the last 3 characters should be visible and all the preceding characters should be replaced by * (asterisk).
   Example: `john.doe@gmail.com` should be replaced by `*****doe@gmail.com`
   2. Else if the username is more than 1 character and less than or equal to 3 characters long, only the last character should be visible and rest of the username should be masked accordingly.
   Example: `ben@yahoo.com` should be replaced by `**n@yahoo.com`
   `ab@hotmail.com` should be replaced by `*b@hotmail.com`
   3. Else if it's only 1 character, it should be replaced by *
   Example: `a@gmail.com` should be replaced by `*@gmail.com`